BEA AquaLogic Enterprise Security™®

Administration and Deployment Guide

Version 2.1
Document Revised: May 19, 2006

Copyright

Restricted Rights Legend

Third-Party Software License Agreement

INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You acknowledge that this software is not designed or intended for use in the design, construction, operation or maintenance of any nuclear facility.

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

BEA, BEA JRockit, BEA Liquid Data for WebLogic, BEA WebLogic Server, Built on BEA, Jolt, JoltBeans, SteelThread, Top End, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA AquaLogic, BEA AquaLogic Data Services Platform, BEA AquaLogic Enterprise Security, BEA AquaLogic Service Bus, BEA AquaLogic Service Registry, BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Manager, BEA MessageQ, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Enterprise Security, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Java Adapter for Mainframe, BEA WebLogic JDriver, BEA WebLogic JRockit, BEA WebLogic Log Central, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server Process Edition, BEA WebLogic WorkGroup Edition, BEA WebLogic Workshop, and Liquid Computing are trademarks of BEA Systems, Inc. BEA Mission Critical Support is a service mark of BEA Systems, Inc. All other company and product names may be the subject of intellectual property rights reserved by third parties.

All other trademarks are the property of their respective companies.

About This Document

1. ALES Architecture

2. Starting and Stopping ALES Components

3. Configuring SSL for Production Environments

4. Failover and System Reliability

About This Document

This document describes tasks associated with deploying and managing ALES. It is organized as follows:

- Chapter 1, “ALES Architecture,”describes ALES components and deployment architecture.
- Chapter 2, “Starting and Stopping ALES Components,” provides startup and shutdown instructions.
- Chapter 3, “Configuring SSL for Production Environments,” describes how to replace the the ALES demonstration certificates with production-level certificates for secure-SSL communication between ALES components.
- Chapter 4, “Failover and System Reliability,” describes ALES features that support recovery from failure.

Audience

This guide is written for administrators who deploy ALES components on a network and make sure that ALES processes run as intended. These administrators have a general knowledge of security concepts and the Java security architecture. They understand Java, XML, deployment descriptors, and can identify security events in server and audit logs.

Dev2dev Web Site

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

http://dev2dev.bea.com

To view the documentation for a particular product, select that product from the 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.
- *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.

CHAPTER 1

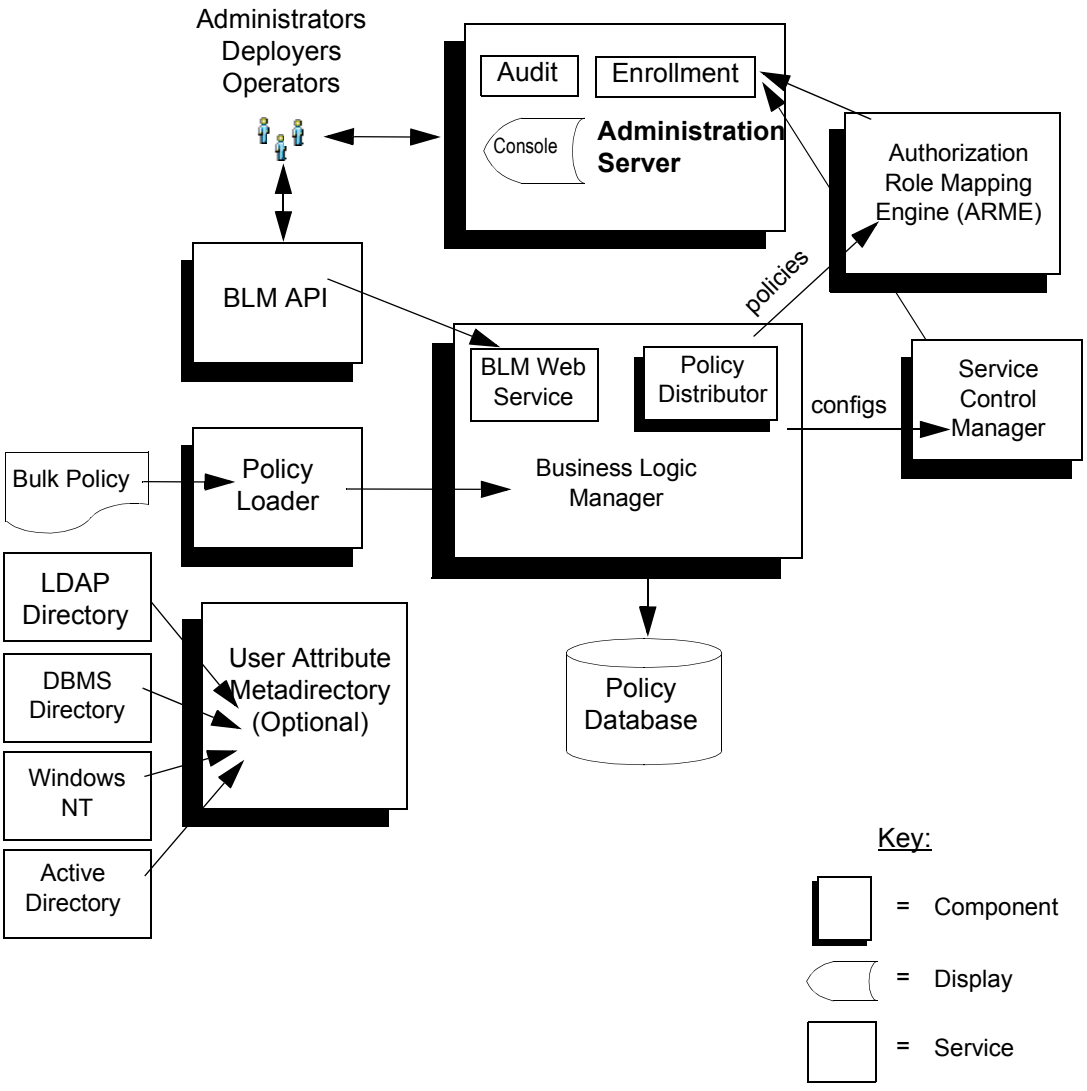
ALES Architecture

This section describes ALES components and provides information about deploying them on the network.

ALES Components

The following diagram gives a high-level view of ALES components.

Figure 1-1 High-Level View of ALES 2.1 Components



Administration Server

The Administration Server is a servlet-based application and can run in both WebLogic and Tomcat. It consists of the following components:

Business Logic Manager–The BLM is responsible for managing security policies stored in the Policy Database. The BLM includes the policy distributor which pushes policy to the runtime tier of ALES. The BLM features an external API for managing policy and configuration.

Policy Database–Maintains policy data in a relational database. This data is distributed to the Security Service Modules by the Policy Distributor.

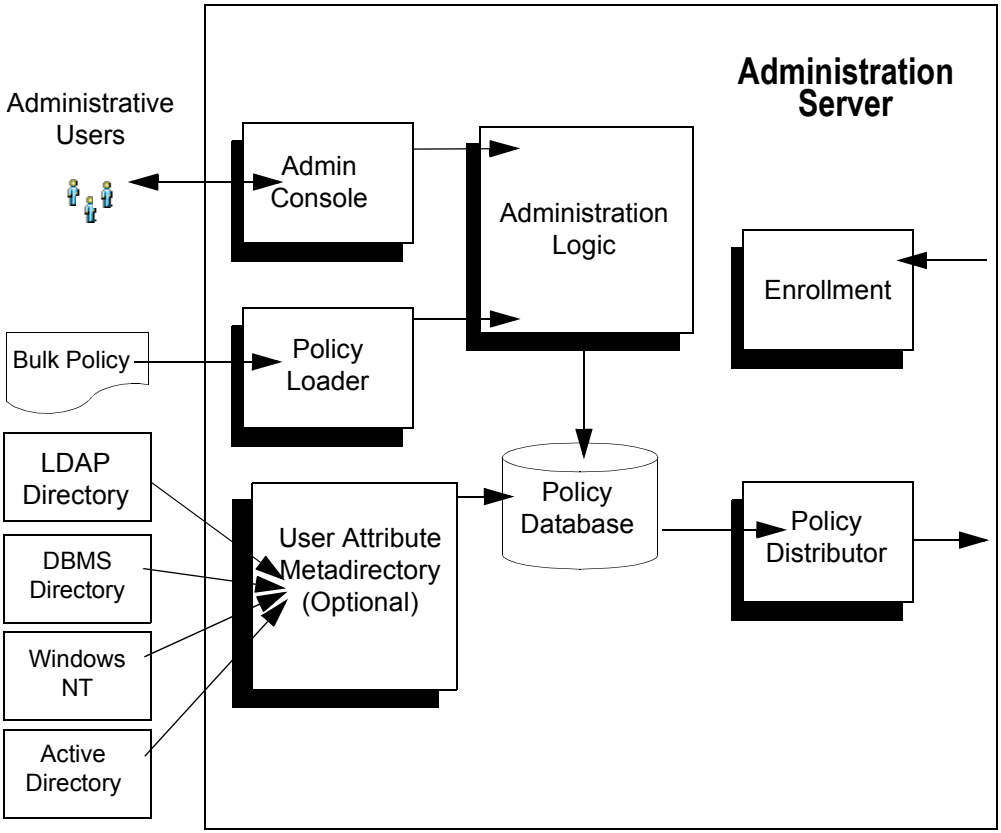
Policy Loader–Imports policy data from an external file. The external file can be generated by another system or another Administrative Server, or it can be manually coded. For additional information on how to use the Policy Loader, see the *Policy Managers Guide*.

Authorization and Role Mapping Engine (ARME)–Enforces security policy for Administration Server and console as it does for any other runtime application.

Administrative Console–Supports administrative policy security and administration delegation through a web browser-based user interface. Security configuration, policy configuration, user attributes (if required), resources, and rules are all managed through the console.

Metadirectory–Stores user attributes from a variety of sources for use in making policy decisions. The metadirectory assembles attributes for each user and caches them for use by Security Service Modules.

Figure 1-2 Administration Server Architecture



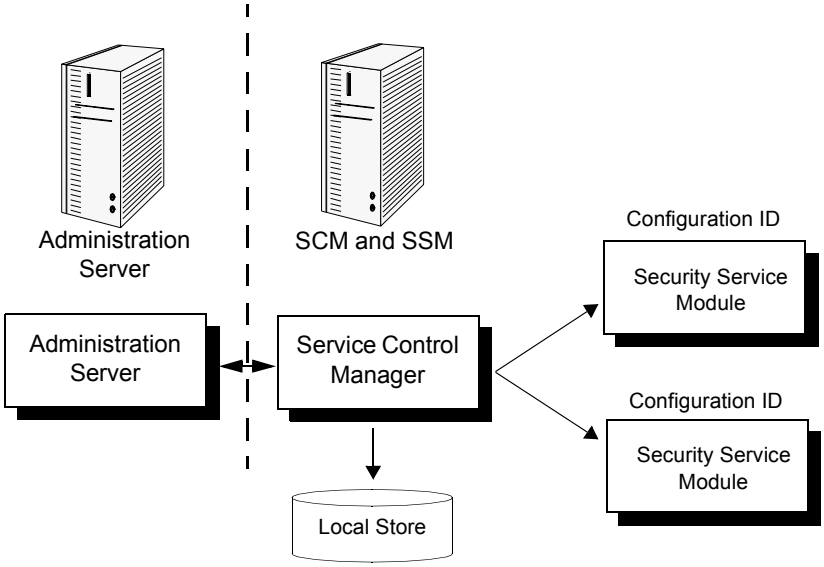
Service Control Manager (SCM)

The Service Control Module (SCM) is an essential component ALES's remote administration mechanism. Each Service Control Module stores SSM configuration data and provides each SSM on its machine the appropriate data.

The Service Control Manager receives and stores both full and incremental configuration updates. When a configuration change relevant to a SSM is made, it is provisioned to the Service Control Manager through the Policy Distributor. The provisioning mechanism ensures that only the configuration data absolutely required by a Service Control Manager is provisioned to that

module. Likewise, the Service Control Manager ensures that only the configuration data absolutely required by an SSM is made available to that module.

Figure 1-3 Service Control Manager



Security Service Module (SSM)

SSMs are a platform specific security plug-ins that are embedded in applications, application servers, and web servers to be secured by ALES. The SSM ties the application server (or applications, web servers) into ALES so that all security administration for the application is performed through ALES.

Configuration data for each module is specified centrally and then distributed to and locally cached on the appropriate machine. A benefit of this architecture is that there is no impact on the application if the Administration Server is stopped.

Table 1-1 below describes the SSM modules provided with ALES.

Table 1-1 SSM Modules

SSM Name	Description
WebLogic Server 8.1	Provides runtime enforcement of security services for applications created for WebLogic Server 8.1 and WebLogic Portal 8.1.
IIS Web Server	Provides runtime enforcement of security services for applications running on the Microsoft Internet Information Server. Supports basic single sign-on between Web servers and between the Web tier and the application tier.
Apache Web Server	Provides runtime enforcement of security services for applications running on the ASF Apache Web Server. Supports basic single sign-on between Web servers and between the Web tier and the application tier.
Web Services	Provides runtime enforcement of security services for generic applications making Web Service calls to obtain ALES security services.
Java	Runtime enforcement of security services for generic Java applications.

Security Providers

Security providers are used to provide authentication, authorization, auditing, role mapping, and credential mapping, and other services. Each SSM can be configured with a set of security providers as described in Table 1-2.

Table 1-2 ALES Security Providers

Provider	Description
Authentication Provider	Performs authentication services for for the SSM. Authentication providers are available to for Microsoft Windows NT, Active Directory, LDAP, relational databases, and others. Identity Asserters are Authentication Providers that accept encrypted identity tokens (e.g., SAML assertions) and return the corresponding authenticated subjects.
Credential Mapper	Allows the Security Service Module to generate credentials for user logins to an external repository or service. This is commonly used for either Single Sign On or access into a remote system on behalf of an authenticated subject (user or group).
Authorization Provider	Controls access to resources based on role and authorization policies. Access decisions provided through a role-based authorization provider incorporate relevant environmental, contextual, and transaction-specific information, allowing security policies to support business processes throughout the organization.
Role Mapping Provider	Supports dynamic role associations by obtaining the set of roles granted to a user for a resource.
Adjudication Provider	Resolves authorization conflicts when multiple authorization providers are in use.
Auditing Provider	Provides an electronic trail of transaction activity. Can include changes to system configuration parameters, policy changes, and transactions. For each audit item, the information can include who, what, when, where, and sometimes why.

Deployment Architecture

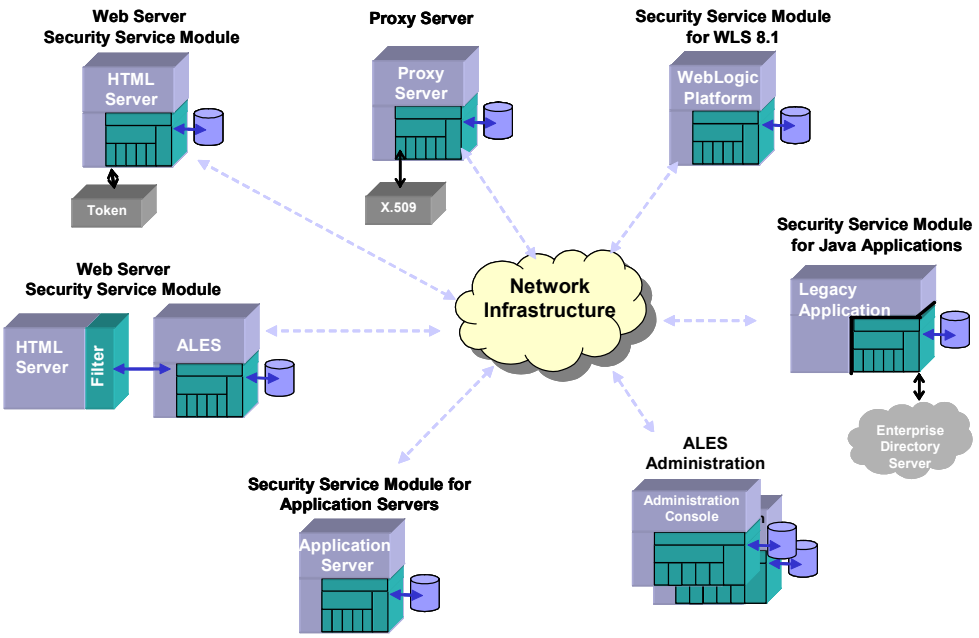
An ALES environment can consist of a single or multiple instances of the Administration Server, one or more Service Control Managers (hosted on individual machines), and any number of Security Service Modules, each associated with an SCM. Each Security Service Module may share or use different configuration or policy data, based on the business needs of an organization.The Administration Server serves as a central point of contact for instances and system administration tools.

Generalized Architecture

Installation of ALES depends on the application environment being secured. The basic requirement is that the Administration Server must be accessible to all Security Service Modules that are "plugged" into the applications being secured in that domain. A Service Control Manager must be installed on any machine running one or more SSMs.

Figure 1-4 below shows SSMs deployed on varying application environments and connecting to the Administration Server on a separate machine.

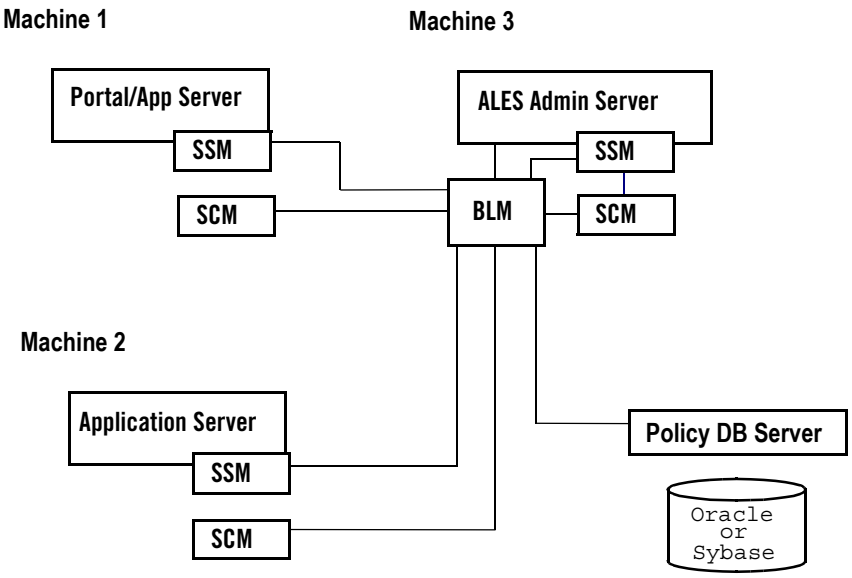
Figure 1-4 Distributed Computing Security Infrastructure



Location of ALES Components

Figure 1-5 below provides some insight into the interconnections of the ALES components.

Figure 1-5 Location of ALES Components



WebLogic Clusters

You can configure multiple servers to be part of a WebLogic cluster to support failover. A cluster is a group of server instances that work together to provide scalability and high-availability for applications. For instructions, see the *Installing the Administration Server*.

CHAPTER 2

Starting and Stopping ALES Components

This chapter details how to start and stop the Administration Server, Security Control Managers, and Security Service Modules on Windows and UNIX systems.

Starting and Stopping the Administration Server On Windows

The Administration Server is installed on Windows as a service application with a default startup type of 'manual'. To configure ALES services for automatic startup, use the Windows Services applet.

Starting the Administration Server starts the following services, where *name* is the machine name:

- ALES ARME.admin.server.asi.*name*
- ALES BLM.asi.*name*
- ALES Service Control Manager
- ALES WebLogic Server—WLS.asi.*name*

Table 2-1 lists the command line commands and Start Menu options for managing Administration Server processes. To use a command line, open a command window, navigate to the installation directory, and enter the command.

Table 2-1 Windows Program Menu Options and Commands

Menu Option	Command	Description
Start Server	`WLESadmin start`	Starts Administration Server processes when running under WebLogic, as well as the SCM on the same machine.
	`WLESadmin console`	Starts WebLogic-hosted Administration Server processes in separate console windows. When starting in console mode, a message like the following appears: `08/25/04 18:21:11utc ERR [3040]iomanager.cpp(95): ***** Opening ERR Stream *****` This is NOT an error message. It indicates a test to ensure that the server can write to an error log. You must start the SCM separately when using this command.
Stop Server	`WLESadmin stop`	Stops Administration Server processes. When running in console mode, you may also stop a process by closing the console window or pressing Ctrl+C.

Starting and Stopping Administration Server on UNIX

The Administration Server is registered with the UNIX init subsystem. By default, it is not configured to start automatically. To configure it for automatic startup, the system administrator must link it into the correct init runlevel.

On Sun Solaris and Linux platforms, you must always start the server as root. A utility, such as SUDO (http://www.courtesan.com/sudo/), can be used to allow non-root users to start and stop it as root without having to give out the root password or violate the Application Security Infrastructure (ASI).

To start and stop Administration Server processes on UNIX, navigate to the install directory and enter the shell script command as listed in Table 2-2.

Table 2-2 UNIX Commands

Command	Description
WLESadmin.sh start or WLESadmin.sh console	Either command starts Administration Server processes as daemon processes. **Note:** Either command provides the same results.
WLESadmin.sh stop	Stops Administration Server processes. A process can also be stopped by closing the console window or pressing Ctrl+C.

Administration Server Startup Option on Linux

To allow the Administration Server start up after a reboot on Linux, set it to start on runlevel3 (non-graphical runlevel) and runlevel5 (graphical runlevel). To do this, run the following command as root:

```
chkconfig --level 35 WLESadmin on
```

The database configuration is available to these scripts on boot so long as configurations are located in the `/etc/profile` directory. If the configuration is not located in this directory, edit `bin/WLESadmin.sh`, set the appropriate environment variables and paths before rebooting.

To check the Administration Server runlevel, run:

```
chkconfig --list WLESadmin
```

Starting and Stopping SCMs and SSMs on Windows

The SCM is installed on Windows as a service application with a default startup type of 'manual'. To configure an SCM for automatic startup, use the Windows Services applet.

Table 2-3 lists the command line commands and Start Menu options for starting/stopping SCMs and SSM instances. To use a command line in Windows, open a command window, go to the SCM or SSM instance install directory, and enter the command.

The SCM must be running before starting the SSM instance. If the SSM instance is on the same machine as the Administration Server, the SCM may have been started when the Administration Server was booted. If the SSM instance is on a different machine, you must first start its SCM.

Table 2-3 Windows Start Menu Options and Commands

Menu Option	Command	Description
Refresh SCM	`WLESscm refresh`	Clears cached configuration data and loads fresh SSM configuration data from the Administration Server.
Start SCM	`WLESscm start`	Starts the Service Control Manager.
Start SCM (console mode)	`WLESscm console`	Starts the Service Control Manager in a console window.
Stop SCM	`WLESscm stop`	Stops the Service Control Manager. In console mode, you may also stop it by closing the console window or pressing `Ctrl+C`.
Refresh ARME	`WLESarme refresh`	Updates the SSM to include the most recent policy data from the Application Server.
Start ARME	`WLESarme start`	Starts the SSM instance.
Start ARME (console mode)	`WLESarme console`	Starts the SSM instance in a console window.
Stop ARME	`WLESarme stop`	Stops the SSM instance. In console mode, you may also stop the instance by closing the console window or pressing `Ctrl+C`.

Starting and Stopping SCMs and SSMs on UNIX

To start and stop SCMs and SSM instances on UNIX, go to the `bin` directory where the SCM or SSM instance is installed and enter the commands listed in Table 2-4. You must start the Service Control Manager before starting the SSM instance.

Note: For an additional SCM start-up option on Linux, see "SCM Start-Up Option on Linux" on page 2-5.

Table 2-4 Unix Commands

Command	Description
`WLESscm.sh refresh`	Clears cached configuration data and loads fresh Security Service Module configuration information from the Administration Server.

`WLESscm.sh start`
or
`WLESscm.sh console`
	Starts the Service Control Manager as a daemon process.
	Note: Either command provides the same result.

`WLESscm.sh stop`	Stops the Service Control Manager. The SCM can also be stopped by closing the console window or pressing Ctrl+C.
`WLESarme.sh refresh`	Updates the SSM instance to include the most recent policy data from the Administration Server.

`WLESarme.sh start`
or
`WLESarme.sh console`
	Either command starts the SSM instance as a daemon process.
	Note: Either command provides the same result.

`WLESarme.sh stop`	Stops the SSM instance. You may also close the console window or press `Ctrl+C`.

SCM Start-Up Option on Linux

To allow the SCM to start up after a reboot on Linux, set it to start on runlevel3 (non-graphical runlevel) and runlevel5 (graphical runlevel). To do this, run the following command as root:

```
chkconfig --level 35 WLESscm on
```

To check the runlevel of the Service Control Manager, run:

```
chkconfig --list WLESscm
```

CHAPTER 3

Configuring SSL for Production Environments

ALES uses an implementation of the Transport Layer Security (TLS) 1.0 specification, also referred to as SSL. The server (WebLogic Server 8.1 or Tomcat) hosting ALES supports TLS on a dedicated listening port that defaults to 7010. To establish a secure connection, a client (Web browser or Java application) connects to the Administration Server by supplying the port and the secure address (HTTPS) in the connection URL, e.g., `https://myserver:7010`. The Administration Server returns a certificate to identify itself to the client.

When you install ALES, demonstration certificates are provided and configured automatically for working in a development environment. However, it is very important that these certificates not be used in a production environment.

Secure Sockets Layer (SSL) is described in the following sections.

SSL Basics

Basic information about SSL and ALES is contained in the following sections.

Private Keys, Digital Certificates, and Trusted Certificate Authorities

Private keys, digital certificates, and trusted certificate authorities establish and verify server identity. SSL uses public key encryption for authentication. With public key encryption, a public key and a private key are generated for a server. Data encrypted with the public key can only be decrypted using the corresponding private key and vice versa. The private key is carefully protected so that only the owner can decrypt messages that were encrypted using the public key.

The public key is embedded within digital certificate along with additional information describing the owner of the public key, such as name, street address, and e-mail address. A private key and digital certificate provide identity for the server.

The data embedded in a digital certificate is verified by a certificate authority (CA) and is digitally signed with the digital certificate of the certificate authority. Well-known certificate authorities include Verisign and Entrust. The trusted CA certificate establishes trust for a certificate.

Web browsers, servers, and other SSL-enabled applications generally accept as genuine any digital certificate that is signed by a trusted certificate authority and is otherwise valid. For example, a digital certificate can be invalid because it has expired, or the digital certificate of the CA used to sign it expired, or because the host name in the digital certificate of the server does not match the URL specified by the client.

One-Way SSL versus Two-Way SSL

You can configure SSL to use either one-way or two-way authentication:

One-way SSL

To establish an SSL connection, the server must present a certificate to the client, but the client is not required to present a certificate to the server. To successfully negotiate an SSL connection, the client must authenticate the server, but the server accepts any client into the connection. One-way SSL is common on the Internet where customers want to create secure connections before sharing personal data. Often, clients use SSL to log on so that the server can authenticate them. By default, the Administration Server is configured for one-way SSL using demo certificates.

Two-Way SSL

To establish the SSL connection, the server must present a certificate to the client and the client must also present a certificate to the server. ALES can be configured to require clients to submit valid and trusted certificates before completing the SSL connection.

Keystores

A keystore is a mechanism designed to create and manage private key/digital certificate pairs and trusted CA certificates.

All private key entries in a keystore are accessed through unique aliases and password that is specified when creating the private key in the keystore. The default alias for the ALES Administration Server certificates is `ales-webserver`. **Note:** Aliases are case-insensitive; the aliases `Hugo` and `hugo` would refer to the same keystore entry.

ALES explicitly trusts all certificate authorities whose entries are found in the keystore configured as trusted. Although ALES does not use the alias to access trusted CA certificates, the keystore does require an alias when loading a trusted CA certificate into the keystore.

Upon installation, two keystores are used to establish trust between the Administration Server and clients:

- `Webserver.jks`— The keystore is located in the Administration Servers `ssl` directory. It contains:
  - a demonstration private key for the Administration Server.

  - the identity for the Administration Server in a public certificate that is signed by a trusted ALES Demo CA and bound to the server's hostname.
  - a public certificate for the ALES Demo CA itself.

- `DemoTrust.jks`— This keystore is located in the `ssl` directory of the SSM or SCM instance. It used by enrollment clients when connecting from an SSM or SCM instance. It contains the public certificate of the same trusted ALES Demo certificate authority that is in `webserver.jks`. This keystore is used when running `enroll.bat/sh` (for SSM) or `enrolltool.bat/sh` (for SCM) with the `demo` argument. When using the `secure` argument, the SSM enroller uses `$JAVA_HOME/lib/security/cacerts`, while the SCM uses its own trust.jks keystore.

For descriptions of common keytool commands, see "Keytool Utility'" on page 3-9.

How the Administration Server Establishes Trust

The client types connecting to the Administration Server are: (1) Internet Explorer browsers accessing the administration console, and (2) SSM enrollment clients. The method used to establish trust depends on the client type.

- Internet Explorer browsers. Browser clients will not have the ALES demo certificate or demo CA certificate in the trusted store, so a security alert window will display when accessing the administration console. The user can use the window to trust the Administration Server's demo certificate. **Note:** The alert window does not display when the Administration Server is configured to use a valid signed certificate.
- SSM and SCM enrollment clients. An enrollment client uses its DemoTrust.jks keystore to establish trust. When the client tries to enroll, the Administration Server presents its public certificate for verification to the enrollment client. The client will trust the certificate, because the DemoTrust.jks keystore that it is using in "demo" mode has the same ALES Demo CA certificate.

The important thing to remember when updating certificates is that the server and client both trust a common CA.

Configuring SSL

To configure SSL for a production environment you must create a keystore to replace `Webserver.jks` and configure the Administration Server to use it. Then you may configure ALES to use one-way or two-way SSL.

- "Create a Keystore and Load Signed Certificates" on page 3-5

- “Configuring One-Way SSL” on page 3-6
- “Configuring Two-Way SSL” on page 3-7

Procedures described in this section make use of Sun’s keytool utility. For information about this tool, see “Keytool Utility‘” on page 3-9.

Create a Keystore and Load Signed Certificates

1. Create the keystore and private key as follows:
   a. Create a `secureWebserver.jks` keystore and generate the private key using keytool utility as follows:

      ```
      keytool -genkey -alias ales-webserver -keyalg RSA -keystore
      secureWebserver.jks
      ```

   b. When prompted, enter the keystore password and general information about the certificate, (company, contact name, etc.). This information is displayed to users who attempting to access a secure page in the application.
   c. When prompted for the key password, enter the same password used for the keystore itself. This can be accomplished by pressing ENTER.
2. Create a Certificate Signing Request (CSR) as follows:
   a. Create `certreq.csr` by entering:

      ```
      keytool -certreq -alias ales-webserver -keyalg RSA -file certreq.csr
      -keystore secureWebserver.jks
      ```

   b. Submit `certreq.csr` to the Certificate Authority.
3. Import the certificate into the keystore as follows:
   a. Download a Chain Certificate from the Certificate Authority. Then import it into the keystore using the following command:

      ```
      keytool -import -alias cacerts -keystore secureWebserver.jks
      -trustcacerts -file <filename_of_the_chain_certificate>
      ```

   b. Import the new certificate using the following command.

      ```
      keytool -import -alias ales-webserver -keystore secureWebserver.jks
      -trustcacerts -file <your_certificate_filename>
      ```

   c. **For SCM only**, add the CA's certificate chain to `trust.jks` in the `ssl` directory.

```
keytool -import -alias cacerts -keystore trust.jks -trustcacerts -file
<filename_of_the_chain_certificate>
```

Configuring One-Way SSL

The procedure for configuring the new keystore (`secureWebserver.jks`) for production use on depends on the type of server hosting ALES. This section provides instructions for WebLogic Server and Tomcat.

Configure One-Way SSL on WebLogic Server

Perform the following steps to use the secure the keystore when using WebLogic Server.

Note: Go to http://e-docs.bea.com/wls/docs81/secmanage/ssl.html for other ways to do this via the WLS Administration console.

1. Copy `secureWebserver.jks` to the `ssl` directory where the Administration Server is installed (the default is `BEA_HOME\ales21-admin\ssl)`.
2. Modify the server's configuration file (`BEA_HOME/asiDomain/config.xml`) as follows.
   a. Replace every occurrence of `webserver.jks` appears with `secureWebserver.jks`.
   b. Change the `ServerPrivateKeyAlias` attribute to match the alias that is assigned to the certificate in the `secureWebserver.jks` keystore. In the example above it was `ales-webserver`.
   c. Change the `ServerPrivateKeyPassPhrase` attribute to match the password for the `secureWebserver.jks` keystore.
3. Restart the Administration Server.

   After performing these steps, running `enroll.bat/sh` (for a SSM) or `enrolltool.bat/sh` (for a SCM) will pass in `secure` instead of `demo` as an argument.

Configure One-Way SSL on Tomcat

Perform the following steps to use the secure the keystore when using WebLogic Server.

Note: Go to http://tomcat.apache.org/tomcat-5.0-doc/ssl-howto.html for more information about SSL under Apache Tomcat.

1. Copy `secureWebserver.jks` to the ssl directory where the Administration Server is installed (the default is `BEA_HOME\ales21-admin\ssl)`.
2. Modify the server's configuration file (`TOMCAT_HOME/config/server.xml`) as follows.

a. Replace every occurrence of `webserver.jks` with `secureWebserver.jks`.

b. Add `keystorePass=<your_password>` next to the `keystoreFile` attribute.

3. Restart the Administration Server.

After performing these steps, running `enroll.bat/sh` (for a SSM) or `enrolltool.bat/sh` (for a SCM) will pass in `secure` instead of `demo` as an argument.

Configuring Two-Way SSL

The procedure for configuring the new keystore (`secureWebserver.jks`) for two-way SSL depends on the type of server hosting ALES. This section provides instructions for WebLogic Server and Tomcat.

Configure Two-Way SSL on Weblogic Server

To configure the Administration Server for two-way SSL on WebLogic server:

1. Configure one-way SSL as described in "Configuring One-Way SSL" on page 3-6.
2. Log in to the WebLogic Administration Console.
3. Expand the Servers node and select name `adminserver`.
4. Select the Configuration-->Keystores and SSL tab.
5. Click the Show link under Advanced Options.
6. In the Server attributes section of the window, set the Two-Way Client Cert Behavior attribute. The available options are shown in Table 3-1.

Table 3-1 Two Way SSL Cert Behavior Options

Option	Description
Client Certs Not Requested	The default (meaning one-way SSL).
Client Certs Requested But Not Enforced	Requires a client to present a certificate. If a certificate is not presented, the SSL connection continues.
Client Certs Requested And Enforced	Requires a client to present a certificate. If a certificate is not presented or if the certificate is not trusted, the SSL connection is terminated.

7. Click Apply.

After performing these steps, running `enroll.bat/sh` (for a SSM) or `enrolltool.bat/sh` (for a SCM) will pass in `secure` instead of `demo` as an argument.

Configure Two-Way SSL on Apache Tomcat

To configure the Administration Server for two-way SSL on WebLogic server:

1. Configure one-way SSL as described in "Configuring One-Way SSL" on page 3-6.
2. Open `TOMCAT_HOME/config/server.xml` in a text editor and set the value of `clientAuth` as follows.

Value	Description
false	When set to 'false', Tomcat will NOT require all SSL clients to present a client Certificate in order to use this socket. (1-way SSL)
want	Tomcat will request a client Certificate, but not fail if one isn't presented. (Optional 2-way SSL)
true	Tomcat will require all SSL clients to present a client Certificate in order to use this socket. (Mandatory 2-way SSL)

After performing these steps, running `enroll.bat/sh` (for a SSM) or `enrolltool.bat/sh` (for a SCM) will pass in `secure` instead of `demo` as an argument.

Keytool Utility‘

Sun Microsystem’s keytool utility is included in JDK installations. For complete information about this tool, consult the Sun Microsystems website. See also detailed command usage options for Windows (http://java.sun.com/j2se/1.4.2/docs/tooldocs/windows/keytool.html) and Solaris/Linux (http://java.sun.com/j2se/1.4.2/docs/tooldocs/solaris/keytool.html).

When using the keytool utility, observe the following:

- The keytool utility does not allow you to import existing private keys into the keystore.
- When using the keytool utility, the default key pair generation algorithm is DSA. Specify another key pair generation such as RSA algorithm when using ALES.
- ALES currently operates only on JKS keystores. The JKS format is Java's standard keystore format and is the format created by the keytool command-line utility.

Table 3-2 shows the keytool commands to use when creating and using JKS keystores.

Table 3-2 Common Keytool Commands

Command	Description
keytool -genkey -alias aliasforprivatekey -keystore keystorename -storepass keystorepassword	Generates a new private key entry and self-signed digital certificate in a keystore. If the keystore does not exist, it is created.
keytool -import -alias aliasforprivatekey -file certificatefilename.pem -keypass privatekeypassword -keystore keystorename -storepass keystorepassword	Updates the self-signed digital certificate with one signed by a trusted CA.
keytool -import -alias aliasfortrustedca -trustcacerts -file trustedcafilename.pem -keystore keystorename -storepass keystorepassword	Loads a trusted X.509 CA certificate or PKCS#7 certificate chain into a keystore. If the keystore does not exist, it is created.
keytool -certreq -alias aliasforprivatekey -sigalg RSA -file certreq_file -keypass privatekeypassword -storetype keystoretype -keystore keystorename -storepass keystorepassword	Generates a CSR (using the PKCS#10 format) to be sent to a trusted CA. The trusted CA authenticates the certificate requestor and returns a digital certificate, which replaces the existing self-signed digital certificate in the keystore.
keytool -list -keystore keystorename	Displays what is in the keystore.

Table 3-2 Common Keytool Commands

Command	Description
keytool -delete -alias aliasforprivatekey -keystore keystorename -storepass keystorepassword -alias privatekeyalias	Delete a private key/digital certifcate pair for the specified alias from the keystore.
keytool -help	Provides online help for keytool.

CHAPTER 4

Failover and System Reliability

This section describes features of AquaLogic Enterprise Security that support recovery from failure. It covers the following topics:

Understanding Failover

In general, failover is the ability of a product to detect a failure for a particular component and switch to a working replica of that component without losing functionality. ALES support two failover scenarios:

- Runtime failover – makes sure that an ALES SSM continues to provide security services even if external components it relies on (such as the authentication database, for example) become unavailable during runtime. This assures runtime availability. This failover mechanism is achieved by configuring secondary sources of information for ALES security providers. See Figure 4-1 for an illustration of failover during runtime of an SSM.
- Administration time failover – makes sure that ALES administration services are accessible even if the primary ALES Administration Server fails. This failover is handled by

configuring a secondary Administration Server. The secondary server is the redundant one and should be accessed if the primary one cannot be used. Figure 4-2 and Figure 4-3 show how ALES supports administration failover to a secondary Administration Server.

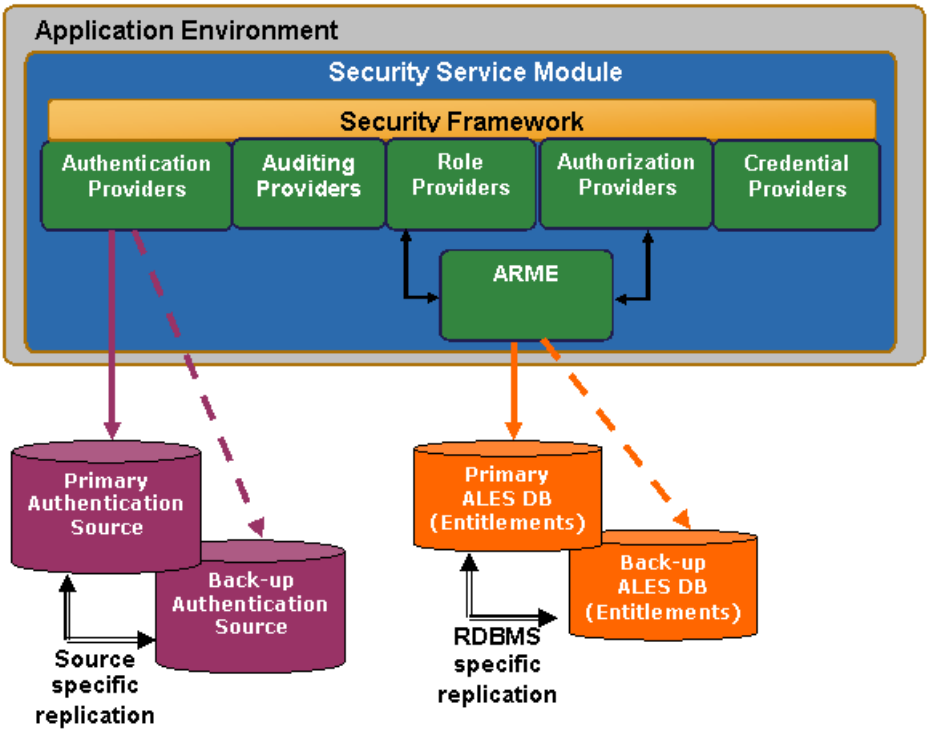
Assuring Runtime Availability for SSMs

ALES security providers depend on data stores for authentication, authorization, and credential mapping. You can configure ALES for failover in these three important cases:

- Authentication failover is provided by configuring the SSM to point to primary and secondary user data stores. The replication of these data stores is handled by the native functionality of the data store, such as:
  - database replication for a relational database system
  - LDAP master/slave configuration
  - primary and secondary domain controllers in a Windows NT doman
- Credential mapping failover is provided by configuring the ALES Database Credential Mapper to primary and secondary databases.

The ALES SSMs have no runtime dependency on the Administration Server.

Figure 4-1 Runtime Availability

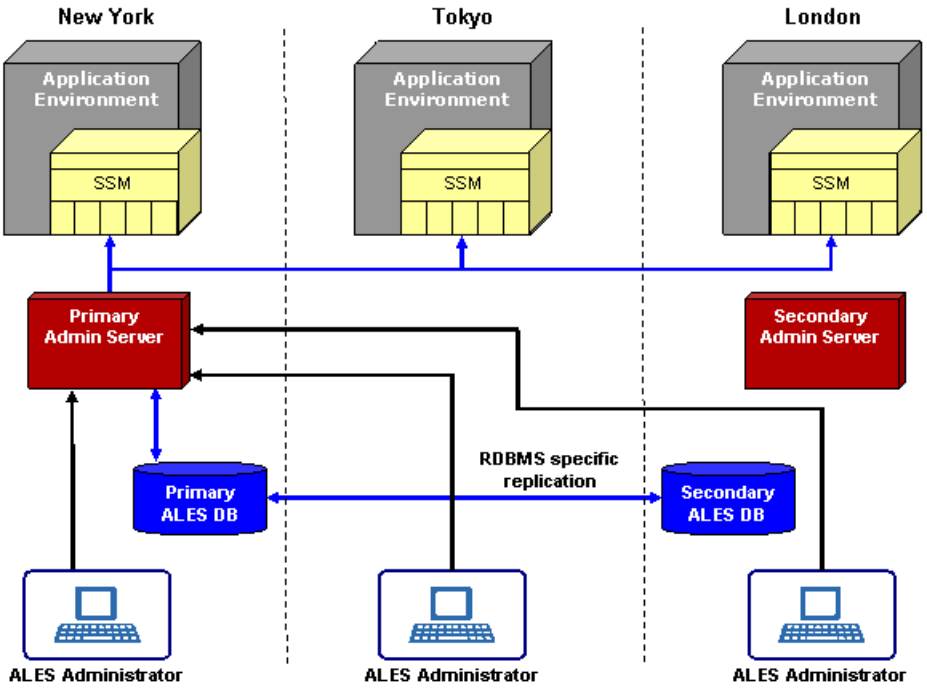


Assuring Administrative Availability

You can provide failover capability for ALES administration functions by installing redundant Administration Servers: a primary and a secondary. The secondary Administration Server is used only when the primary becomes unavailable.

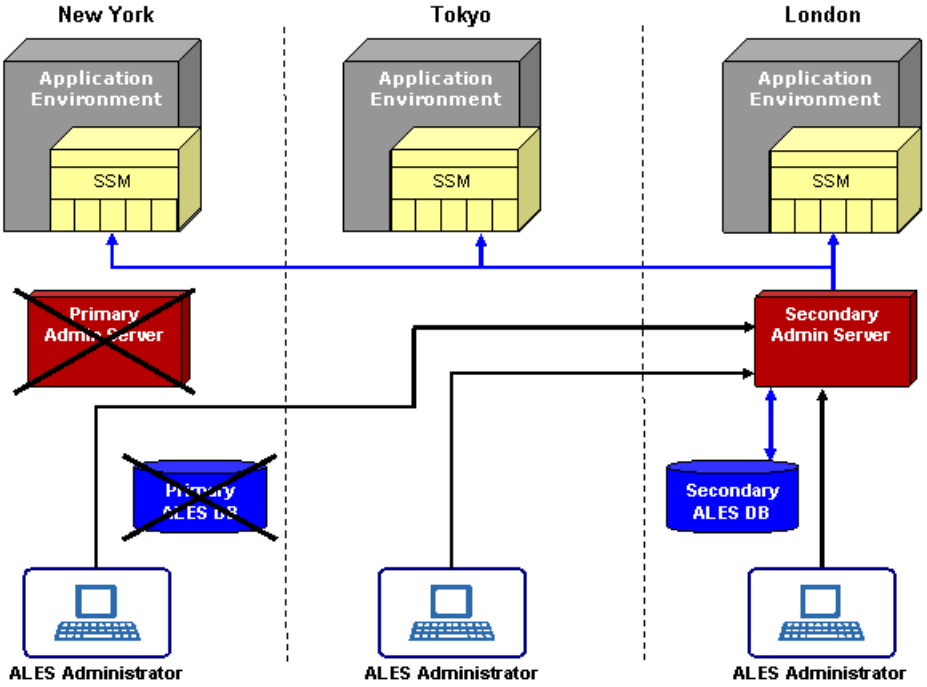
For example, consider the global deployment illustrated in Figure 4-2. In this case, the enterprise has applications staged on servers in New York, Tokyo and London. The enterprise has deployed redundant ALES Administration Servers in its New York and London data centers as well as a replicated database to store ALES policies and entitlements information. Under normal conditions, administrators interact with the primary Administration Server in New York only. When policies are updated, the Administration Server pushes the changes to all the SSMs in the global environment.

Figure 4-2 Administrative Availability (Working Normally)



Now consider the case when the data center in New York goes down, illustrated in Figure 4-3. The SSMs detect that the primary Administration Server is down and connect to the secondary Administration Server. The secondary Administration Server detects that the primary database is down and connects to the secondary database server (the replica).

Figure 4-3 Administrative Availability (After Failure)



One benefit of the ALES architecture is that even if all the Administration Servers go down (either for maintenance or due to failure), including the secondary Administration Servers, there is no impact on the applications in production or on the security services provided by those Security Service Modules and providers that you have configured. You cannot install or enroll new Security Service Modules until the primary Administration Server is running or you have reconfigured the secondary server as the primary. You can only enroll Security Service Modules using a primary Administration Server. When the primary database is available, it will be used by both the primary and secondary Admininstration Servers.

For information on how to configure the Administration Server for failover, see "Setting up Administration Servers for Failover" on page 4-9.

Failover Considerations for the Database Server

Figure 4-3 shows how the logical view of failover functionality when the primary database server fails. The number of redundant database servers you configure can vary; however, a minimum of two is recommended to maintain reliable services. It is up to the system administrator to set up database failover and configure data replication between the database instances.

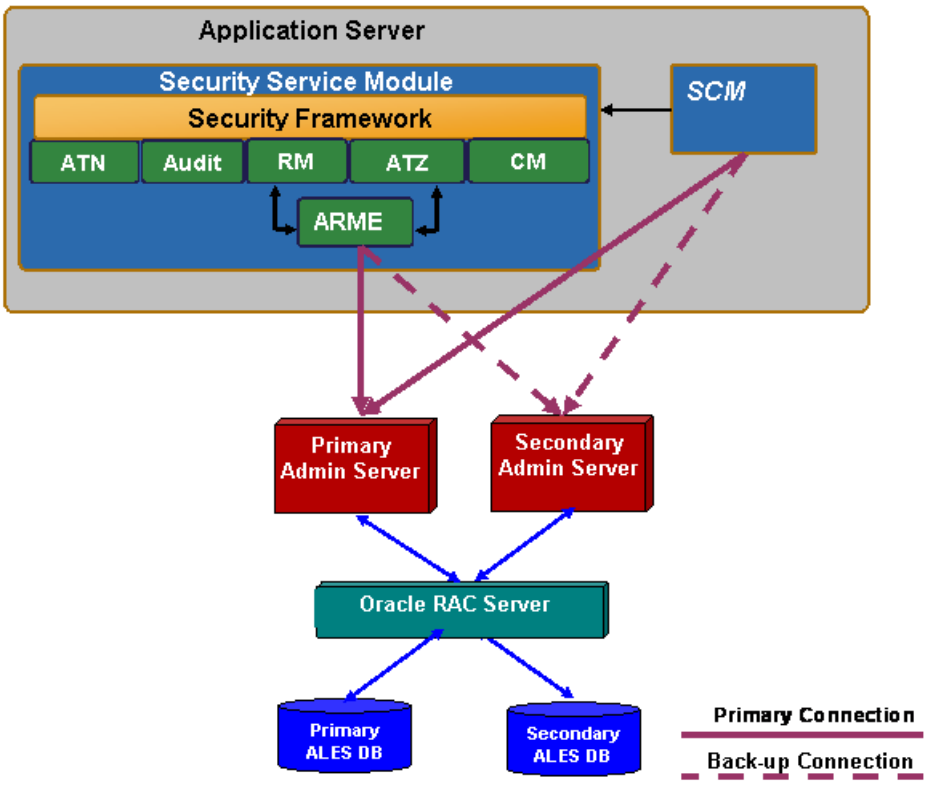
Because the database server contains all of the configuration and security data used by the Administration Application, to protect your applications and resources, you want to make sure it is highly available and reliable. This can be accomplished by implementing recommendations from your database manufacturer (for example, through the use of clustering architecture or hot standby).

There are two approaches for making sure that two instances of ALES database contain the same data:

- Use Oracle RAC for Oracle databases (see Figure 4-4) or a similar approach recommended by the database vendor. This approach allows the ALES providers to be configured with only one address, assuming transparent failover for the database is provided by the database vendor.
- Use the replication mechanism recommended by the database vendor. In this case, you set up a primary database and secondary database with unique connection information. The connection information for the secondary database can be added the ALES Database provider and also the ALES Database Credential Mapper. You configure this conection information in the Administration Console on the Failover tabs.

Figure 4-4 illustrates the failover mechanism for the ALES Administration Server using Oracle RAC.

Figure 4-4 ALES Administration Servers using Oracle RAC



Common methods of archiving high availability include periodic back-ups, fault tolerant disks, and copying files manually whenever they are changed. This is also the case for any optional external data sources you have configured. Both Sybase and Oracle offer database backup methods. Refer to Sybase and Oracle documentation for details. A database backup can be used for database recovery in the case of disk failure.

Failover Considerations for a Security Service Module

You can use the Administration Console to configure failover support for database-related providers and LDAP authentication providers. Configuration for database-related providers

includes the specification of the secondary database and support for LDAP authenticators includes the specification of the secondary LDAP server.

The following providers support configuration of a secondary database:

- Database Credential Mapping provider
- Database Authentication provider
- ASI Authorization provider
- ASI Role Mapper provider

The ASI Authorization Provider contacts an external process to evaluate its authorization queries. If that process dies, the ASI Authorization provider denies access to all resources. The ASI Authorization provider can be configured to contact the Administration database to retrieve subject attributes and group membership for use in authorization and delegation decisions. If the database connection fails, the provider connects to the configured secondary database. The provider tries to reconnect to the failed database after a configurable time-out. If all database connections fail and defined policies operate on user attributes and group membership, all access is denied.

The following providers support configuration of a secondary LDAP server:

- Novell LDAP Authenticator
- Active Directory Authenticator
- iPlanet Authenticator
- Open LDAP Authenticator

The NT Authenticator already supports multiple domain controllers. The WebLogic Authenticator, WebLogic Authorizer and WebLogic Role Mapper use the internal LDAP server for WebLogic server as its data store. No support for a redundant source is required.

Failover Considerations for a Service Control Manager

When the Security Control Manager server starts, it contacts an Administration Server to make sure that it is using the latest version of configuration data. When configuration data is received by the SCM, it is cached locally. When configuration data is modified, the Administration Server pushes the updates to the SCM. Failover for the SCM server is implemented as follows:

1. You can configure the SCM with addresses for primary and secondary Administration Servers. During installation, you can provide the address for a secondary Administration Server. After installation, you can set the addresses in the *SCM_HOME*/config/SCM.properties. The domain.asi.primary.pdurl property points to the primary Administration Server and the domain.asi.secondary.pdurl property points to the secondary Administration Server. By default, if you did not provide secondary admin server information during install, both of these properties point to the current Administration Server installed when the SCM was installed.

2. If no Administration Server is available, the SCM continues to operate using the previously cached set of policies and configuration data. If the SCM is coming up for the first time or does not have a cache then it will stay up and continue looking for an Administration Server to connect to. Once a primary or secondary Administration Server is available, the SCM will get its configuration data and cache it.

Setting up Administration Servers for Failover

You can install two Administration Servers: a primary and a secondary. The secondary Administration Server is used for the purpose of failover when the primary becomes unavailable. The order in which the Administration Servers are installed is not important; using the Administration Console, you designate which one is suppose to be the primary and which one is the secondary. See "Configure the Secondary Server Trust Synchronization Mechanism" on page 4-11.

When an Administration Server is installed, a set of unique certificates is generated for it. Common trusted certificates enable SSMs and SCMs to connect by 2-way SSL. To enable failover, the trust stores of the primary and secondary Administration Servers need to be synchronized and also periodically kept synchronized when additional SSMs and SCMs are enrolled. The following sections describe how to set up and configure Administration Server trust synchronization.

- "Initialize the Secondary Server Trust Stores" on page 4-10
- "Configure the Secondary Server Trust Synchronization Mechanism" on page 4-11

Installing the Secondary Administration Server

The secondary Administration Server must be set up in the same manner as the primary Administration Server. It should be installed on a separate machine from the one on which the primary Administration Server has been installed.

1. Install the servlet container that will host the Admin web application, WebLogic Server 8.1 or Apache Tomcat.
2. Run the admin installation program to install the secondary Administration Server.
3. Enter the following information:
   a. When prompted for the Enterprise Domain, make sure to enter the same domain name that you entered during the primary installation (default is `asi`).
   b. When prompted for the Secondary Server URL, leave this blank.
   c. When prompted for the Database Configuration, make sure to use the exact same database username that you specified during the primary installation.
   d. The database passwords used by the primary and secondary Administration Servers should be identical to each other.
   e. It is recommended that you use the same passwords you used to install the primary Administration Server; however, you can use instance specific passwords to protect the various sensitive artifacts on the Administration Servers. For example, you may use different key passwords for the CA, Admin, SSM, and SCM identities that you entered in the primary Administration Server installation. The same applies to the Identity, Peer, and Trust key store passwords.

   Note: Do not install the database schema at the conclusion of the secondary Administration Server installation process.
4. Follow the steps described in "Initialize the Secondary Server Trust Stores" on page 4-10.
5. Start the secondary Administration Server just as you normally would start a primary Administration Server.
6. Follow the steps described in "Configure the Secondary Server Trust Synchronization Mechanism" on page 4-11.

Initialize the Secondary Server Trust Stores

Before starting the secondary Administration Server, you must synchronize the various trust stores used by the secondary Administration Server with those of the primary. If this is not done, the secondary Administration Server will not trust the SSMs and SCMs currently enrolled with the primary Administration Server, and as a result, there can be problems during failover.

To initialize the secondary Administration Server trust stores:

1. On the secondary Administration Server, create the following directories:
   - *ALES_HOME*/primary-admin-ssl
   - *ALES_HOME*/primary-scm-ssl
2. Copy the /ssl directory from the primary Administration Server to the secondary Administration Server machine into the *ALES_HOME*/primary-admin-ssl directory.
3. Copy the /ssl directory from the primary Service Control Manager to the secondary Administration Server machine into the *ALES_HOME*/primary-scm-ssl directory.
4. From the /bin directory of the secondary Administration Server installation, execute the initialize_backup_trust.bat (on Windows platforms) or initialize_backup_trust.sh (on UNIX platforms) command. When prompted for the primary Service Control Manager SSL directory, enter the path to the *ALES_HOME*/primary-scm-ssl directory. Likewise, when prompted for the primary Admin SSL directory, enter the path to the *ALES_HOME*/primary-admin-ssl directory.

Configure the Secondary Server Trust Synchronization Mechanism

Even though the secondary Administration Server trust stores are synchronized with those of the primary Administration Server when you complete the procedure described in "Initialize the Secondary Server Trust Stores" on page 4-10, it is possible for them to become out-of-sync over time. This happens when a new SSM or SCM is enrolled with the primary Administration Server. The trust stores of the primary Administration Server are updated with the new SSM or SCM certificate during enrollment, but since enrollment happens only with the primary Administration server, the secondary Administration Server trust stores do not have the new certificates. A similar trust situation occurs when an SSM or SCM is un-enrolled.

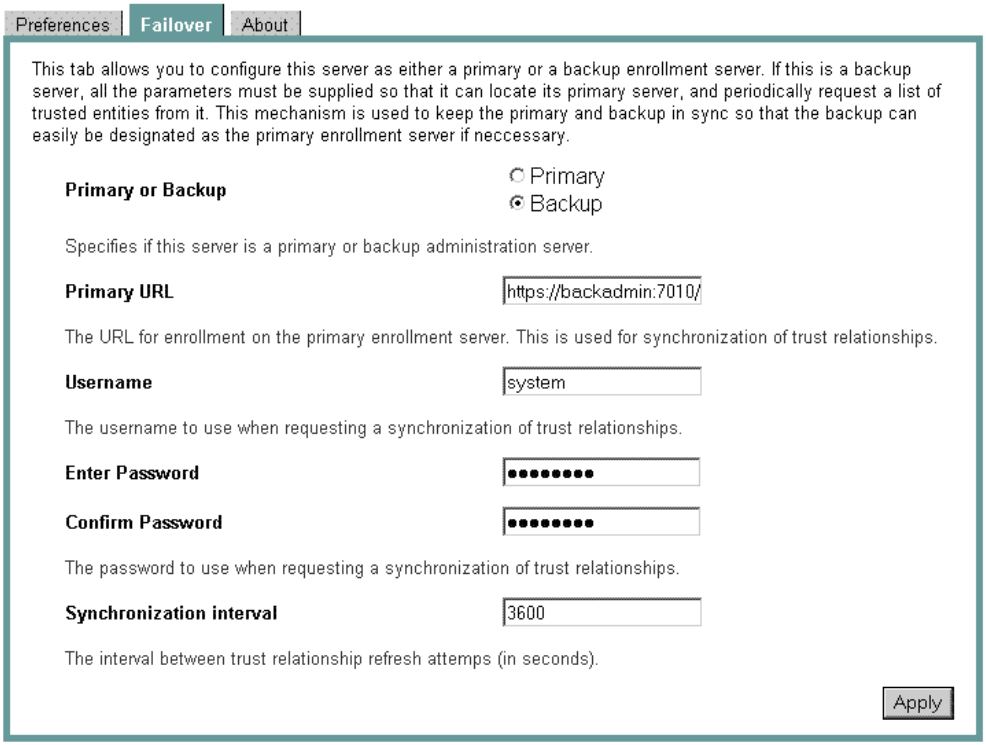
To prevent the trust stores from becoming unsynchronized, the Administration Server has a trust synchronization mechanism that should be enabled on the secondary Administration Server. The trust synchronization mechanism on the secondary Administration Server periodically polls the primary Administration Server for any updates to its trust store, and if a change has occurred, the mechanism updates the secondary Administration Server's trust store with the contents of the primary. It is very important that you enable the trust synchronization mechanism only on the secondary Administration Server.

To configure the secondary Administration Server for trust synchronization:

1. In the Administration Console, click on Administration Console at the top of the navigation tree and then select the Set Console Preferences page.

2. Click the Failover tab.

Figure 4-5 Configuring a Backup Admin Server in the Administration Console



On the Failover tab, you can configure this Administration Server as either a primary or a secondary (backup) Administration Server. In case of the secondary server, you must specify the parameters that permit the secondary Administration Server to locate the primary server and periodically request a list of trusted entities. This mechanism keeps the trust stores of the primary and secondary Administration Servers synchronized. If this is a primary server, you don't need to do anything except ensuring that the Primary option is checked.

3. Select Backup.

4. In the Primary URL text box, enter the URL of the primary Administration Server. This URL is used to synchronize a trust relationship. The URL is the same URL used to access the Administration Console in the primary Administration Server.

5. In the Username text box, enter the admin username (default is “system”).
6. In the Enter Password and Confirm Password text boxes, enter the password for the admin user.
7. In the Synchronization interval text box, enter the number of seconds between attempts of trust relationship synchronization. The value for this setting depends on how frequently SSM or SCM instances are enrolled and un-enrolled from the primary Administration Application in your environment.
8. Click Apply.

<~runChNum>