BEA AquaLogic Enterprise Security™®

Administration and Deployment Guide

Version 2.1
Document Revised: December 19, 2005

Copyright

Restricted Rights Legend

Third-Party Software License Agreement

Sun Microsystems, Inc.'s XACML implementation v2.0

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,

CONSEQUENTIAL, 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

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

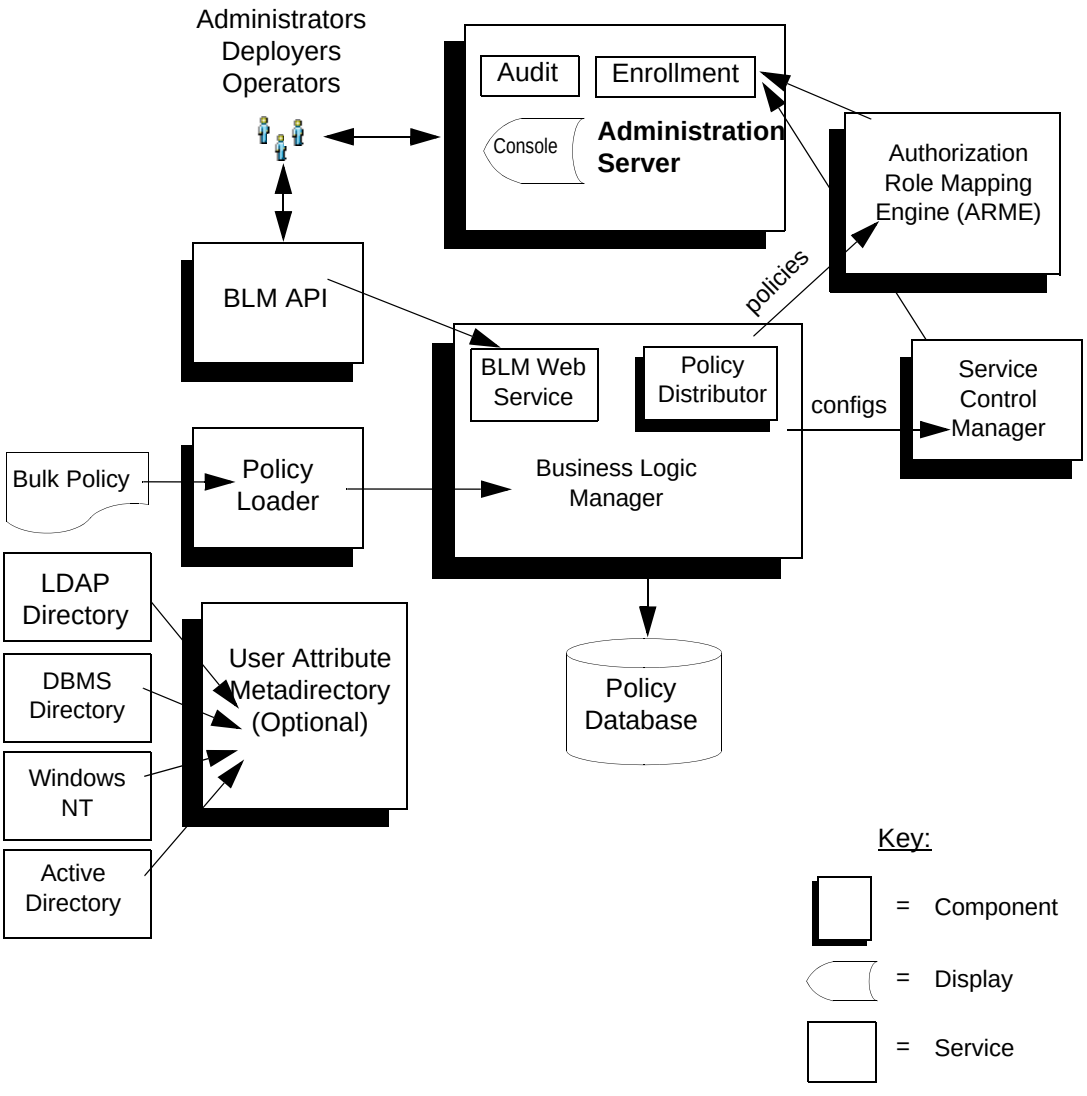
ALES Architecture

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.

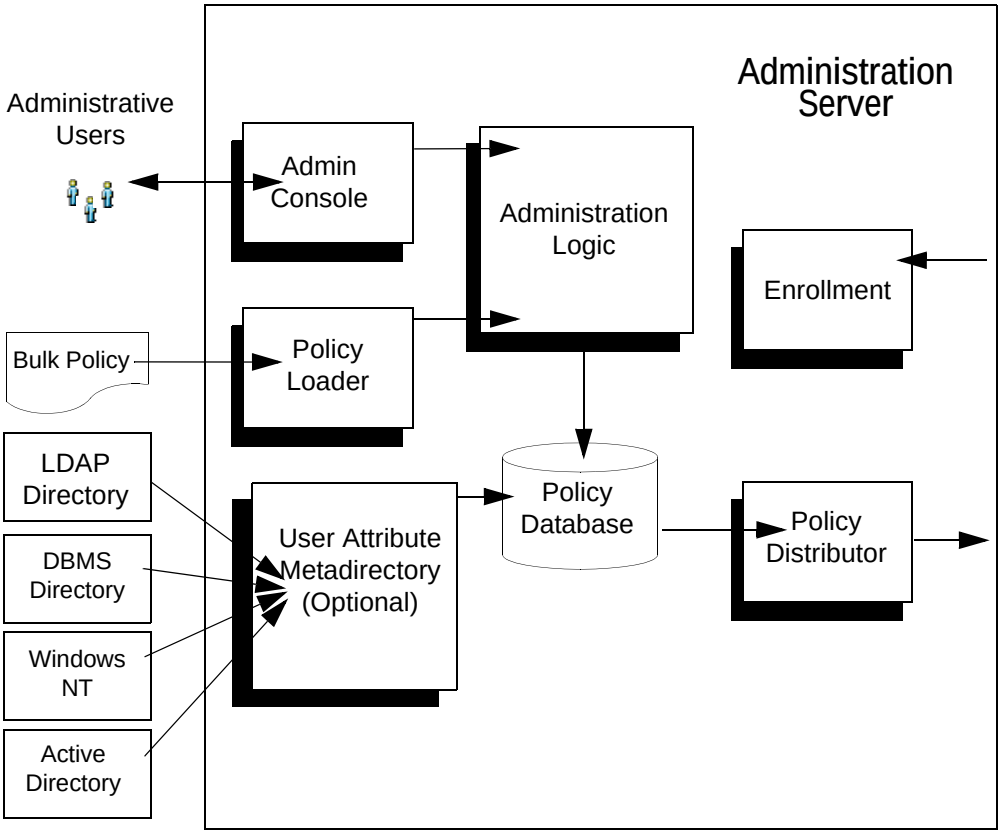
Policy Loader–Imports policy data from an external file. The external file can be generated by another system or another Administrative Server, or it can be manually coded. For additional information on how to use the Policy Loader, see the *Policy Managers Guide*.

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



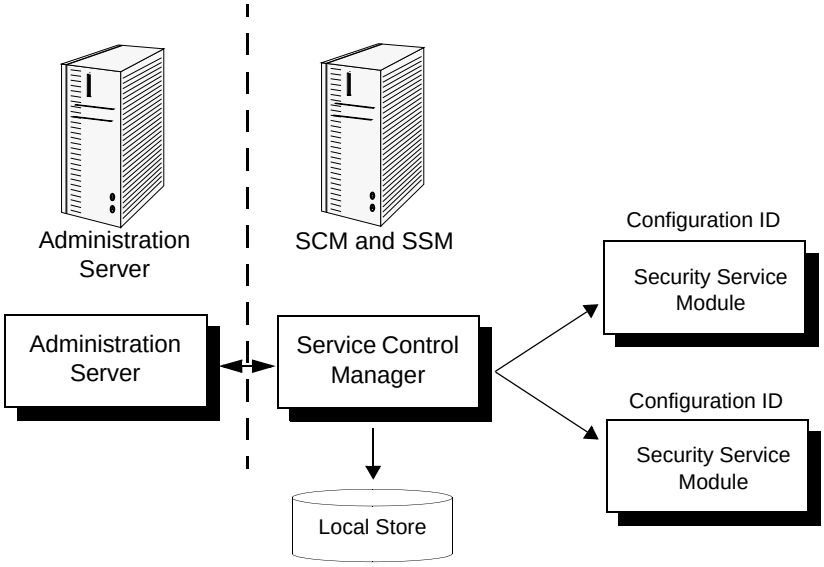
Service Control Manager (SCM)

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

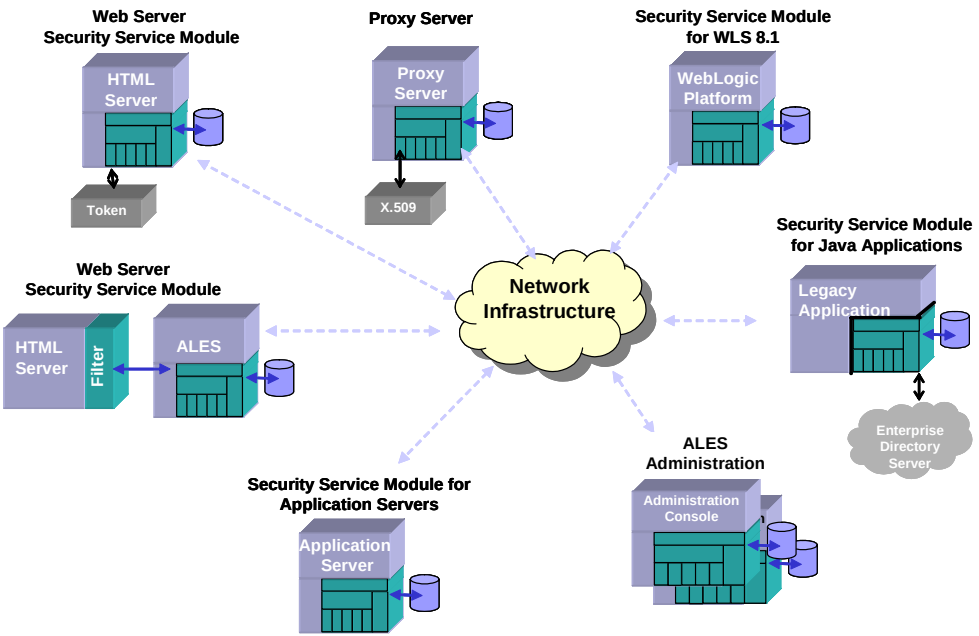
An ALES environment can consist of a single or multiple instances of the Administration Server, one or more Service Control Managers (hosted on individual machines), and any number of Security Service Modules, each associated with an SCM. Each Security Service Module may share or use different configuration 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.

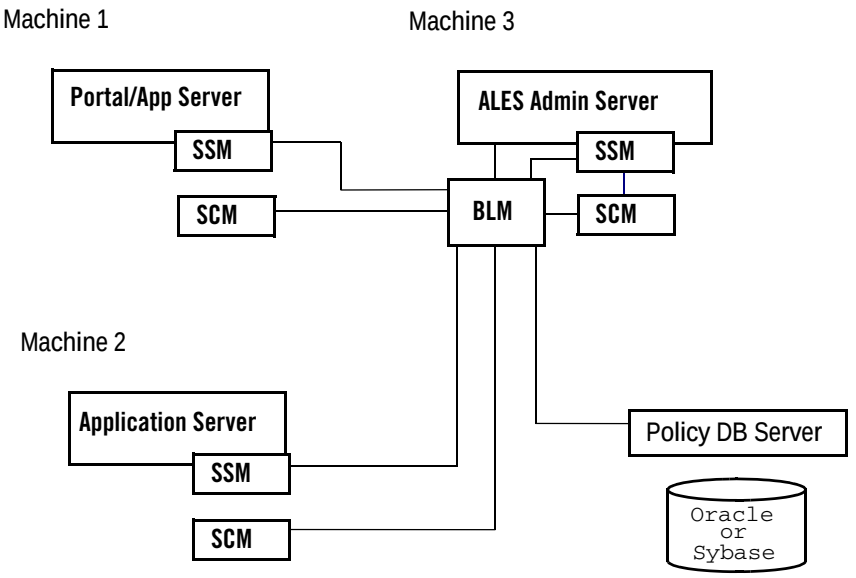
Figure 1-4 Distributed Computing Security Infrastructure



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" on page 3-2

"One-Way SSL versus Two-Way SSL" on page 3-3

"How the Administration Server Establishes Trust" on page 3-4

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 ALES certificates is `ales-webserver`. **Note:** Aliases are case-insensitive; the aliases `Hugo` and `hugo` would refer to the same keystore entry.

All certificate authorities in a keystore identified as trusted by ALES are 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 ALES in a certificate that is signed by a trusted BEA Demo CA, and the BEA Demo CA itself. It also contains the hostname name.

- `DemoTrust.jks`— This keystore is located in the SSM instance's ssl directory and is used by enrollment clients when they connect from an SSM instance or SCM. It contains the same trusted BEA Demo certificate authority that is in `webserver.jks`. This keystore is used when the "demo" argument is used when running "enroll.bat/sh". When using "secure" argument it uses the JAVA_HOME/lib/security/cacerts keystore instead.

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--because browser clients will not have the ALES demo certificate or demo CA certificate in the trusted store, 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:** This windows does not display when the Administration Server is configured to use a valid signed certificate.
- SSM enrollment clients--An SSM enrollment client uses its `DemoTrust.jks` keystore to establish trust. When the SSM client tries to enroll, the Administration Server presents its public certificate for verification to the SSM client. The client will trust the certificate because the DemoTrust.jks keystore that it is using has the same 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 3-5
- "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 secureKeystore.jks
   -trustcacerts -file <your_certificate_filename>
   ```

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.

   Now when you run "enroll.bat/sh" tool from an SSM instance pass in the argument "secure" instead of "demo".

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 the "enroll.bat/sh" tool from an SSM instance will pass in the argument "secure" instead of "demo".

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 3-5.
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 the "enroll.bat/sh" tool from an SSM instance will pass in the argument "secure" instead of "demo".

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 3-5.
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 the "enroll.bat/sh" tool from an SSM instance will pass in the argument "secure" instead of "demo".

Keytool Utility‘

Sun Microsystem’s keytool utility is included in JDK installations. For complete information about this tool, consult the Sun Microsystems website. 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 -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 privatekeyfilename.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 CA certificate into a keystore. If the keystore does not exist, it is created.
-certreq -alias alias -sigalg sigalg -file certreq_file -keypass privatekeypassword -storetype keystoretype -keystore keystorename -storepass keystorepassword	Generates a CSR, using the PKCS#10 format. Sent the CSR to be sent to a trusted CA. The trusted CA authenticates the certificate requestor and returns a digital certificate to replace the existing self-signed digital certificate in the keystore.
keytool -list -keystore keystorename	Displays what is in the keystore.
keytool -delete -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

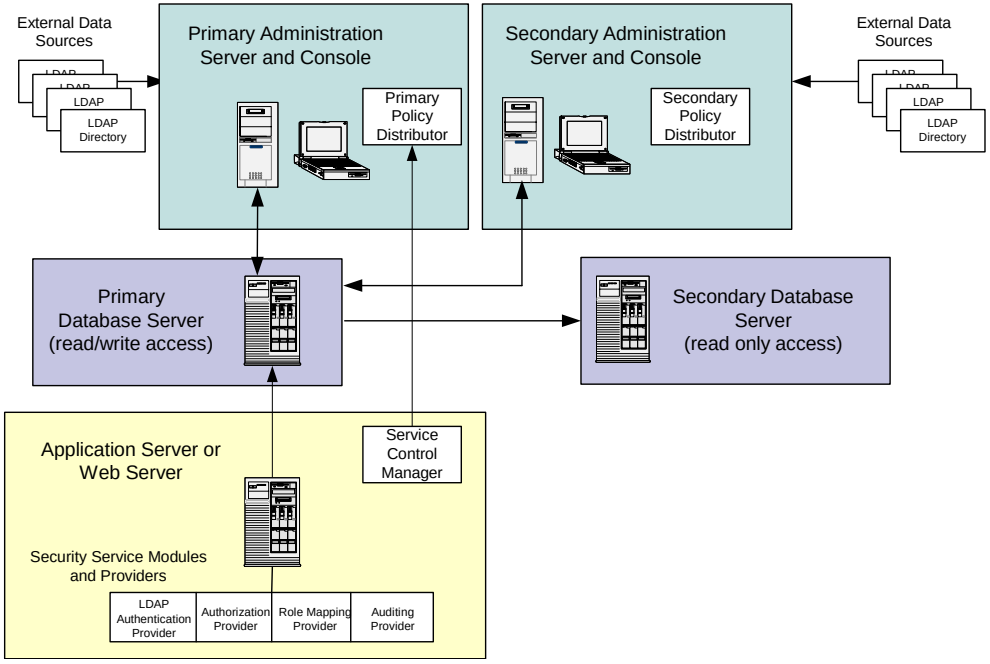
This section describes ALES features that support recovery from failure. It covers the following topics:

Understanding Failover

Figure 4-1 shows how ALES supports failover, through the use of a redundant component architecture. The Administration Console, external data sources (used by the LDAP providers), and the database server, all support failover through configuration in the Administration Console. To ensure reliable operations, you must consider the ramifications of the failure of each component.

To accommodate your basic needs, you can install two Administration Servers: a primary and a secondary. The number of redundant database servers you configure is totally up to you, although a minimum of two is recommended to maintain reliable services. The secondary Administration Server is only used when the primary becomes unavailable; likewise with the database server.

Figure 4-1 ALES System Failover



Failover Considerations for ALES

The following sections discuss the failover considerations for the various ALES components:

- "Failover Considerations for the Administration Server" on page 4-2
- "Failover Considerations for the Database Server" on page 4-3
- "Failover Considerations for a Security Service Module" on page 4-5
- "Failover Considerations for a Service Control Manager" on page 4-6

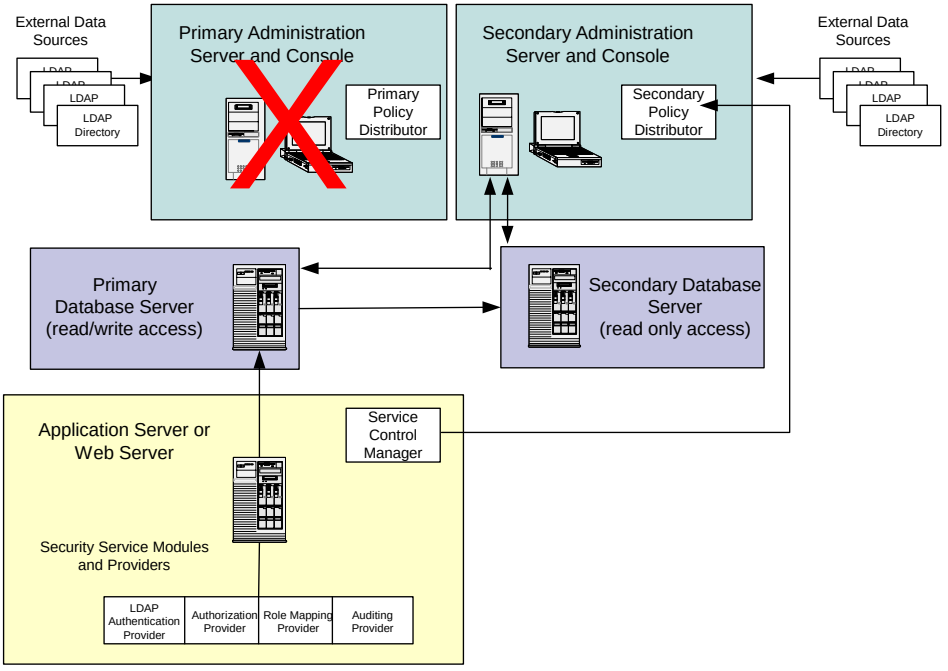
Failover Considerations for the Administration Server

Figure 4-2 shows how failover works when the primary Administration Server fails. 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. As long as you have back up instances of your policy database, external data sources, and back ups of application files, you can safely restart the Administration Server on another machine without interrupting services. However, 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.

For information on how to configure the Administration Server for failover, see "Configuring the Administration Server for Failover" on page 4-29.

Figure 4-2 Administration Server Failover



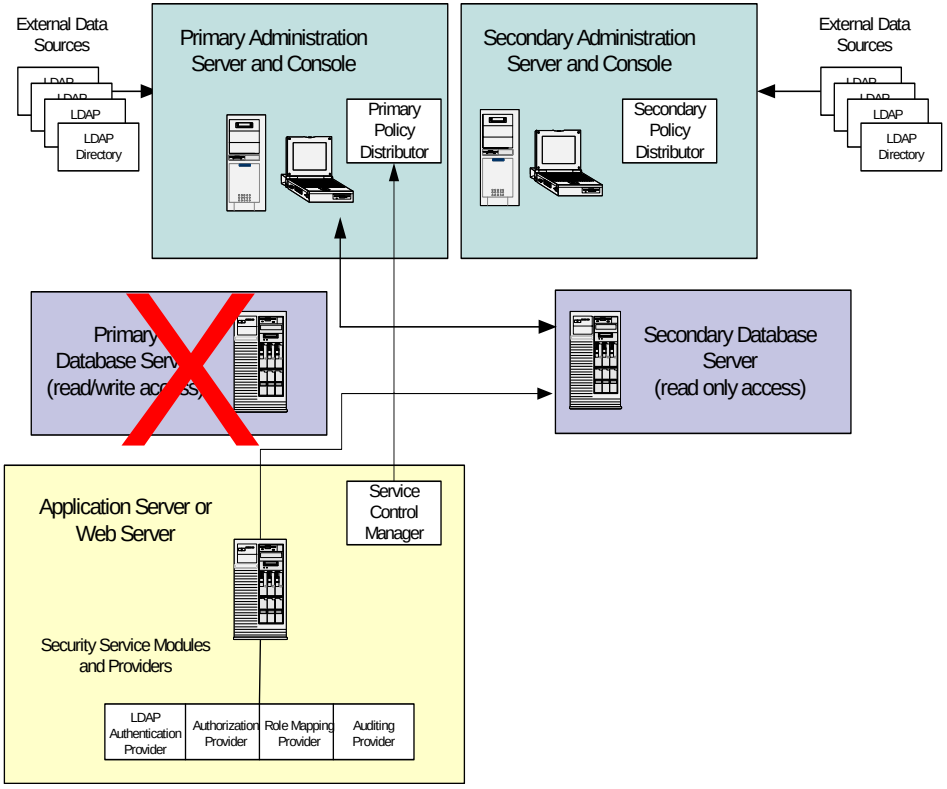
Failover Considerations for the Database Server

Figure 4-3 shows how failover works when the primary Database Server fails. The number of redundant database servers you configure is totally up to you. A minimum of two is

recommended to maintain reliable services: a primary database is read/write, while the secondary provides read-only access. The secondary database, created by using the database replication procedures, contains only the data that is needed for the Security Service Modules, rather than a complete set of the configuration and security data. The secondary database is primarily used for failover in Security Service Modules and not the Administration Server.

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; hot standby).

Figure 4-3 Database Failover



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. The backup can be used for database recovery in the case of disk failure. To provide high availability for the Security Service Modules when using the security providers supplied with the product, you must perform certain database replication procedures as described in the "Oracle Database Replication" on page 4-7 and "Sybase Database Replication" on page 4-16.

Failover Considerations for a Security Service Module

Failover support through the Administration Console is provided for database-related providers and LDAP authentication providers. Configuration for database related providers includes the specification of the secondary (backup 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

Note: 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 may contact an external metadirectory 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 failover replicated metadirectory. The provider tries to reconnect to the failed database after a configurable time-out. If all metadirectory connections fail and the policy evaluation is configured to require user attributes and group membership, all access is denied. For additional information on starting and stopping services, see "Starting and Stopping ALES Components" on page 2-1.

The following providers support configuration of a secondary LDAP server:

- Novell LDAP Authenticator
- Active Directory Authenticator
- iPlanet Authenticator

- Open LDAP Authenticator

Note: 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

The Service Control Manager has a mechanism for enrolling with multiple Administration Servers so that configuration can be administered from multiple Administration Consoles. A process manager is used to monitor the status of the Service Control Manager process and, if it stops responding or fails, the process manager restarts the Service Control Manager. The process manager runs as a Windows NT Service or UNIX daemon. This allows the Service Control Manager to be automatically restarted on machine reboots.

Note: If a Service Control Manager fails, all Security Service Modules controlled by the manager continue to run, but any new Security Service Module instances will not be able to start, because they cannot retrieve their configuration. However, a Security Service Module can still enroll with the Administration Server, even if the Service Control Manager fails.

Understanding Database Replication

To provide reliability and high availability of Security Service Modules, replication of part of the policy database is necessary. The replicated database objects needed for this replication process include tables and indices that are required by the Security Service Modules. They contain the data that are used by the following providers:

- Database Authentication provider
- Database Credential Mapper provider
- ASI Authorization and ASI Role Mapper providers

Note: The ASI Authorization and ASI Role Mapper providers are implemented using an out-of-process evaluation engine (the ARME process). If this process fails, access for all resources is denied. This process is monitored and restarted if an error occurs.

Most of the policy database used for security policy and for configuration administration and management is not replicated. The administration and management functions are performed against one and only one database. This means that the components such as the Administration

Application, the Policy Distributor, and the Metadirectory Synchronization process are only connected to one database.

Warning: Your administrator needs to ensure that the primary database is backed up regularly using a database backup tool or using a hardware backup strategy to protect the database against fatal systematic errors, should one ever occur.

The steps included in following sections describe how to create a read-only database replica. While some scripts automate this process, to complete the replication, you need to perform certain steps manually. For instructions on how to replicate databases in the Oracle and Sybase environments, see the following sections:

- "Oracle Database Replication" on page 4-7
- "Sybase Database Replication" on page 4-16

Oracle Database Replication

ALES uses the Materialized View (MV) Replication method to replicate the database objects from a master database (also called the master site) to one or more materialized view databases (also called the materialized view sites). In an earlier version of Oracle, such as Oracle 8i, a materialized view replication was called a snapshot replication, and a materialized view site was called a snapshot site. The master database is the database that is administered through the Administration Application. The database schema of the master database is installed and set up during the Administration Application installation. The materialized view database provides failover and availability for the Security Service Modules. You can set up the Security Service Modules at a later time, according to your business needs.

The following sections describe the replication tasks:

- "Preparing for Oracle Database Replication" on page 4-7
- "Setting Up Oracle Database Replication" on page 4-12

Preparing for Oracle Database Replication

Use the following items as a checklist to prepare your replication environment.

- "Master and Materialized View Site Requirements" on page 4-8
- "Master Site Requirements" on page 4-10
- "Materialized View Site Requirements" on page 4-10

- "Requirements for the Machine Running the Replication Setup Scripts" on page 4-11
- "Setting the Required Replication Setup Parameters" on page 4-11

Warning: If you are concerned with the database password being used in the scripts, you can run the replication setup manually. All passwords are designated as clear text in the `set_repl_env_oracle.sh` and `set_repl_env_oracle.bat` scripts. When you are through, you can either delete the script or remove the password from the script.

Master and Materialized View Site Requirements

Requirements for the master site and the materialized view site include the following:

- You are able to login to the database as database administrator (DBA) and you have the privileges to restart the database.
- Ensure that the database name is in the form of `DB_NAME.DB_DOMAIN`. This ensures that the database links can be set up correctly. You can check this by running this SQL statement after logging into the database:

  ```
  select * from global_name;
  ```

  If the `DB_DOMAIN` is missing, correct it. This task requires DBA privileges.
- Change the initialization parameters for both databases (the master database and the materialized view database) according to Table 4-1. The value for each initialization parameter depends on your use of the database. Refer to Table 4-1, and set the parameters as follows:
  - If the default values for the parameters that are defined as "Required" in the Comment field do not suit your needs, change them.
  - Set the parameters that are defined as "Optional" in the Comment field according to the use of your database.

For detailed information on these parameters, refer to your Oracle Documentation. For Oracle 9i, see Chapter 6: "Planning Your Replication Environment" in *Advanced Replication* (for Oracle 9.2.0.5). For Oracle 10.0.2, see TBD). Also check the Oracle documentation for details on how to set the initialization parameters.

Table 4-1 Initialization Parameters for Oracle 9.2.0.5

Parameter	Recommended Value	Comment
GLOBAL_NAMES	true	Required
JOB_QUEUE_PROCESSES	at least 1	Required
OPEN_LINKS	at least 4	Required
PROCESSES	at least 40	Required
REPLICATION_DEPENDENCY_TRACKING	true	Required
COMPATIBLE		Optional
PARALLEL_MAX_SERVERS		Optional
PARALLEL_MIN_SERVERS		Optional
SHARED_POOL_SIZE		Optional
PARALLEL_AUTOMATIC_TUNING		Optional
UTL_FILE_DIR	A valid directory in the host machine of the database	Required: the directory must be created

Table 4-2 Initialization Parameters for Oracle 10.1.2

Parameter	Recommended Value	Comment
GLOBAL_NAMES	true	Required
JOB_QUEUE_PROCESSES	at least 1	Required
OPEN_LINKS	at least 4	Required
PROCESSES	at least 40	Required
REPLICATION_DEPENDENCY_TRACKING	true	Required
COMPATIBLE		Optional
PARALLEL_MAX_SERVERS		Optional

Table 4-2 Initialization Parameters for Oracle 10.1.2 (Continued)

Parameter	Recommended Value	Comment
PARALLEL_MIN_SERVERS		Optional
SHARED_POOL_SIZE		Optional
PARALLEL_AUTOMATIC_TUNING		Optional
UTL_FILE_DIR	A valid directory in the host machine of the database	Required: the directory must be created

Master Site Requirements

Requirements for the master site include the following:

- Your master database must be either an Oracle Enterprise edition or an Oracle Standard edition.
- Ensure that the replication administrator REPADMIN and PROXY_ADMIN does not already exist in this database. If either one already exists, you cannot automate the replication setup using the scripts because the scripts will fail to create them. You need to set up the master site manually. You can use SQL `select * from all_users` to check for its existence.
- Ensure that the policy database schema is already installed. You need to know the schema owner (WLES user) and login password.
- Ensure that the size of the TABLESPACE used by the WLES user is large enough to accommodate the extra storage needed by replication.

Materialized View Site Requirements

Requirements for the materialized view site include the following:

- In this host machine, set up an Oracle Local Service Name for the master database using the same name as the master database name, in the form of DB_NAME.DB_DOMAIN.
- Ensure that the replication administrator MVADMIN does not already exist in this database. If it already exists, you cannot automate the replication setup using the scripts because the scripts will fail to create it. You need to set up the materialized view site manually.

- Ensure that the same WLES user as in master database does not exist in this database as this user is created. However, you need to locate a TABLESPACE for this WLES user before the setup using the scripts. By default, the temporary TABLESPACE for this user is set to TEMP.
- Decide how frequently you want the replicated objects refreshed in this database. Set the refresh time interval to a multiple of one minute.

Requirements for the Machine Running the Replication Setup Scripts

On this machine, set up Oracle Local Service Names for the master database and materialized view database, using the same names as their database names, in the form of:

```
DB_NAME.DB_DOMAIN
```

Setting the Required Replication Setup Parameters

Table 4-3 lists and describes the parameters that you must set before you can set up the replication. You must set these parameters in the setup script:

```
set_repl_env_oracle.bat (Windows)
```

or

```
set_repl_env_oracle.sh (Unix/Linux)
```

In addition to the parameters listed in Table 4-3, you need to set ORACLE_HOME, if it is not yet set in your environment, and the __REPL_SETUP_TYPE__ parameter, depending on what kind of replication setup you choose.

Table 4-3 Oracle Replication Setup Parameters

Parameter Name	Description	Example
__MASTER_SERVICE_NAME__	Service name of master database (DB_NAME.DB_DOMAIN).	mstrdb.bea.com
__MASTER_SYSTEM_PASSWD__	Password for SYSTEM user (DBA) in master database.	Pswd000
__REPADMIN_PASSWD__	Password for REPADMIN (to be created/dropped, replication admin) in master database.	Pswd111
__PROXY_MVADMIN_PASSWD__	Password for PROXY_MVADMIN (to be created/dropped) in master database.	Pswd222

Table 4-3 Oracle Replication Setup Parameters (Continued)

Parameter Name	Description	Example
__MASTER_PURGE_INTEVAL__	Time interval in minutes to purge completed deferred transactions at master site.	60
__MV_SERVICE_NAME__	Service name of materialized view (MV) database (DB_NAME.DB_DOMAIN).	mvdb.bea.com
__MV_SYSTEM_PASSWD__	Password for SYSTEM user (DBA) in materialized view database.	Pswd333
__MVADMIN_PASSWD__	Password for MVADMIN (to be created/dropped, replication admin) in MV database.	Pswd444
__TBLSPACE_DBUSER_WLES__	Tablespace for WLES database user (__DBUSER_WLES_UPPERCASE__) in MV database.	USERS
__MV_PURGE_INTEVAL__	Time interval in minutes to purge completed deferred transactions at MV site.	60
__MV_PUSH_INTEVAL__	Time interval in minutes to push deferred transactions to master at MV site.	60
__REFRESH_INTEVAL__	Time interval in minutes to refresh the refresh group at MV site.	2
__DBUSER_WLES_UPPERCASE__	The policy database user name in both the master and materialized view databases; must be in UPPER CASE. It cannot exist in the materialized view database before setting up replication. This user is created to use tablespace: __TBLSPACE_DBUSER_WLES__	WLES_USER
__DBUSER_PASSWD_WLES__	The database user password in both master and materialized view databases.	Pswd555

Setting Up Oracle Database Replication

Use the following procedures to set up the Oracle database replication using scripts or manually.

Warning: You must set up the master site before replication setup can be performed for a materialized view site.

Using Scripts to Set Up Oracle Database Replication

To use scripts to set up replication, perform the following steps:

1. Change to the directory:

   `BEA_HOME/bin`

2. To set up the environment and input parameters, edit the following file:

   `set_repl_env_oracle.bat` (Windows)

   or

   `set_repl_env_oracle.sh` (Unix/Linux)

   Save a copy of this file before editing. The parameter `__REPL_SETUP_TYPE__` gives you the option to set up either a master site, a materialized site or both.

3. Run the following script:

   `replicate_oracle.bat` (Windows)

   or

   `replicate_oracle.sh` (Unix/Linux)

   You are prompted to continue. A message is displayed when the script completes, indicating whether the execution was successful or if it failed. The following log file shows the status of the execution:

   `BEA_HOME/log/replicate_oracle.log`

4. After the setup completes, edit or delete the following file and close the window:

   `set_repl_env.oracle.bat` (Windows)

or

`set_repl_env_oracle.sh` (Unix/Linux)

Setting Up Oracle Database Replication Manually

To set up replication manually, perform the following steps:

1. In the `BEA_HOME/data/oracle/` directory, locate these SQL files:

   `rep_mastersite.sql`

   `rep_mvsite.sql`

2. Save a copy of each one.
3. Edit them by globally replacing the variables with your settings, as instructed at the beginning of each file.
4. Manually execute the step-by-step SQL statements using the SQLPLUS tool. The comments in these SQL files describe what each step does.

Using Scripts to Clean Up Oracle Database Replication

Warning: Replication clean up must be performed for all materialized view sites before the master site can be cleaned up.

To use scripts to clean up the replication, perform the following steps:

1. Change to the directory:

   `BEA_HOME/bin`

2. Set the environment and input parameters in the following file:

   `set_repl_env_oracle.bat` (Windows)

   or

   `set_repl_env_oracle.sh` (Unix/Linux)

   Save a copy of this file before editing it. The parameter `__REPL_SETUP_TYPE__` gives you the option to clean up either a master site, a materialized site or both.

3. Run the following script:

   `clean_repl_oracle.bat` (Windows)

   or

`clean_repl_oracle.sh` (Unix/Linux)

You are prompted to continue. A message is displayed when the script completes, indicating whether the script was successful or if it failed. The following log file shows the status of the execution:

`BEA_HOME/log/clean_repl_oracle.log`

4. After the cleanup completes, edit or delete following file and close the window:

   `set_repl_env_oracle.bat` (Windows)

   or

   `set_repl_env_oracle.sh` (Unix/Linux)

Cleaning Up the Oracle Database Replication Manually

To clean up the replication manually, perform the following steps:

1. In the `BEA_HOME/data/oracle/` directory, locate the following SQL files:

   `rep_clean_master.sql`

   `rep_clean_mv.sql`

2. Save a copy of each one.
3. Edit the files by globally replacing the variables with your settings, as instructed at the beginning of each file.
4. Manually execute the SQL statements step-by-step using the SQLPLUS tool. The comments in these SQL files describe what each step does.

Miscellaneous Oracle Database Replication Tasks

To complete the replication of the Oracle database, perform the following steps:

1. The database name must be in the form of `DB_NAME.DB_DOMAIN`. If the database name is not in this form, login to the database as DBA and run the following SQL command to change it:

   ```
   alter database rename global_name to DB_NAME.DB_DOMAIN;
   ```

2. Change the initialization parameters (refer to Oracle documentation on how change the initialization parameters):

   `db_name: to DB_NAME`

   `db_domain: to DB_DOMAIN`

```
service_name: to DB_NAME.DB_DOMAIN
```

3. Change the listener for the new `global_name`.
4. If you change the `global_name`, update the Oracle listener setting, and then restart the listener. The `GLOBAL_NAME` is recorded in the Oracle configuration file called `listener.ora` in the `ORACLE_HOME/network/admin/` directory.
5. Change the setting for the Local Service Name in all Oracle clients.

   If you change the local service name that points to this database, you must update it on all applicable client machines. Change the `SERVICE_NAME` of the `CONNECT_DATA` to the new `service_name` in client configuration file `tnsnames.ora`, usually in the `ORACLE_HOME/network/admin/` directory. If your system uses Directory Naming or Oracle Naming methods, update them instead of `tnsnames.ora`. Refer to your Oracle documentation for details.

Sybase Database Replication

ALES uses the Sybase Adaptive Server Enterprise (ASE) Replicator to replicate the database objects from a primary database in primary ASE server to one or more replicate databases in the replicate ASE servers. The ASE Replicator process is an external application that connects to and interacts with the ASE server, and coordinates all replication processing. The policy in the primary database is administered though the Administration Server. The database schema in the primary database is installed and set up during the Administration Application installation. The replicate databases provide failover and availability for the ALES Security Service Modules. They can be set up according to your business needs at a later time. For more information, see the *ASE Replicator User's Guide* from Sybase.

The following sections describe the replication tasks:

- "Preparing for Sybase Database Replication" on page 4-16
- "Setting Up Sybase Database Replication" on page 4-21
- "Cleaning Up Sybase Database Replication" on page 4-25

Preparing for Sybase Database Replication

The following sections provide checklists so you can prepare your replication environment:

- "Privileges for the Primary and the Replicate ASE Servers" on page 4-17
- "Primary ASE Server and Primary Database Requirements" on page 4-17

- “Replicate ASE Server and Replicate Database Requirements” on page 4-18
- “Requirements for the Machine Used to Run Sybase Database Replication Setup Scripts” on page 4-19
- “Parameters Needed for Sybase Database Replication Setup” on page 4-19

Warning: If you are concerned with the database password being used in the scripts, BEA recommends that you run the replication setup manually. All passwords are designated in clear text in the `set_repl_env_sybase.sh` and `set_repl_env_sybase.bat` scripts. When you are through, you can either delete the script or remove the password from the script.

Privileges for the Primary and the Replicate ASE Servers

Make sure you are able to login to the servers as DBA (sa) and you have the privilege to restart the servers.

Primary ASE Server and Primary Database Requirements

The primary ASE server and primary database requirements are as follows:

- Your server must be of ASE version 12.5.0.3 or later. This version includes the ASE Replicator software. You need to patch it if the server is of earlier version of 12.5.
- The ASE Replicator is started and is running in this machine, even though if can be run from other machines that has ASE Replicator software installed.
- Ensure the ASE Replicator system user login does not already exist in this server. You can specify the name of this login in the setup scripts. If the system user login already exists, you cannot automate the replication setup using the scripts because the scripts will fail to create it, and you will need to set up the replication manually. You can use SQL select name from syslogin to check its existence.
- Ensure that the policy database schema is already installed. You need to know the schema owner (`WLES` user) and login password.
- Ensure that you (sa) have created the database for the replication process to use as the distribution database. Before creating the database, you may need to create two database devices, one used for data and another used for transaction log. Allocate sufficient database storage. You can use the `disk init …` and `create database …` isql commands, or use another GUI tool to set these up. Also, while this database is dropped when running the replication cleanup scripts, the database devices are not. See the *Administration Application Installation Guide* for instructions on how to set up the database.

- In this host machine, set up the following server entries in Sybase configuration file using either the Sybase tool Dsedit or using a text editor. The configuration file is:

  `sql.ini` (Windows)

  or

  `interfaces` (Unix/Linux).

- Primary ASE server—the same name as the primary database server.
- ASE Replicator—the name and port number you choose here is used later when you start the ASE Replicator.
- Replicate ASE server—the same name as the replicate server.

Replicate ASE Server and Replicate Database Requirements

The replicate ASE server and replicate database requirements are as follows:

- Ensure that this ASE server is installed and configured with logical page size equal to or larger than the logical page size of the primary ASE server. Use SQL statement: `select @@maxpagesize` to check them out.
- Ensure the replication Maintenance user login does not already exist in this server. The same ASE Replicator system user login is used. If it already exists, you cannot automate the replication setup using the scripts because the scripts will fail to create it, and you will need to set up the replicate database manually.
- Ensure the same WLES user login as in primary ASE server does not exist in this server, as it is created. If it already exists, you cannot automate the replication setup using the scripts because the scripts will fail to create it, and you will need to set up the replication manually.
- Ensure that you (sa) have created a new database to use as the replicate database. This database is used to store replicated policy data. Before creating the database, you may need to create two database devices, one used for data and another used for transaction log. Allocate sufficient database storage. Transaction log size can be much smaller than that in policy database in the primary ASE server. Also, while this database and the database devices are not be dropped when running the replication cleanup scripts, the schema for WLES user is dropped.

Requirements for the Machine Used to Run Sybase Database Replication Setup Scripts

The requirements for the machine used to run the Sybase database replication setup Scripts are as follows:

Note: If you use the primary ASE server machine to run the Sybase database replication setup scripts, you do not have to do anything and you can skip this section.

- In this host machine, set up the following server entries in the Sybase configuration file using Sybase tool Dsedit or using text editor. The configuration file is:

  `sql.ini` (Windows)

  or

  `interfaces` (Unix/Linux)

  You can copy this file from the primary ASE server machine.

- Primary ASE server—the same name as the primary database server.
- ASE Replicator—the server name of the ASE Replicator.
- Replicate ASE server—the same name as the replicate server.

Parameters Needed for Sybase Database Replication Setup

Table 4-4 lists the initialization parameters you need to set before you set up the replication. You set these parameters in:

`set_repl_env_sybase.bat` (Windows)

or

`set_repl_env_sybase.sh` (Unix/Linux)

You also need to set SYBASE if it is not yet set in your environment. In addition, you need to set `__REPL_SETUP_TYPE__` depending on what kind of replication setup you choose.

<~runChNum>

Table 4-4 Sybase Initialization Parameters

Parameter Name	Description	Example
__PRIMARY_DBSERVER__	Name of the primary ASE server.	PRISVR
__PRIMARY_DBNAME__	Name of the primary policy database in the primary ASE server.	policy
__SA_PASSWD_PRIMARY__	Password for sa (DBA) in the primary ASE server.	Pswd000
__DISTRIBUTION_DBNAME__	Distribution database name in the primary ASE server. This database is used by the replication process (or Replicator). The database contains the database schema (tables, etc.) used by the replicator.	distrDB
__ASE_REPLICATOR_NAME__	Name of the ASE Replicator. This name should be the same as that in the Sybase configuration file.	PRISVR_RPL
__REPLICATE_DBSERVER__	Name of the replicate ASE server.	RPLSVR
__REPLICATE_DBNAME__	Name of the secondary policy database in the replicate ASE server.	policy
__SA_PASSWD_REPLICATE__	Password for sa (DBA) in the replicate ASE server.	Pswd111
__USER_REPADMIN__	ASE Replicator system username/login and Maintenance username/login.	repadmin
__PASSWD_REPADMIN__	Password for user __USER_REPADMIN__.	Pswd222
__DBUSER_WLES__	Policy database username/login in both primary and replicate ASE servers.	WLESuser
__DBUSER_PASSWD_WLES__	Policy database password in both primary and replicate ASE servers.	Pwd333
__REPL_SETUP_TYPE__	Indicates whether to set or clean up both primary and replicate databases. The value is one of these: both, primary, or replicate.	both

You also need to know the values for __SIZE_IDXCOL__ and __SIZE_COL__ when you set up a replication manually. The value for these two can be found in the policy database in the primary ASE server using SQL statement:

```
select __SIZE_IDXCOL__ = idx_column_size, __SIZE_COL__ =
reg_column_size from __DBUSER_WLES__ column_sizes
```

Note: `__DBUSER_WLES__` is substituted with the value described.

Setting Up Sybase Database Replication

Use the following procedures to set up the Sybase database replication using scripts or manually.

- "Using Scripts to Set Up Sybase Database Replication" on page 4-21
- "Setting Up Sybase Database Replication Manually" on page 4-22
- Setting up the Primary ASE Server and the Primary Database
- "Starting the ASE Replicator" on page 4-23
- "Adding a Remote Server in Primary ASE Server for the Replicate ASE Server" on page 4-24
- "Setting Up the Replicate ASE Server and the Primary Database" on page 4-24
- "Setting up the Sybase Database Replication Process" on page 4-25

Warning: The Primary ASE server and database must be set up and ASE Replicator started before you can perform replication setup on a replicate database.

Using Scripts to Set Up Sybase Database Replication

To use scripts to set up Sybase database replication, perform these steps:

1. Change to the following directory:

   `BEA_HOME/bin`

2. Edit the following file to set the environment and input parameters:

   `set_repl_env_sybase.bat` (Windows)

   or

   `set_repl_env_sybase.sh` (Unix/Linux)

   Save a copy of this file before editing it. The parameter `__REPL_SETUP_TYPE__` allows you to set up either a primary, replicate database, or both.

3. Run the following script:

   `replicate_sybase.bat` (Windows)

or

`replicate_sybase.sh` (Unix/Linux)

4. You are prompted to continue.
5. You are then prompted to restart the primary ASE server. Go to the primary ASE server machine, and stop and start the ASE server. After the server is started, press <Enter> to continue.
6. You are prompted to start the ASE Replicator. Go to the primary ASE server machine and start the ASE Replicator process. After the server is started, press <Enter> to continue.

   You are prompted to continue. A message is displayed when the script completes, indicating whether the script was successful or if it failed. The following log file shows the status of the execution:

   `BEA_HOME/log/replicate_sybase.log`

7. Edit or delete the file:

   `set_repl_env_sybase.bat` (Windows)

   or

   `set_repl_env_sybase.sh` (Unix/Linux)

8. Close the window after the setup completes.

Setting Up Sybase Database Replication Manually

To set up Sybase database replication manually, perform these steps:

1. In the `BEA_HOME/data/sybase/` directory, locate and copy the following SQL files:

   `rep_pdb_conf.sql`

   `rep_rsvr_rdb.sql`

   `rep_rdb_conf.sql`

   `rep_replication.sql`

2. Edit them by globally replacing the variables with your settings, as instructed in each file.
3. Manually execute the SQL statements step-by-step using the isql tool. The order is described in the following sections. When you are setting up a second replicate database, you need to perform the following tasks:
   - Setting up the Primary ASE Server and the Primary Database

Setting up the Primary ASE Server and the Primary Database

Login to the primary ASE server as sa (DBA) and execute: `rep_pdb_conf.sql`. You can do this statement-by-statement. You also need to restart the primary ASE server manually, after running this script. This SQL file performs the following tasks.

1. Enables and configures CIS in the primary ASE server.
2. Sets up ASE Replicator system user login and assigning it to replication role.
3. Adds the ASE Replicator system user to the policy database (primary database) and grants its permissions.
4. Defines remote servers in the primary ASE server.
5. Defines the local server name for the primary ASE server.
6. Defines a server named `local` as a remote alias for the primary ASE server.
7. Defines a remote server for the ASE Replicator.
8. Configures the `tempdb` database.
9. Sets up ASE Replicator system user and changes the database option in the distribution database.
10. Adds the ASE Replicator system user to the distribution database.
11. Grants permissions to ASE Replicator system user in the distribution database.
12. Changes the database option in the distribution database.
13. Sets up the ASE Replicator system user in database `sybsystemprocs`.
14. Creates the `sp_helpddb` stored procedure in database `sybsystemprocs`, and grants execution permission to ASE Replicator system user.

Starting the ASE Replicator

To start the ASE replication, perform these steps:

1. Go to the primary ASE server machine and change to the directory: `RPL-12_5/bin` in the Sybase installation directory.
2. If ASE Replicator is started the first time:

a. Make sure the `RPL-12_5` directory has write permission.

b. Using the proper command options, execute the following command:

`aserep.bat` (Windows)

or

`aserep.sh` (Unix/Linux)

Check the *ASE Replicator User's Guide* for details.

Note: You can restart the ASE Replicator by running either: `aserep.bat` or `aserep.sh` using the same command options, or by running the `RUN` script that is located under `SYBASE/RPL-12_5/replicatorName` and providing the -u and -p options.

Adding a Remote Server in Primary ASE Server for the Replicate ASE Server

To add a remote server in the primary ASE server for the replicate ASE server, perform these steps:

1. Login to primary ASE server as sa (DBA).
2. Execute the following script:

   `rep_rsvr_rdb.sql`

Setting Up the Replicate ASE Server and the Primary Database

Login to the replicate ASE server as sa (DBA) and execute: `rep_rdb_conf.sql`. You can do this statement-by-statement. Before starting, make sure the replicate policy database already exists. This SQL file performs the following tasks.

1. Adds the `WLES` user login.
2. Adds the `WLES` user login to the replicate database.
3. Creates the replicate tables and indices in replicate database, according to the logical page size setting of primary database.
4. Adds a Maintenance user login (for example, `repadmin`). This is the same as ASE Replicator system user login.
5. Adds `repadmin` as a user in replicate database.
6. Grants permission to `repadmin`.

Setting up the Sybase Database Replication Process

Login to primary ASE server as ASE Replicator system user login and execute: `rep_replication.sql`. You can do this statement-by-statement. Before starting, make sure the ASE Replicator is running. This SQL file performs the following tasks:

1. Creates the primary database connection if it does not yet exist.
2. Creates a replicate database connection.
3. Suspends both the replicate and primary database connections.
4. Creates a publication.
5. Adds the articles (replicated objects) to publication.
6. Creates a subscription.
7. Adds the articles (replicated objects) to subscription.
8. Materializes the replication articles (replicated objects).
9. Resumes both replicate and primary database connections to start replication.

Cleaning Up Sybase Database Replication

Use the following procedures to automate replication cleanup or to do it manually. Replication cleanup also stops the ASE Replicator and drops the distribution database in the primary ASE server. If you experience a failure while using the scripts to clean up replication, you are advised to finish the cleanup manually.

Before attempting cleanup, make sure that there is no database connection for `WLES` user login to the replicate ASE server and for ASE Replicator system user login to both replicate and primary ASE servers. If there are connections, cleanup will fail. You can use `sp_help loginName` to check whether the user still has connections.

- "Using Scripts to Clean Up Sybase Database Replication" on page 4-26
- "Cleaning Up the Sybase Database Replication Manually" on page 4-27
- "Cleaning Up the Sybase Database Replication Process" on page 4-27
- "Cleaning Up the Replicate ASE Server and Primary Database" on page 4-28
- "Removing the Remote Server" on page 4-28

- "Stopping ASE Replicator" on page 4-28
- "Cleaning Up the Primary ASE Server and the Primary Sybase Database" on page 4-29
- "Completing Sybase Database Replication Cleanup" on page 4-29

Warning: You must perform the replication clean up for all replicate databases before the primary database can be cleaned up and the ASE Replicator removed.

Using Scripts to Clean Up Sybase Database Replication

To use scripts to clean up Sybase database replication, perform these steps:

1. Make sure that there is no database connection for `WLES` user login to the replicate ASE server, and for ASE Replicator system user login to both the replicate and primary ASE servers.
2. Change to the following directory:

   `BEA_HOME/bin`

   Set the environment and input parameters in the following file:

   `set_repl_env_sybase.bat` (Windows)

   or

   `set_repl_env_sybase.sh` (Unix/Linux)

   Save a copy of this file before editing it. The parameter `__REPL_SETUP_TYPE__` gives you the option to clean up either a primary database and ASE server, a replicate database and ASE server, or both.
3. Run the following script:

   `clean_repl_sybase.bat` (Windows)

   or

   `clean_repl_sybase.sh` (Unix/Linux)

   You are prompted to continue. A message is displayed when the script completes, indicating whether the execution was successful or if it failed. The following log file shows the status of the execution:

   `BEA_HOME/log/clean_repl_sybase.log`
4. Edit or delete the following file:

   set_repl_env_sybase.bat (Windows)

or

set_repl_env_sybase.sh (Unix/Linux)

5. Close the window after the cleanup completes.
6. To complete replication cleanup, remove the ASE Replicator process by deleting all files from the `RPL-12_5/replicatorName` Sybase installation directory.

Cleaning Up the Sybase Database Replication Manually

To clean up Sybase database replication manually, perform these steps:

1. In the `BEA_HOME/data/sybase/` directory, locate the following SQL files and save a copy of each one:

   `rep_clean_replication.sql`

   `rep_clean_rdb.sql`

   `rep_clean_rsvr_rdb.sql`

   `rep_clean_stop_rpl.sql`

   `rep_clean_pdb.sql`

2. Edit each file globally, replacing the variables with your settings, as instructed in each file.
3. Manually execute the SQL statements step-by-step using isql tool. The order is described below. When you are cleaning up a replicate database alone, you need to perform the following tasks:
   - "Cleaning Up the Replicate ASE Server and Primary Database" on page 4-28
   - "Removing the Remote Server" on page 4-28

Cleaning Up the Sybase Database Replication Process

Login to the primary ASE server as the ASE Replicator system user login and execute: `rep_clean_replication.sql`. You can do this statement-by-statement. This SQL file performs the following tasks:

1. Suspends both the replicate and primary database connections.
2. Drops the articles (replicated object) from subscription.
3. Drops the subscription.
4. Drops the articles (replicated object) from publication.

5. Drops the publication.
6. Drops the replicate database connection.
7. Drops the primary database connection if it is not being used for any other publications; resumes the connection if it is used.

Cleaning Up the Replicate ASE Server and Primary Database

Login to replicate ASE server as sa (DBA) and execute: `rep_clean_rdb.sql`. You can do this statement-by-statement. This SQL file performs the following tasks:

1. Drops the replicate tables. Indices are dropped automatically.
2. Drops the `WLES` user in replicate database.
3. Deletes the `WLES` user login in replicate ASE server.
4. Drops the Maintenance user in replicate database.
5. Deletes the Maintenance user login in the replicate ASE server.

Removing the Remote Server

In the Primary ASE Server for the Replicate ASE Server, follow these steps to remove the remote server:

1. Login to the primary ASE server as sa (DBA).
2. Execute the following script:

   ```
   rep_clean_rsvr_rdb.sql
   ```

3. In the primary ASE server, remove the remote server for the replicate ASE server.

Stopping ASE Replicator

To stop the ASE Replicator, perform these steps:

1. Login to the ASE Replicator, using the ASE Replicator system user login.
2. Execute the following script to shut down the ASI Replicator:

   ```
   rep_clean_stop_rpl.sql
   ```

Note: Alternatively, shutdown ASE Replicator manually.

Cleaning Up the Primary ASE Server and the Primary Sybase Database

Login to primary ASE server as sa (DBA) and execute: `rep_clean_pdb.sql`. You can do this statement-by-statement. This SQL file performs the following tasks.

1. Drops the procedure `sp_helpddb` in database `sybsystemprocs`.
2. Drops the ASE Replicator system user in database `sybsystemprocs`.
3. Drops the ASE Replicator system user in primary policy database.
4. Removes remote servers in the primary ASE server.
5. Removes the local server name for the primary ASE server.
6. Removes the server named `local` as a remote alias for the primary ASE server.
7. Removes the remote server for the ASE Replicator.
8. Drops the distribution database.
9. Deletes the ASE Replicator system user login.

Completing Sybase Database Replication Cleanup

To complete replication cleanup, remove the ASE Replicator process by deleting all of the files under `RPL-12_5/replicatorName` in your Sybase installation directory.

Configuring the Administration Server for Failover

You can install two administration servers: a primary and a secondary. The secondary administration server is only used as a backup when the primary becomes unavailable. You must install the secondary server before you can configure it for backup purposes. For information on installing a secondary server, see the Administration Application Installation Guide.

To configure the Administration Server for failover:

1. In the Administration console, click on `Administration Console` at the top of the navigation tree to view the Console Preferences page.
2. Click the Failover tab.This tab allows you to configure this Administration Server as either a primary or a secondary (backup) Administration Server. If this is a secondary server, you must specify all parameters so the primary server can be located and can periodically request a list of trusted entities. This mechanism keeps the primary and secondary

synchronized so that the secondary server can be designated as the primary Administration Server if necessary. 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 for enrollment on the Secondary server. This URL is used to synchronize a trust relationship.
5. In the Username text box, enter the username to use when requesting synchronization of a trust relationship.
6. In the Enter Password and Confirm Password text boxes, enter the password to use when requesting synchronization of a trust relationship.
7. In the Synchronization interval text box, enter the number of seconds between trust relationship synchronization attempts in the Synchronization Interval text box.

   The value for this setting depends on how frequently Security Service Module or Service Control Manager instances are enrolled and unenrolled from the primary Administration Application.
8. Click Apply.