Oracle ® Entitlements Server 10*g* (10.1.4.3)

Knowledgebase

September 2008

ORACLE

FAQs, Oracle ® Entitlements Server 10*g* (10.1.4.3)

Silent Mode Installations

Failover and System Reliability

Collecting Performance Statistics

Host Name or IP Address Change

Database Password Changes

Resource Discovery

Configuring an XACML Client for a Custom Identity Asserter

Running the Java SSM in Java Development Environments

Debugging SSL Connectivity

Debugging Policies

Running Multiple Administration Servers on One Machine

Pointing the Administration Server to a Different Database

CHAPTER 1

Silent Mode Installations

When the Administration Server is installed, its configuration data is stored in an XML file. This file may be used to perform 'silent installs' of the server on other machines.

Entries in the configuration file correspond to the responses entered during a normal install. They can be modified as needed on the new machine.

Note: See **SSM Installation and Configuration Guide** for instructions on how to silently install Security Modules.

Modify the Configuration File

To modify the configuration file for a silent install of the Administration Server:

1. Make a copy of the configuration file and open it in an editor. The file path is *BEA_HOME*/ales32-admin/config/silent_install_admin.xml.
2. Use Table 1-1 to modify the installation parameters. These are specified in XML syntax as name/value pairs. The values that can be modified are in the **value=** field. For example, in the entry below, you could change the directory name:

   ```
   <data-value name="ALES_ADMIN_DIR" value="C:\bea\ales32-admin" />
   ```

Table 1-1 Silent Installation Configuration File

Data-Value Name	Description	Examples
BEAHOME	BEA_HOME directory	C:\bea
ALES_ADMIN_DIR	Administration Server install directory	`C:\bea\ales32-admin`
SCM_INSTALL_DIR	SCM install directory.	`C:\bea\ales32-scm`
WEB_SERVER_TYPE	Servlet container type being used.	weblogic81 weblogic92 weblogic10 tomcat
WEB_SERVER_DIR	Servlet container directory. **Note:** When using Tomcat, the directory name cannot contain spaces.	C:\bea\weblogic81
ADMIN_APP_PORT	Port for the servlet container's administration console.	7000
ADMIN_APP_SSL_PORT	SSL port for the Administration Server	7010
ADMIN_JAVA_HOME	Administration Server JDK	C:\bea\jrockit90_150_04
SCM_JAVA_HOME	SCM JDK	C:\bea\jrockit90_150_04
ENTERPRISE_DOMAIN_NAME	Must be **asi**.	
CERTIFICATE_DURATION	Years the security certificate remains in effect.	10
DATABASE_CLIENT	Database type	ORACLE92 ORACLE10 SYBASE125 SYBASE15 PointBase 5.1 MS SQL Server 2000 MS SQL Server 2005 DB2

Table 1-1 Silent Installation Configuration File (Continued)

Data-Value Name	Description	Examples
DB_DRIVER_LOC	(MS SQL, Pointbase, DB2 only) Directory containing the database driver. **Note:** DB2 Driver license jar is shipped in separate jar. Append both jars and separate using OS-specific classpath separator.	
JDBC_URL	URL on which to reach the database	jdbc:sybase:Tds:ALESDB:5000
JDBC_DRIVER	Java classname of the database driver.	com.sybase.jdbc3.jdbc.SybDri
DATABASE_LOGIN_ID	Username to access the database.	
DATABASE_LOGIN_PASS	Password for the above account. You must replace "*@db.password@*" with the actual value.	
KEY STORES: CA_KEY_PASS PEER_KEY_PASS TRUSTED_CA_KEY_PASS SCM_KEY_PASS SSM_KEY_PASS ADMIN_KEY_PASS	Key store passwords used for internal component communications. If left blank, randomly generated passwords are used. Otherwise, provide a password for each entry.	
INSTALL_DB_SCHEMA	Specify whether or not to install the policy database schema.	no
SCM_INTERFACE_LIST	A comma-separated list of IP addresses of the network interfaces to which to bind the Service Control Manager.	169.254.25.129

Perform a Silent Installation

To run the Administration Server installation in silent mode:

1. Copy the modified configuration XML to a location on the machine.
2. Launch the install using the following command:

   Windows: oes320admin_win32.exe -mode=silent -silent_xml=<*path_file*>

   UNIX/Linux: oes320admin_solaris32.bin -mode=silent -silent_xml=<*path_file*>

   where

   <*path_file*>—path and file name of the configuration file

CHAPTER 2

Failover and System Reliability

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

Understanding Failover

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

- Runtime failover — makes sure that SSMs continue to provide security services even if the external components it relies on (such as the authentication database) become unavailable during runtime. This failover mechanism is achieved by configuring secondary sources of information for OES security providers. See Figure 2-1 for an illustration of failover during runtime of an SSM.

- Administration time failover — makes sure that OES administration services are accessible even if the primary Administration Server fails. This failover is achieved by configuring a secondary Administration Server. The secondary server is a redundant server that should be accessed if the primary one cannot be used. Figure 2-2 and Figure 2-3 show how administration failover is achieved using a secondary Administration Server.

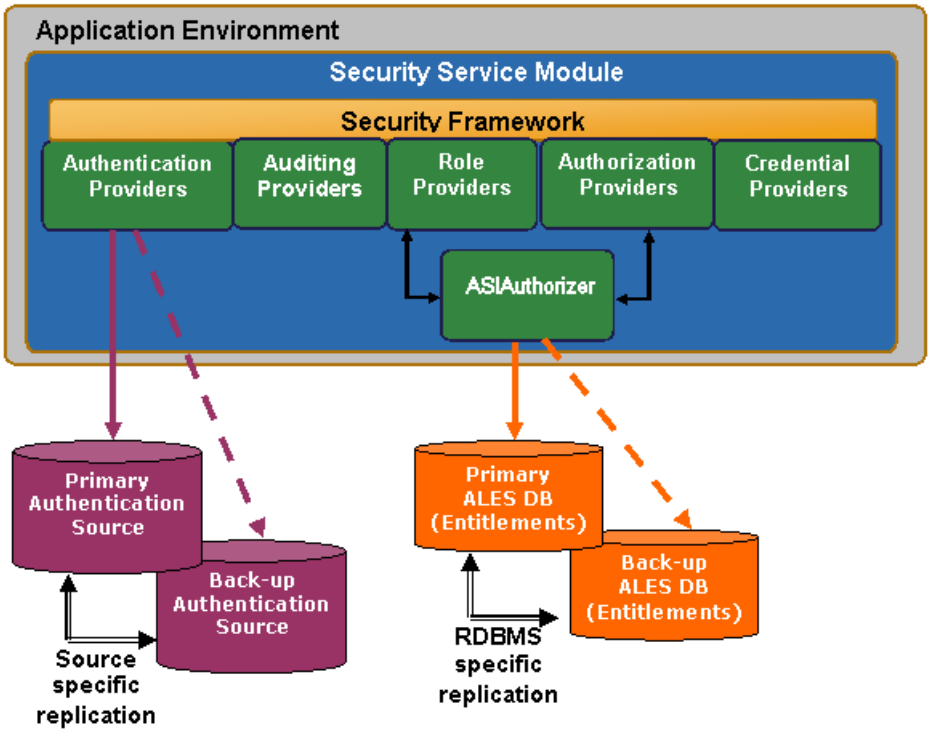
Assuring Runtime Failover for SSMs

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

- Authentication failover is provided by configuring the SSM to point to primary and secondary user data stores. Replication of the 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 domain
- Credential mapping failover is provided by configuring the Database Credential Mapper to use primary and secondary databases.

Note that SSMs have no runtime dependency on the Administration Server.

Figure 2-1 Runtime Availability

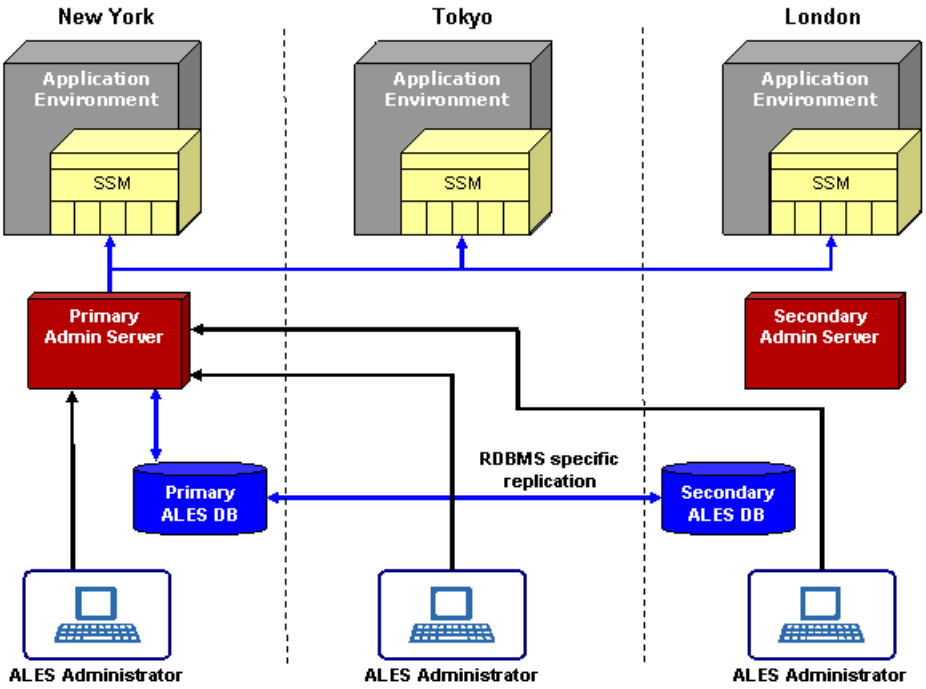


Assuring Administrative Server Availability

Failover for administration functions can be achieved by installing a secondary, redundant Administration Server that will be used only when the primary becomes unavailable.

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

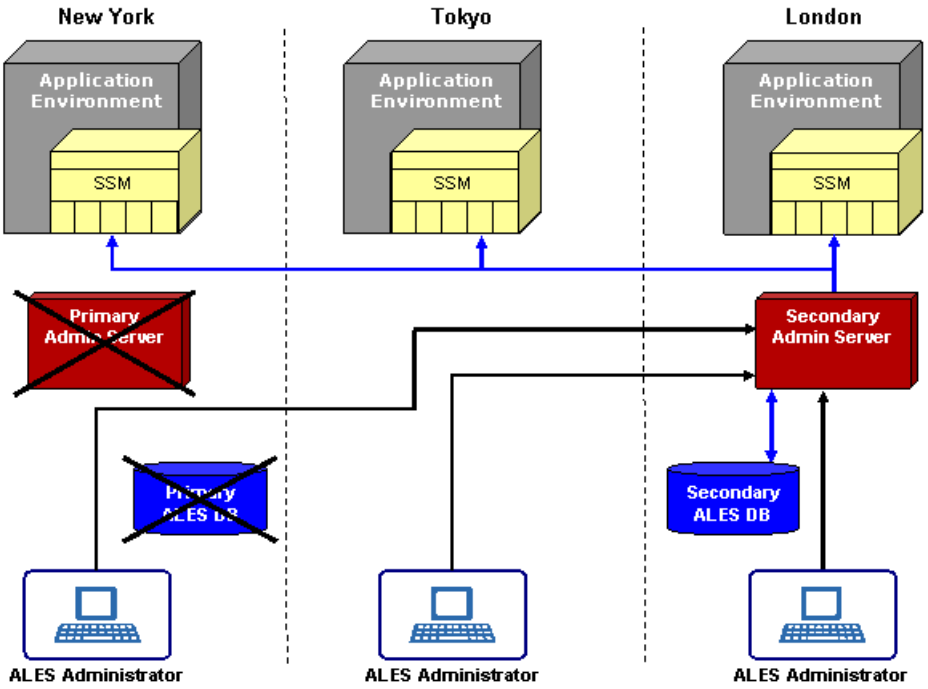
Figure 2-2 Administrative Availability (Working Normally)



If the data center in New York goes down, as illustrated in Figure 2-3, the SSMs detect this failure and connect to the secondary Administration Server. The secondary Administration Server uses the primary database unless it becomes unavailable, in which case the server connects to the secondary database server (the replica).

Note that both the primary and secondary Administration Server can use either the primary or secondary database servers.

Figure 2-3 Administrative Availability (After Failure)



One benefit of the architecture is that even if all Administration Servers, including secondary Administration Servers go down (for maintenance or due to failure), there is no impact on the applications in production or on the security services provided by the SSMs.

For information on how to configure the Administration Server for failover, see "Setting up a Failover Administration Server" on page 2-9.

Failover Considerations for the Database Server

Figure 2-3 provides a logical view of failover functionality when the primary database server fails.

Because the database server contains all the configuration and security data used by the Administration Application to protect applications and resources, it must be highly available and

reliable. This can be accomplished by implementing the recommendations of the database manufacturer (for example, through the use of clustering architecture or hot standby).

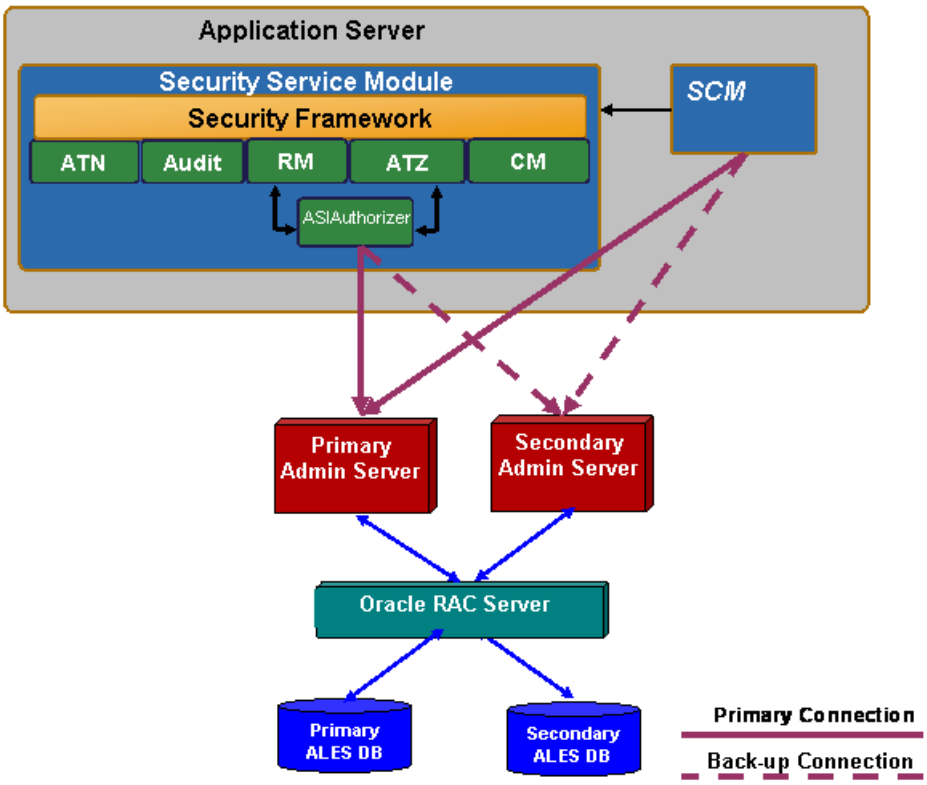
The number of redundant database servers you configure can vary; however, a minimum of two is recommended to maintain reliable services. It is up to the system administrator to set up database failover and configure data replication between the database instances.

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

- Use Oracle RAC for Oracle databases (see Figure 2-4) or implement a similar approach recommended by the database vendor. This allows OES 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, set up the primary database and secondary database with unique connection information. The connection information for the secondary database can be added to the OES Database provider and the Database Credential Mapper. You configure this connection information in the Administration Console on each provider's **Failover** tab.

Figure 2-4 illustrates the failover mechanism for the Administration Server using Oracle RAC.

Figure 2-4 Administration Servers using Oracle RAC



Common methods of achieving high availability include performing periodic back-ups, using fault tolerant disks, and manually copying files whenever they are changed. This is also the case for any optional external data sources in use. A database backup can be used for database recovery in the case of disk failure.

Failover Considerations for SSMs

Use the Administration Console to configure failover support for database-related and LDAP authentication providers by specifying the secondary database or LDAP server, as described in the following sections.

Secondary Databases

A secondary database can be configured for Database Credential Mapping, Database Authentication, ASI Authorization, and ASI Role Mapper providers.

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. This 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 and will also try 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.

Secondary LDAP Servers

A secondary LDAP server can be configured for Novell LDAP, Active Directory, iPlanet, and Open LDAP authenticators.

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

Failover Considerations for SCMs

When a SCM starts, it contacts the Administration Server to obtain and cache the most current configuration data. When configuration data is modified, the Administration Server pushes the updates to the SCM.

Failover for the SCM server is implemented as follows:

- An SCM can configured with addresses for primary and secondary Administration Servers. This can be specified during installation when the install program prompts for this information or specified after installation by modifying the following properties in the SCM's `SCM.properties` file:

  `domain.asi.primary.pdurl` — primary server address
  `domain.asi.secondary.pdurl` — secondary server address

  Note: If a secondary server is not specified during installation, both properties will specify the primary server.

- If no Administration Server is available, the SCM continues to operate using the previously cached policies and configuration data. If the SCM is starting for the first time or does not have a cache, it will continue searching for an Administration Server. Once a primary or

secondary Administration Server is available, the SCM connect to it and obtains its configuration data.

Setting up a Failover Administration Server

The following tasks must be performed to set up a secondary Administration Server to provide failover support:

1. Install the secondary Administration Server on a separate machine as described in "Install a Secondary Administration Server" on page 2-9.

   Note: For complete instructions on installing Administration Servers, see the Administration Server Installation Guide.

2. Initialize the secondary Administration Server's trust stores as described in "Initialize the Secondary Server Trust Stores" on page 2-10.

3. Enable periodic trust synchronization on the secondary Administration Server as described in "Enable Trust Synchronization" on page 2-11.

4. Restart the secondary Administration Server (CP6 and higher).

5. **OPTIONAL** (CP6 AND HIGHER): Set `PD.secondaryEnabled = true` in `BEA_HOME/ales32-admin/config/WLESblm.properties` on the *primary* Administration Server. For more information, see `http://download.oracle.com/docs/cd/E12890_01/ales/docs32/adminref/blmconf_ref.html#wp1164086`.

Install a Secondary Administration Server

The secondary Administration Server must installed on a separate machine and set up in the same manner as the primary Administration Server.

1. Install the container to host Administration Server (WebLogic Server or Apache Tomcat).

2. Run the Administration Server installation program to install the secondary Administration Server. When prompted, provide the information described in Table 2-1.

Table 2-1 Installing the Secondary Administration Server

Prompt	Description
Secondary Server URL	Leave blank
Database Configuration	With the exception of the **Install Database Schema** property, specify the same information specified when the primary Administration Server was installed. **CAUTION:** Be sure to clear the **Install Database Schema** checkbox so that the schema is not installed.
Key Protection Password Selection	It is recommended that you select the **Advanced Password Configuration** option and then specify the passwords when prompted. The password entered do not have to match those on the primary server. If the **Advanced Password Configuration** option is used, the passwords can be used to decrypt SCM and SSM cache files, which may be useful for debugging.

When complete, initialize the secondary trust stores as described in the next section.

Initialize the Secondary Server Trust Stores

When the secondary Administration Server is installed, a set of unique certificates is generated for it. Before starting the server, you must synchronize its trust stores with those in use on the primary Administration Server.

To initialize the secondary Administration Server trust stores:

1. On the secondary Administration Server, create the following directories:

   `OES_HOME/primary-shared-keys`

2. Copy the contents of the `OES_SHARED/keys` directory from the primary Administration Server to the secondary Administration Server machine into the `OES_HOME/primary-shared-keys` directory.

3. From the secondary server's `/bin` directory, execute `initialize_backup_trust.bat|.sh`. When prompted for the primary shared SSL directory, enter the path to the `OES_HOME/primary-shared-keys` directory.

When complete, enable trust synchronization as described in the next section.

Enable Trust Synchronization

Whenever a new SSM/SCM is enrolled — or an existing SSM/SCM is unenrolled — only the trust stores on the primary Administration Server are updated. Unless the secondary Administration Server's trust store is also updated, problems may occur during failover because the secondary Administration Server will not trust the new SSMs. To alleviate this, whenever a new SSM is enrolled, run the `initialize_backup_trust` script file - unless the SSM is installed locally on an existing BEA that is already enrolled.

To insure that the trust stores on secondary Administration Server's are current, the secondary server can be configured to perform periodic synchronization with the primary server's trust stores.

Note: It is very important that the trust synchronization be enabled on the secondary Administration Server only.

To configure the trust synchronization on the secondary Administration Server:

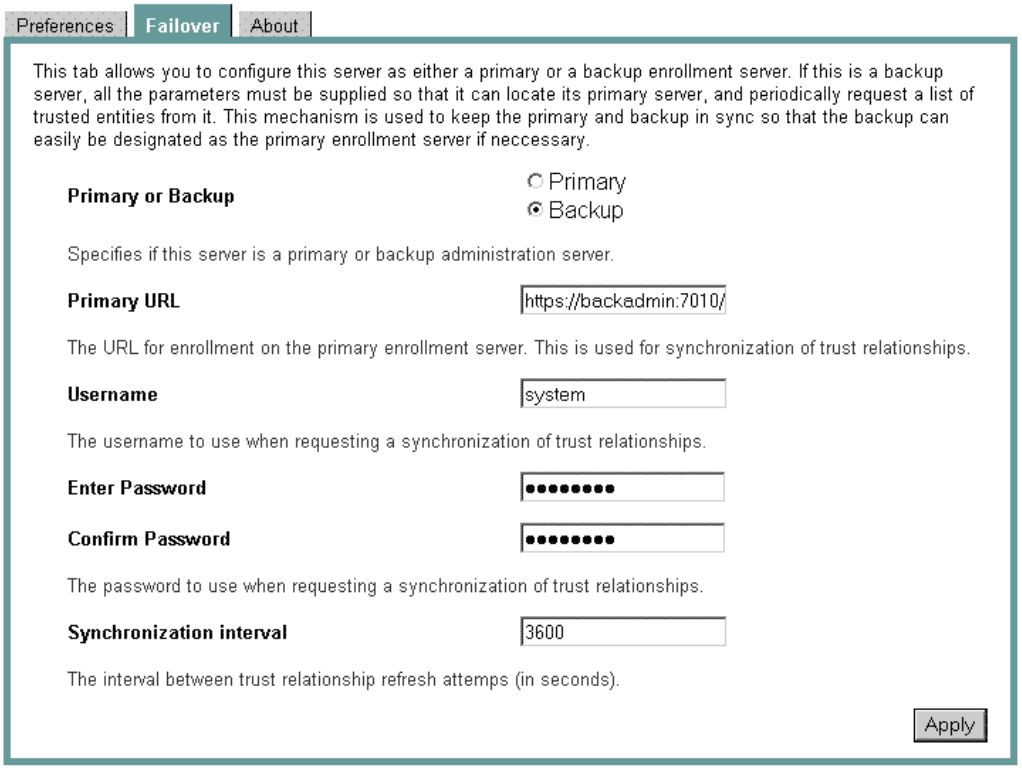
1. Start the secondary Administration Server and access the Administration Console.
2. In the Administration Console, select **Administration Console** at the top of the navigation tree in the left pane.
3. In the right pane, select the **Failover** tab.
4. Select the **Backup** radio button. Then complete the fields as described in Figure 2-2 and click **Apply**.

Table 2-2 Secondary Server's Failover Tab

Field	Description
Primary URL	The URL of the primary Administration Server in the format: `https://<server_name>:7010/asi` where `<server_name>` is the server name or IP address. **Note:** This the same URL used to access the primary server's Administration Console.
Username	Name of administrator user (the default is *admin*).
Enter Password & Confirm Password	Password of the administrator user (the default is *password*).
Synchronization Interval	Number of seconds between synchronization attempts This value depends on how frequently SSM or SCM instances are enrolled and un-enrolled with the primary Administration Server.

A sample completion of the **Failover** tab is shown in Figure 2-5.

Figure 2-5 Configuring a Backup Server in the Administration Console



CHAPTER 3

Collecting Performance Statistics

This section describes the performance statistics feature, which enables collection of data about authentication and authorization for purposes of troubleshooting and performance analysis. It covers the following topics:

Enabling Performance Statistics Collection

The performance statistic feature is controlled by an Auditing security provider, the PerfDBAuditor provider. Performance statistics are gathered for each Security Service Module in your OES installation. In order to collect performance statistics for an SSM, you must enable and configure a PerfDBAuditor provider for that SSM.

Adding a PerfDBAuditor Provider

To add a PerfDBAuditor provider to an SSM other than a WebLogic Server SSM, use the Administration Console. See "Using Performance Statistics with WebLogic Server 9.x\10.0" on page 3-2 for information about how to enable performance statistics collection with the WebLogic Server SSM.

1. Open the **Security Configuration** folder.

2. Open the **Service Control Manager** folder that contains the Security Service Module for which you want to enable performance statistics collection and then open the Security Service Module folder.
3. Open the **Auditing** folder, and click **Auditor**.
4. Click **Configure a new Perf DBAuditor**.
5. On the **General** tab, assign a name for the provider and click **Create**.
6. Click the **Details** tab and configure the PerfDBAuditor. See "Configuring Performance Statistics Collection" on page 3-3 for information about how to set these values.
7. Click **Apply**.

Note: Changes made to a provider do not take effect until after it is explicitly deployed and the associated Security Service Module is restarted.

After you have added a PerfDBAuditor provider to your SSM configuration, you can disable it either by removing it, or by clearing the **Enable Performance Statistics** checkbox on the provider's **Details** configuration page in the OES Administration Console.

Using Performance Statistics with WebLogic Server 9.x\10.0

To add a PerfDBAuditor provider to a WebLogic Server SSM, use the WebLogic Server administration console:

1. In the WebLogic Server Administration Console, navigate to **Security Realms** > *<active security realm>* > **Providers** > **Auditing** and click **New**.

   The Create a New Auditing Provider page appears.
2. In the **Name** field, enter a name for the Auditing provider.
3. From the **Type** dropdown field, select **PerfDBAuditor** as the provider type and click **OK**.
4. Select **Providers** > **Auditing** and click the name of the new Auditing provider to complete its configuration.
5. On the Configuration: Provider-Specific page for the Auditing provider, set the desired values. See "Configuring Performance Statistics Collection" on page 3-3 for information about how to set these values.
6. Click **Save** to save the changes.

7. In the Change Center, click **Activate Changes** and then restart WebLogic Server.

After you have added a PerfDBAuditor provider to your SSM configuration, you can disable it either by removing it, or by clearing the **Enable Performance Statistics** checkbox on the provider's Provider-Specific configuration page in the WebLogic Server Administration Console. You must then restart WebLogic Server for this change to take effect.

Limitations of Performance Statistics in the WebLogic Server SSM

Performance statistics for authorization in the WebLogic Server SSM are available only if you use the ASI Authorization provider. Performance statistics for authentication in the WebLogic Server SSM are not available unless you use the SSM Java API for authentication.

Configuring Performance Statistics Collection

Any changes in the configuration of the PerfDBAuditor provider require restarting the SSM to take effect. You can configure the following settings in the PerfDBAuditor provider:

Basic Behavioral Settings

Performance Statistics Interval
The interval setting specifies data collection interval, in minutes. This determines the length of periods during which the performance statistics data is accumulated before it is dumped to the database tables. All of the internal statistics counters are reset at the beginning of each interval. It should be a positive integer number. Required. The default is 5 minutes.

Performance Statistics Duration
You can collect performance statistics either in circular buffer mode or continuous mode. Circular buffer mode means that, after a specified amount of time elapses, new records are written over the oldest records from the same SSM in the database tables. This prevents performance statistics from growing to an unlimited extent. In continuous mode, records are not overwritten, but there is no limit imposed by the performance statistics feature to the potential size of the database tables.

The Performance Statistics Duration setting specifies whether to operate in circular buffer mode or continuous mode. A positive integer value causes performance statistics to be collected in circular buffer mode and specifies, in minutes, how long the statistics collection proceeds before new records start to overwrite the oldest ones. A special value of 0 means that no loopback will occur; statistics collection proceeds in continuous mode. The value of this field should be either a positive integer number, greater than the interval, or 0, which is the default. It is a required setting.

In either mode, when an SSM is restarted, all previously existing data is cleaned from the database. Performance statistics data is not preserved across SSM restarts.

Enable Performance Statistics
The Enable Performance Statistics checkbox specifies whether the performance statistics collection is enabled or disabled. It serves as a temporary means of disabling the statistics collection without removing the PerfDBAuditor provider from the SSM's configuration. You must restart the SSM after changing this setting before it will take effect. Required. The default is enabled.

Database Connection Settings

JDBC Driver Classname
Specifies which Java class will be used for communication with the database. Required; the default is `oracle.jdbc.driver.OracleDriver`.

JDBC Connection URL
Specifies the connection string to use with the specified driver class. Formats for the database URL and driver class name vary depending on the type of database you are using. For example:

- `jdbc:oracle:thin:@<hostname>:<portnum>:SID` or
- `jdbc:sybase:Tds:<hostname>:<portnum>/<dbname>`

Required.

Database User Login
Specifies the login name of database user with sufficient rights for working with the performance-related tables. This user must possess write and delete privileges for those tables. Required.

Database User Password
The password for the database user specified in the login setting. This password will be stored, in an encrypted form, in the User Store and distributed to the SSM for accessing the database. Required.

JDBC Connection Properties
A parameter for specifying any additional database connection properties that may be needed, in name=value format. Optional.

Database Table Settings

The following specify elements of the database schema used for storing performance statistics data. The default database tables are part of the default OES database schema. If you for some reason need to use different tables, you need to create them yourself in your database schema.

Authentication Statistics Table
The name of the table that contains authentication-related performance statistics. Optional, but at least one of Authentication Statistics Table or Authorization Statistics Table must be present. Default value is PERF_ATH_STAT.

Authorization Statistics Table
The name of the table that contains authorization-related performance statistics. Optional, but at least one of Authentication Statistics Table or Authorization Statistics Table must be present. Default value is PERF_ATZ_STAT.

Authorization Attributes Statistics Table
The name of the table that contains authorization attributes-related performance statistics. Optional. The default value is PERF_ATZ_ATTR_STAT.

Authorization Functions Statistics Table
The name of the table that contains authorization functions-related performance statistics. Optional. The default value is PERF_ATZ_FUNC_STAT.

Using Performance Statistics

The performance statistics feature gathers the following information, for each SSM configuration ID and host name, aggregated for each time interval specified by the Performance Statistics Interval setting:

- Number of requested and successful authentications
- Number of requested and successful authorizations
- Average latency of an authentication request, in milliseconds
- Average latency of an authorization request (the duration of calls to `isAccessAllowed` from start to end), in milliseconds
- For any user attribute required for policy evaluation or response:
  - Average retrieval time, in milliseconds
  - Total number of retrievals

- For each external function called during evaluation:
  - Average execution time, in milliseconds
  - Total number of calls

Performance statistics are stored in the database tables described in "Performance Statistics Database Schema" on page 3-6. To access the performance statistics, use SQL to retrieve the information you are interested in.

When an SSM is restarted, all previously existing data is cleaned from the database. Performance statistics data is not preserved across SSM restarts. Note also that performance statistics entries are uniquely identified by hostname and the configuration ID of the SSM. If you have two SSMs on the same host with the same configuration ID, their performance records will collide and only one will be stored successfully.

Performance Statistics Database Schema

Performance statistics are stored in four tables in the standard OES database schema:

Authentication Statistics Table: PERF_ATH_STAT

This table contains authorization-related performance statistics.

Table 3-1 Authentication Statistics Table: PERF_ATH_STAT

Column	Type	Description
location	varchar(100)	The SSM that is the source of the statistics, recorded as <*hostname*> + <*SSM Configuration ID*> + AthEvent
id	number(12)	A sequential record ID.
starttime	date	The starting time of the interval.
interval	number(12)	The length of the interval in seconds.
totalreq	number(12)	The total number of authentication requests during the interval.
successes	number(12)	The number of successful authentication requests during the interval.
avrlatency	float(10)	Average request latency in milliseconds.

Authorization Statistics Table: PERF_ATZ_STAT

This table contains authorization-related performance statistics.

Table 3-2 Authorization Statistics Table: PERF_ATZ_STAT

Column	Type	Description
location	varchar(100)	The SSM that is the source of the statistics, recorded as <*hostname*> + <*SSM Configuration ID*> + `AtzEvent`
id	number(12)	A sequential record ID.
starttime	date	The starting time of the interval.
interval	number(12)	The length of the interval in seconds.
totalreq	number(12)	The total number of authorization requests during the interval.
successes	number(12)	The number of successful authorization requests during the interval.
avrlatency	float(10)	Average request latency in milliseconds.

Authorization Attributes Statistics Table: PERF_ATZ_ATTR_STAT

This table contains performance statistics related to user attributes required for policy evaluation during authorization.

Table 3-3 Authorization Attributes Statistics Table: PERF_ATZ_ATTR_STAT

Column	Type	Description
location	varchar(100)	The SSM that is the source of the statistics, recorded as <*hostname*> + <*SSM Configuration ID*>> + `AtzAttr`
id	number(12)	A sequential record ID.
name	varchar(100)	The name of the attribute for which statistics are collected.
totalreq	number(12)	The total number of authorization requests requiring this user attribute for evaluation during the interval.
avrlatency	float(10)	Average request latency in milliseconds.

Authorization Functions Statistics Table: PERF_ATZ_FUNC_STAT

This table contains performance statistics related to external functions called during authorization.

Table 3-4 Authorization Functions Statistics Table: PERF_ATZ_FUNC_STAT

Column	Type	Description
location	varchar(100)	The SSM that is the source of the statistics, recorded as <*hostname*> + <*SSM Configuration ID*> + AtzAttr
id	number(12)	A sequential record ID.
name	varchar(100)	The name of the external function for which statistics are collected.
totalreq	number(12)	The total number of authorization requests calling this external function during the interval.
avrlatency	float(10)	Average request latency in milliseconds.

CHAPTER 4

Host Name or IP Address Change

This section describes post-installation configuration changes. The following topics are described:

Configuration for New Host Name

This section describes how to reconfigure components if you change the host name of the system on which the Administration Server is installed.

Administration Server Host Name Change

This section provides instructions for modifying the necessary files when the Administration Server's host name is changed.

1. Shut down the Administration Server and SCM.
2. In *BEA_HOME*/ales32-scm/apps/scm-asi/sar-inf/config.xml, update the host name in the following settings:

```
<listener host="<hostname>"
<proxy best="<hostname>"
<pd best="<hostname>"
)<scm domain="asi" localName="adminconfig", instanceName="SCM.<
hostname>.asi"/>
```

3. In *BEA_HOME*/ales32-admin/bin/WLESadmin.bat, update the host name in the following settings:

   ```
   https://<hostname>:7011/services/ManagedServer
   https://<hostname>:7013
   ```

4. In *BEA_HOME*/ales32-admin/config/WLESWebLogic.conf, update the host name in the following settings:

   a. Change `https://<hostname>` to `https://<hostname>`.

5. In *BEA_HOME*/ales32-admin/config/WLESarme.properties, update the host name in the following settings:

   ```
   https://<hostname>
   instanceName = ARME.admin.server.asi.<hostname>
   ```

6. In *BEA_HOME*/ales32-admin/config/WLESblm.properties, update the host name in the following setting:

   ```
   BLM.host = <hostname>
   ```

7. In *BEA_HOME*/ales32-admin/config/asi.properties, update the host name in the following settings:

   ```
   ASI.BLMAddresses
   ASI.ARMEAddresses
   ASI.PDAddresses
   ```

8. On the SSM machines, update the host name in the following files:

   ```
   BEA_HOME/ales32-ssm/<ssm>/adm/ssm_install.properties
   BEA_HOME/ales32-ssm/<ssm>/template/adm/ssm_instance.properties
   BEA_HOME/ales32-ssm/<ssm>/template/config/WLESarme.properties
   ```

9. Modify the policy database (Oracle and Sybase) to specify the new IP address.

   For Sybase, assuming a username of `wles`, a password of `password`, and a sybase server name of `MYHOST`, perform the following steps:

   a. Start one command window and run `isql -Dwles -Ppassword -SMYHOST`.

   b. Run `select * from engine_addresses` to verify the old address are being used.

c. Delete the records in the `engnie_addresses` table that refer to the old address.

For Oracle:

```
a. sqlplus wles/password@ASI
b. select * from engine_address;
c. truncate table engine_addresses;
```

10. If an SSM was enrolled in secure mode, shut it down and run `enroll.bat secure` to enroll it with the new host name.

SSM Machine Host Name Change

This section provides instructions for modifying the necessary files when a SSM's host name is changed.

1. If the SSM is enrolled in secure mode, run `unenroll.bat secure` **before** you change the host name.

2. Update the following settings in *BEA_HOME*`/ales32-ssm/<`*ssm*`>/adm/silent_install_ssm.xml`

   ```
   <data-value name="SCM_PRIMARY_ADMIN_URL"
   value="https://<hostname>:7010/" />
   ```

   ```
   <data-value name="SCM_BACKUP_ADMIN_URL"
   value="https://<hostname>:7010/" />
   ```

3. Modify *BEA_HOME*`/ales32-ssm/<`*ssm*`>/template/adm/ssm_instance.properties`, update the host name in the following setting:

   ```
   host.name = <hostname>
   ```

   If the host name of the Administration Server machine is also changed, update the `admin.host` setting.

4. For WebLogic SSMs, update the following setting in *BEA_HOME*`/ales32-ssm/<`*ssm*`>/template/bin/set-wls-env.bat`.

   ```
   -Dwles.config.signer=<hostname>
   ```

5. Update the host name in *BEA_HOME*`/ales32-ssm/<`*ssm*`>/template/WLESWeblogic.conf` or `WLESws.wrapper.conf`.

6. In *BEA_HOME*`/ales32-ssm/<`*ssm*`>/template/config/WLESarme.properties`, update the host name in the following settings:

```
PDAddress = https://<hostname>
instanceName = ARME.....asi.<hostname>
```

7. Re-enroll the SSM in secure mode.

Server and SSM On Same Host

This section provides instructions for changing the necessary files when the Administration Server and an SSM are running in separate BEAHOMES on the same machine.

WebLogic Server

In this scenario, the Administration Server and an SSM are running in separate WebLogic servers on the same machine. Each component uses a separate BEAHOME. In the steps that follow the Administration Server is in BEAHOME1 and the SSM is in BEAHOME2.

Follow these steps:

1. If the SSM is enrolled in 'secure' mode, run `unenroll.bat secure`.
2. Shut down all services, including the Administration Server, SCM, and SSM instance. If a component was started in console mode, type CTRL-C to stop it.
3. Modify *BEA_HOME1*`/ales32-scm/apps/scm-asi/sar-inf/config.xml`, update the host name in the following settings:

   ```
   <listener host="<hostname>" port="7013" protocol="https">
   <proxy best="<hostname>" port="7011" protocol="https">
   <pd best="<hostname>" port="7011" protocol="https">
   <scm domain="asi" localName="adminconfig",
   instanceName="SCM.<hostname>.asi"/>
   ```

4. In *BEA_HOME1*`/ales32-admin/config/WLESadmin.bat`, update the host name in the following settings:

   `https://`*<hostname>*`:7011/services/ManagedServer`
   `https://`*<hostname>*`:7013` to `https://`*<hostname>*`:7013`
   `https://`*<hostname>*`:7013`

5. In *BEA_HOME1*`/ales32-admin/config/WLESWebLogic.conf`, update the host name in the following setting:

```
https://<hostname>
```

6. In *BEA_HOME1*/ales32-admin/config/WLESarme.properties, update the host name in the following settings:

   ```
   PDAddress = https://<hostname>
   instanceName = ARME.admin.server.asi.<hostname>
   ```

7. In *BEA_HOME1*/ales32-admin/config/WLESblm.properties, update the host name in the following setting:

   ```
   BLM.host = <hostname>
   ```

8. Modify *BEA_HOME1*/ales32-admin/config/asi.properties to replace the old host name for the following:

   ```
   ASI.BLMAddresses
   ASI.ARMEAddresses
   ASI.PDAddresses
   ```

9. In *BEA_HOME2*/ales32-ssm/<*ssm*>/adm/ssm_install.properties, update the host name in the following settings:

   ```
   host.name = <hostname>
   full.host.name = <full hostname>
   admin.host = <hostname>
   scm.primary.admin.url = https://<hostname>:7010
   scm.backup.admin.url = https://<hostname>:7010
   ```

10. In *BEA_HOME2*/ales32-ssm/<*ssm*>/adm/silient_install_ssm.xml as follows:

    ```
    <data-value name="SCM_PRIMARY_ADMIN_URL"
    value="https://<hostnamename>:7010/" />
    ```

    ```
    <data-value name="SCM_BACKUP_ADMIN_URL"
    value="https://<hostname>:7010/" />
    ```

11. In *BEA_HOME2*/ales32-ssm/<*ssm*>*/template/*adm/ssm_instance.properties, update the host name in the following settings:

    ```
    host.name = <hostname>
    admin.host = <hostname>
    ```

12. For WebLogic SSMs, change the following settings in *BEA_HOME2*/ales32-ssm/<wls *ssm*>*/template/*bin/set-wls-env.bat:

    ```
    -Dwles.config.signer=
    ```

13. Replace the host name in *BEA_HOME2*/ales32-ssm/<*ssm*>/*template*/config/WLESWeblogic.conf or WLESws.wrapper.conf.

14. In *BEA_HOME2*/ales32-ssm/<*ssm*>/*template*/config/WLESarme.properties, replace the host name in the following settings:

    ```
    PDAddress = https://<hostname>
    instanceName = ARME.....asi.<hostname>
    ```

15. Modify the policy database (Oracle and Sybase) to specify the new IP address.

    For Sybase, assuming a username of wles, a password of password, and a sybase server name of MYHOST, perform the following steps:

    a. Start one command window and run `isql -Dwles -Ppassword -SMYHOST`.

    b. Run `select * from engine_addresses` to verify the old address are being used.

    c. Delete the records in the `engnie_addresses` table that refer to the old address.

    For Oracle, the steps are similar to those for Sybase:

    ```
    a. sqlplus wles/password@ASI
    b. select * from engine_address;
    c. truncate table engine_addresses;
    ```

16. If the SSM was enrolled in 'secure' mode, run `enroll.bat secure` to enroll the SSM instance with the new host name.

Apache Tomcat

In this scenario, the Administration Server and an SSM are running in separate Tomcat applications on the same machine. Each component uses a separate BEAHOME. In the steps that follow the Administration Server is in BEAHOME1 and the SSM is in BEAHOME2.

Follow these steps:

1. If the SSM is enrolled in 'secure' mode, run `unenroll.bat secure`.

2. Shut down all services, including the Administration Server, SCM, and SSM instance. If a component was started in console mode, type CTRL-C to stop it.

3. In *BEA_HOME1*/ales32-scm/apps/scm-asi/sar-inf/config.xml update the host name in the following settings:

```
<listener host="<hostname>" port="7013" protocol="https">
<proxy best="<hostname>" port="7011" protocol="https">
<pd best="<hostname>" port="7011" protocol="https">
<scm domain="asi" localName="adminconfig",
instanceName="SCM.<hostname>.asi"/>
```

4. In *BEA_HOME1*/ales32-admin/bin/WLESadmin.bat, update the host name in the following settings:

   ```
   https://<hostname>:7011/services/ManagedServer
   https://<hostname>:7013
   ```

5. In *BEA_HOME1*/ales32-admin/config/WLESTomcat.conf, update the host name in the following setting:

   Change `https://<hostname>`

6. In *BEA_HOME1*/ales32-admin/config/WLESarme.properties, update the host name in the following settings:

   ```
   PDAddress = https://<hostname>
   instanceName = ARME.admin.server.asi.<hostname>
   ```

7. In *BEA_HOME1*/ales32-admin/config/WLESblm.properties, update the host name in the following setting:

   ```
   BLM.host = <hostname>
   ```

8. In *BEA_HOME1*/ales32-admin/config/asi.properties, update the host name in the following settings:

   ```
   ASI.BLMAddresses
   ASI.ARMEAddresses
   ASI.PDAddresses
   ```

9. In *BEA_HOME2*/ales32-ssm/<*ssm*>/adm/ssm_install.properties, update the following settings:

   ```
   host.name = <hostname>
   admin.host = <hostname>
   scm.primary.admin.url = https://<hostname>:7010
   scm.backup.admin.url = https://<hostname>:7010
   ```

10. In *BEA_HOME2*/ales32-ssm/<*ssm*>/adm/silent_install_ssm.xml, update the following:

    ```
    <data-value name="SCM_PRIMARY_ADMIN_URL"
    value="https://<hostname>:7010/" />
    ```

```
<data-value name="SCM_BACKUP_ADMIN_URL"
value="https://<hostname>:7010/" />
```

11. In *BEA_HOME2*/ales32-ssm/*<ssm>/template/*adm/ssm_instance.properties, update the following settings:

    ```
    host.name = <hostname>
    admin.host = <hostname>
    ```

12. Update the host name in *BEA_HOME2*/ales32-ssm/*<ssm>/template/*config/WLESTomcat.conf or WLESws.wrapper.conf.

13. In *BEA_HOME2*/ales32-ssm/*<ssm>/template/*config/WLESarme.properties, update the host name in the following setting:

    ```
    PDAddress = https://<hostname>
    instanceName = ARME.....asi.<hostname>
    ```

14. Modify the policy database (Oracle and Sybase) to specify the new IP address.

    For Sybase, assuming a username of wles, a password of password, and a sybase server name of MYHOST, perform the following steps:

    a. Start a command window and run `isql -Dwles -Ppassword -SMYHOST`.

    b. Run `select * from engine_addresses` to verify the old address are being used.

    c. Delete the records in the engnie_addresses table that refer to the old address.

    For Oracle:

    a. `sqlplus wles/password@ASI`

    b. `select * from engine_address;`

    c. `truncate table engine_addresses;`

15. If the SSM was enrolled in ‘secure’ mode, re-enroll it.

Configuration for a New IP Address

This section describes how to reconfigure OES components if the IP address is subsequently changed. The steps you follow depend on how the OES components are installed:

- “Administration Server IP Address Change” on page 4-9
- “SSM IP Address Change” on page 4-9

Administration Server IP Address Change

If the host IP of the Administration Server and SCM is changed, follow these steps.

1. Shut down the Administration Server and SCM.
2. Update the following settings in *BEA_HOME*/ales32-scm/config/SCM.properties:

   ```
   OS.interface = <ipaddress>
   ```

3. Modify *BEA_HOME*/ales32-ssm/<*ssm*>/adm/ssm_install.properties to replace the old IP address with the new IP address in the following line:

   ```
   scm.interface.list = 10.120.3.140
   ```

4. Restart the SCM, Administration Server, and the SSM.

SSM IP Address Change

If the host IP of the SSM is changed, follow these steps.

1. Modify *BEA_HOME*/ales32-ssm/<*ssm*>/adm/ssm_install.properties to replace the old IP address in the following line:

   ```
   scm.interface.list = 10.120.3.140
   ```

2. Restart the SCM, the Administration Server, and the SSM.

Database Password Changes

This section describes how to change the account password used to access the authentication database or the ASI Authorizer metadirectory database.

Auth Provider Password (WLS SSM)

1. Modify the password for the OES user in the database.
2. Log in to the WebLogic Server console for the `asiDomain`.
3. Specify the new password in the **Database User Password** field under the **Security Realms > asiadmin > Providers > Database Authenticator > Provider Specific** tab.

   Set the Database User Password to match the password you used in Step 1. This must be associated with a valid database login for your database with read access to the tables that involve the authentication process.
4. Click on **Activate Changes**.
5. Run the asipassword utility.
6. Restart the Administration Server.

Auth Provider Password (WLS 8.1 SSM)

1. Modify the password for the OES database user in the database.
2. Change the password in the Administration Console:

a. Open the SSM configuration where the provider is defined.

b. On the **Authentication Providers** tab, select the **Database Authentication** provider.

c. Click the **Details** tab and set the **Database User Password** to match the password used in Step 1. This must be associated with a valid database login for your database with *Read* access to the tables that involve the authentication process.

Enter the same password in the confirmation field.

d. Click **Apply** to save your changes.

Note: Changes made to a provider do not take effect until after it is explicitly deployed and the associated SSM is restarted.

3. Run the asipassword utility.
4. Restart the Administration Server.

ASI Authorizer Password (WLS SSM)

1. Modify the password for the OES user in the database.
2. Log in to the WebLogic Server console for the **asiDomain**.
3. Specify the new password in the Database User Password field under **Security Realms > asiadmin > Providers > ASI Authorizer > Provider-Specific** tab.
4. Click **Activate Changes**.

ASI Authorizer Password (WLS 8.1 SSM)

1. Modify the password for the metadirectory database user in the database.
2. Open the SSM configuration where the provider is defined.
3. On the **Authorization** folder, click **Authorization** and select the **ASI Authorization** provider.
4. Click the **Details** tab and change the **Database Login Password** field for the metadirectory database user.
5. Confirm the password change and click **Apply**.

Note: Changes made to a provider do not take effect until after it is explicitly deployed and the associated SSM is restarted.

6. Run the asipassword utility.
7. Restart the Administration Server.

Resource Discovery

The most challenging aspect of writing policy for an application is discovering all the application resources that must be secured. This process is greatly simplified by running the SSM in 'discovery' mode and then performing one or more user sessions that reflect actual use in the application. Based on the activities performed during the user session, OES will generate an initial policy set to files that can then be imported into OES.

Note: Do not use discovery mode in a production environment. Use it only during development to create the initial security policy.

Enabling Discovery Mode

Resource discovery is enabled by setting the ASI Authorization and ASI Role Mapping providers to run in discovery mode. In this mode, these providers always return 'true' when evaluating user requests and generate the initial policy files based on those requests.

To enable discovery mode, modify the command line that starts the SSM by adding the following system properties:

```
com.bea.security.providers.authorization.asi.AuthorizationProviderImpl.discoverymode=true
```

```
com.bea.security.providers.authorization.asi.RoleProviderImpl.discoverymode=true
```

The system properties are set using the -D switch in the appropriate file. As an example, to enable resource discovery for the WLS SSM, add the following lines to the SSM's `set-wls-env.bat` file:

```
set WLES_JAVA_OPTIONS=%WLES_JAVA_OPTIONS%
-Dcom.bea.security.providers.authorization.asi.AuthorizationProviderImpl.d
iscoverymode=true
set WLES_JAVA_OPTIONS=%WLES_JAVA_OPTIONS%
-Dcom.bea.security.providers.authorization.asi.RoleProviderImpl.discoverym
ode=true
```

For each SSM, Table 6-1 indicates the name and location of the file that must be modified.

Table 6-1 Setting System Properties for Discovery Mode

SSM Type	File Name	Default Location
Java	`set-env.bat (.sh)`	*BEA_HOME*`\ales32-ssm\java-ssm\instance\<`*instancename*`>\bin`
Web Services	`wlesws.wrapper.conf`	*BEA_HOME*`\ales32-ssm\webservice-ssm\instance\<`*instancename*`>\config`
WebLogic Server 8.1	`set-wls-env.bat (.sh)`	*BEA_HOME*`\ales32-ssm\wls8-ssm\instance\<`*instancename*`>\bin`
WebLogic Server 9.x/10.x	`set-wls-env.bat (.sh)`	*BEA_HOME*`\ales32-ssm\wls-ssm\instance\<`*instancename*`>\bin`

Running in Discovery Mode

After enabling discovery mode as described in the previous section, start the secured application. Then perform a user session by logging in to the application, exercising requests for resources, and invoking application functions.

It is important to note that the generated files are meant to serve as a starting point for defining a policy set to fully secure the application. In particular, note the following:

- The recorded policy data is based only on requests made during the user session; no policy data will be generated for parts of the application that are not used.
- Depending on the Resource hierarchy you use to define the application's resources, the imported policy may contain more Resources than actually needed. This can be a particular problem when securing a WLP application. Unlike other applications, access to a WLP resource is denied if not explicitly granted.

When generating the files, user requests are transformed into a policy import format. Under this format, a request consists of four elements: Subject, Resource, Action, Attributes. Each element

has different restrictions on the allowable character set. The providers automatically normalize any invalid characters to produce a valid entry. See Character Restrictions in Policy Data for further details.

Importing the Policy

The files generated by discovery mode will be located in the SSM's domain directory. To import them, use the Policy Import tool.

Once imported, the policy can be managed using the Entitlements Administration Application.

Configuring an XACML Client for a Custom Identity Asserter

This document describes how to configure an XACML client to invoke a custom identity asserter for authentication.

1. Create and build a custom identity asserter.

   Use the sample UsernameIdentityAsserter located under `\ales32-dmin\examples\SampleProviders\UsernameAsserter`.

2. Copy the custom identity asserter (for example, `UsernameIdentityAsserter.jar`) to the following directories:

   a. On the Administration Console machine: `<ALES_ADMIN_HOME>\lib\providers\css`

   b. On the Security Module machine:
   `<ALES_SSM_HOME>\webservice-ssm\lib\providers\css directory`

   If this directory is not there, restart the Admin Server to load the provider.

3. Login to the Administration Console.

4. Click on the webservice ssm instance under Service Control Managers.

   The SSM configuration page is displayed.

5. Choose Java SSM 3.0 - > WS SSM 3.0 for Configuration Version under the General tab and apply the change.

6. Choose “Configure a new UsernameIdentityAsserter ...” in the Authentication Providers section of the Webservice SSM.

7. Choose “USERID_TOKEN” as the active type for this “UsernameIdentityAsserter”.
8. Keep the “Base64Decoding required” checkbox unchecked.
9. Create and apply the previous two changes.

   The token type should be identical to the one you have configured.
10. Click on the “Reorder the Configured Authentication Providers” link on the authentication page and ensure that the “UsernameIdentityAsserter” is at the top.
11. If any other authentication provider has been configured for this Webservice SSM, its control flag should be set to optional.
12. Distribute the SSM configuration change.
13. Shut down the WS-SSM and add the following line into the `<ALES_SSM_HOME>\webservice-ssm\instance\instance-name\config\WLESws.wrapper.conf` configuration file: `Wrapper.java`.

    classpath.70=D:/bea1001/ales32-ssm/webservice-ssm/lib/smwsCustomAssertion.jar
14. Add the following entry to the `<ALES_SSM_HOME>\webservice-ssm\lib\com\bea\security\ssmws\soap\.castor.xml` file.

```
<class name=" com.bea.security.ssmws.credentials.TestCredHolderImpl ">
  <map-to cst:xml="USERID_TOKEN" />
    <field name="cookie" type="java.lang.String" >
      <bind-xml node="text"/>
    </field>
</class>
```

15. Add the following entry to the `<ALES_SSM_HOME>\webservice-ssm\lib\com\bea\security\ssmws\credentials\.castor.xml` file.

```
<class name="com.bea.security.ssmws.credentials.TestCredHolderImpl ">
  <map-to cst:xml="USERID_TOKEN" cst:ns-uri="  @
    http://security.bea.com/ssmws/ssm-soap-types-1.0.xsd " />
    <field name="cookie" type="java.lang.String" >
      <bind-xml node="text"/>
    </field>
</class>
```

16. Add the following entry to the `<ALES_SSM_HOME>\webservice-ssm\lib\com\bea\security\ssmws\authorization\xacml\context\.castor.xml` file.

```
<class name="com.bea.security.ssmws.credentials.TestCredHolderImpl">
  <map-to cst:xml="USERID_TOKEN" cst:ns-uri=" @
    http://security.bea.com/ssmws/ssm-soap-types-1.0.xsd " />
    <field name="cookie" type="java.lang.String">
      <bind-xml node="text"/>
    </field>
</class>
```

17. Add `@ log4j.logger.com.bea.security.ssmws.server=DEBUG` to the `<ALES_SSM_HOME>\webservice-ssm\instance\instance-name\config\log4j.properties` file.

18. Copy the attached ssmwsCustomAssertion.jar to <ALES_SSM_HOME>\webservice-ssm\lib.

19. Restart WS SSM so it can pick up the latest configuration

20. deploy the provider.

After configuration of UsernameIdentityAsserter to your webservice-ssm, you can send XACML Atz request to the ws-ssm using the following XACMLrequest. Make modifications and use the sample XACML client located under ales32-ssm\webservice-ssm\examples\XACMLClient to test this configuration. The sample has to be modified to use the custom token; in this case, the "USERID_TOKEN" instead of the built-in ALESIdentityAsserter. Also ensure you pass the value of the custom token using the AttributeType entity when you construct a SubjectType.

In the subject element of this request, it is set to use USERID_TOKEN as the asserter and the value of the token is passed in <USERID_TOKEN>. Change the value of the token, resource, and action according to your policy and send the request. You should get a XACMLresponse back.

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

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<soapenv:Body>
  <Request xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
    <Subject xsi:type="ns1:SubjectType"
     xmlns:ns1="urn:oasis:names:tc:xacml:2.0:context:schema:os">
      <Attribute
AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
DataType="http://security.bea.com/ssmws/ssm-ws-1.0.wsdl#USERID_TOKEN"
```

```
xsi:type="ns1:AttributeType">
      <AttributeValue xsi:type="ns1:AttributeValueType">
            <USERID_TOKEN
xmlns="http://security.bea.com/ssmws/ssm-soap-types-1.0.xsd">weblogic</
USERID_TOKEN>
          </AttributeValue>
        </Attribute>
        <Attribute
AttributeId="http://security.bea.com/ssmws/ssm-ws-1.0.wsdl#level"
DataType="http://www.w3.org/2001/XMLSchema#string"
xsi:type="ns1:AttributeType">
          <AttributeValue
xsi:type="ns1:AttributeValueType">3</AttributeValue>
        </Attribute>
      </Subject>
      <Resource xsi:type="ns3:ResourceType"
xmlns:ns3="urn:oasis:names:tc:xacml:2.0:context:schema:os">
        <Attribute
AttributeId="urn:oasis:names:tc:xacml:2.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#string"
xsi:type="ns3:AttributeType">
          <AttributeValue
xsi:type="ns3:AttributeValueType">MyApp/stock/app</AttributeValue>
        </Attribute>
      </Resource>
      <Action>
        <Attribute
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"
xsi:type="ns4:AttributeType"
xmlns:ns4="urn:oasis:names:tc:xacml:2.0:context:schema:os">
```

```
          <AttributeValue
xsi:type="ns4:AttributeValueType">any</AttributeValue>
        </Attribute>
      </Action>
      <Environment/>
    </Request>
  </soapenv:Body>
</soapenv:Envelope>
```

CHAPTER 8

Running the Java SSM in Java Development Environments

The section describes how to run the Java SSM in WebSphere RAD and Eclipse development environments. This includes instructions for setting up sample applications that are provided when the Java SSM is installed.

WebSphere RAD Environments

This section demonstrates how to run the Java SSM in WebSphere RAD using the sample application provided in BEA_HOME\ales32-ssm\java-ssm\examples\JavaAPIExample. These steps were performed using WebSphere RAD 6.0.0 with the pre-installed IBM JDK.

1. Follow the README in BEA_HOME\ales32-ssm\java-ssm\examples\JavaAPIExample to setup the sample policies for this example.
2. Start WebSphere RAD in a new workspace and create a new Java project.
3. Right-click the project and select **Import** to import the Java file from the BEA_HOME\ales32-ssm\java-ssm\examples\JavaAPIExample\src directory into the new project.
4. Add all of the ALES libraries used by the BEA_HOME\ales32-ssm\java-ssm\examples\JavaAPIExample\config\run.bat file to the **Java Build Path** in the Project Properties. To do this:

a. Right-click the project and select **Properties**.

b. Click **Java Build Path**.

c. On the **Libraries** tab, click **Add External JARs** and add the jar files located in the following directories:

`BEA_HOME\ales32-ssm\java-ssm\lib` (except for `pdsoap1.jar`)
`BEA_HOME\ales32-ssm\java-ssm\lib\providers\ales`

d. Click **OK** and allow the project to build. All errors should be resolved.

5. To setup the arguments for the `main()` method, right-click the Java file and select **Run**. Then select **Run** with the green arrow icon.

6. In the **Run** dialog, select **Java Application**. Then click **New**. and supply a well-defined name for the configuration.

7. Click **Search** to find the Java class with the main method. In this example it would be `com.bea.security.examples.JavaAPIExample`.

8. On the **Arguments** tab, paste in the command line options used by run.bat (in the `BEA_HOME\ales32-ssm\java-ssm\examples\JavaAPIExample`) into the **VM arguments** field.

   Example of the settings used:

```
-Dwles.scm.port=7013 -Dwles.arme.port=8100
-Dwles.config.signer=<HOSTNAME>
-Dlog4j.configuration="file:C:/bea/ales32-ssm/java-ssm/instance/jssm/co
nfig/log4j.properties" -Dlog4j.ignoreTCL=true
-Dwles.ssl.passwordFile="C:/bea/ales32-shared/keys/password.xml"
-Dwles.ssl.passwordKeyFile="C:/bea/ales32-shared/keys/password.key"
-Dwles.ssl.identityKeyStore="C:/bea/ales32-shared/keys/identity.jceks"
-Dwles.ssl.identityKeyAlias=wles-ssm
-Dwles.ssl.identityKeyPasswordAlias=wles-ssm
-Dwles.ssl.trustedCAKeyStore="C:/bea/ales32-shared/keys/trust.jks"
-Dwles.ssl.trustedPeerKeyStore="C:/bea/ales32-shared/keys/peer.jks"
-Djava.io.tmpdir="C:/bea/ales32-ssm/java-ssm/instance/jssm/work/jar_tem
p"
-Darme.configuration="C:/bea/ales32-ssm/java-ssm/instance/jssm/config/W
LESarme.properties"
-Dales.blm.home="C:/bea/ales32-ssm/java-ssm/instance/jssm"
-Dkodo.Log=log4j -Dwles.scm.useSSL=true
-Dwles.providers.dir=C:/bea/ales32-ssm/java-ssm/lib/providers
```

9. While in the **Arguments** tab, clear the **Use default working directory** checkbox.

10. Select the **File System** and browse to the build directory of the JavaAPIExample.

    For example,
    `BEA_HOME\ales32-ssm\java-ssm\examples\JavaAPIExample\build\config`.

11. On the **Classpath** tab, select the **User Entries** node on the Classpaths tree. Then click the **Advanced** button and select **Add External Folder** and add the following external folders:

    ```
    BEA_HOME
    BEA_HOME\ales32-ssm\java-ssm\instance\jssm\config
    ```

12. Click **Apply** and then **Run**.

    NOTE: If an exception like the following appears...

    ```
    com.bea.security.management.ConfigurationException: Error initializing
    the SCM SSL context. at
    com.bea.security.internal.css.SCMConfiguration.configureRealm(SCMConfig
    uration.java:512)
    ```

    ...then do the following to switch to a standard JDK installation:

    a. Select **Window->Preferences->Java -> Installed JREs**.

    b. Select **add** and add a JVM (Sun or BEA JRockit).

    c. Select OK and check the newly added JVM.

13. Enter in the arguments to see the example correctly authorizing a subject on a resource.

Troubleshooting

To troubleshoot any problems, enable verbose debugging on the Java-SSM and examine the log for the authorization events. To do this:

1. Open `logj.properties` in the SSM instance's `config` directory in an editor and uncomment the following lines:

   log4j.logger.com.bea.security.providers.authorization = DEBUG
   log4j.logger.com.wles.util.DebugStore=DEBUG

2. Clear out the log files in the SSM instance's `log` directory.

3. Re-run the Java SSM example and examine `system_console.log` in the instance's `log` directory for authorization events, such as:

   ```
   2008-04-14 15:03:28,343 [main] DEBUG
   com.bea.security.providers.authorization.asi.ARME.evaluator.BoolEvaluat
   ```

```
or - evaluateRuleArray(): Evaluate policy:
4405:grant(//priv/buy,//app/policy/jssm/store/book,//role/borrower)
2008-04-14 15:03:28,343 [main] DEBUG
com.bea.security.providers.authorization.asi.ARME.evaluator.BoolEvaluat
or - boolEvaluate() entered for rule with condition: true
2008-04-14 15:03:28,343 [main] DEBUG
com.bea.security.providers.authorization.asi.ARME.evaluator.BoolEvaluat
or - constraint evaluation result is: true
2008-04-14 15:03:28,343 [main] DEBUG
com.bea.security.providers.authorization.asi.ARME.evaluator.BoolEvaluat
or - append return attributes.
2008-04-14 15:03:28,343 [main] DEBUG
com.bea.security.providers.authorization.asi.ARME.evaluator.BoolEvaluat
or - boolEvaluate(): Evaluation result size is 0.
2008-04-14 15:03:28,343 [main] DEBUG
com.bea.security.providers.authorization.asi.ARME.evaluator.BoolEvaluat
or - evaluateRuleArray(): Evaluate result: true
2008-04-14 15:03:28,343 [main] DEBUG
com.bea.security.providers.authorization.asi.ARME.evaluator.BoolEvaluat
or - evaluateGrantPolicy result: true
2008-04-14 15:03:28,343 [main] DEBUG
com.bea.security.providers.authorization.asi.ARME.evaluator.BoolEvaluat
or - authEvalWorker: evalute with roles return GRANT
2008-04-14 15:03:28,343 [main] DEBUG com.wles.util.DebugStore -
queryAccess: DebugStore:

========== Policy Evaluation Info ==========
RequestResource is: //app/policy/jssm/store/book
UserInfo:
    Name: //user/asi/system/
    Groups: //sgrp/asi/allusers/
Resource Present: true
Roles Granted: //role/borrower //role/EntitlementsAdmin
Role Mapping Policies:
    1. Result: true; Policy Type: grant
        Role: //role/borrower
        Resource: //app/policy/jssm/store
        Subject: //user/asi/system/
        Constraints: canbuy="yes"
        Evaluated Attributes and Functions:
            sys_user(dynamic) = system
            canbuy(dynamic) = yes

    2. Result: true; Policy Type: grant
        Role: //role/EntitlementsAdmin
        Resource: //app/policy
        Subject: //user/asi/system/
        Constraints: NONE
```

```
ATZ Policies:
    1. Result: true; Policy Type: grant
        Privilege: //priv/buy
        Resource: //app/policy/jssm/store/book
        Subject: //role/borrower
        Constraints: NONE

========== Policy Evaluation Info ==========
2008-04-14 15:03:28,343 [main] DEBUG
com.bea.security.providers.authorization.asi.ARME.engine.ARME - unlock
policy lock for read
2008-04-14 15:03:28,343 [main] DEBUG
com.bea.security.providers.authorization.asi.AuthorizationProviderImpl
- result is GRANT
2008-04-14 15:03:28,359 [main] DEBUG
com.bea.security.providers.authorization.asi.AccessResultLogger -
Subject Subject:
Principal: system
 privilege buy resource //app/policy/jssm/store/book result PERMIT
```

Eclipse

In Eclipse, create a Java Project from using the sample found in the following directory:

`BEA_HOME/ales32-ssm/JavaSSM/Examples/JavaAPIExample`

1. Create a Java Project and create project from existing source:
2. Set the CLASSPATH in Java Project based on `set-env.bat` of SSM instance. Copy the following external jar files into Java Build Path of Eclipse project:

```
%INSTALL_HOME%\lib\api.jar
%INSTALL_HOME%\lib\css.jar
%INSTALL_HOME%\lib\asn1.jar
%INSTALL_HOME%\lib\saaj.jar
%INSTALL_HOME%\lib\framework.jar
%INSTALL_HOME%\lib\scmapi.jar
%INSTALL_HOME%\lib\log4j.jar
%INSTALL_HOME%\lib\jmx.jar
%INSTALL_HOME%\lib\connector.jar
%INSTALL_HOME%\lib\asi_classes.jar
%INSTALL_HOME%\lib\EccpressoCore.jar
%INSTALL_HOME%\lib\EccpressoJcae.jar
%INSTALL_HOME%\lib\jsafe.jar
%INSTALL_HOME%\lib\jsafeJCE.jar
%INSTALL_HOME%\lib\jsafeFIPS.jar
%INSTALL_HOME%\lib\jsafeJCEFIPS.jar
```

```
%INSTALL_HOME%\lib\sslplus.jar
%INSTALL_HOME%\lib\ssladapter.jar
%INSTALL_HOME%\lib\wlcipher.jar
%INSTALL_HOME%\lib\asitools.jar
%INSTALL_HOME%\lib\process.jar
%INSTALL_HOME%\lib\webservice.jar
%INSTALL_HOME%\lib\webserviceclient.jar
%INSTALL_HOME%\lib\org.mortbay.jetty.jar
%INSTALL_HOME%\lib\javax.servlet.jar
%INSTALL_HOME%\lib\org.apache.jasper.jar
%INSTALL_HOME%\lib\sslserver.jar
%INSTALL_HOME%\lib\sslclient.jar
%INSTALL_HOME%\lib\pdsoap.jar
%INSTALL_HOME%\lib\antlr.jar
%INSTALL_HOME%\lib\ld-server-core.jar
%INSTALL_HOME%\lib\wlsdo.jar
%INSTALL_HOME%\lib\wlxbean.jar
%INSTALL_HOME%\lib\xbean.jar
%INSTALL_HOME%\lib\xqrl.jar
%INSTALL_HOME%\lib\ld-client.jar
%INSTALL_HOME%\lib\rmi-ssm.jar
%INSTALL_HOME%\lib\rmi-types.jar
%INSTALL_HOME%\lib\axis.jar
%INSTALL_HOME%\lib\commons-logging-1.0.4.jar
%INSTALL_HOME%\lib\commons-discovery-0.2.jar
%INSTALL_HOME%\lib\wsdl4j-1.5.1.jar
%INSTALL_HOME%\lib\jaxrpc.jar
%INSTALL_HOME%\lib\providers\ales\CR338979_414_jdk1.4.jar
%INSTALL_HOME%\lib\providers\ales\kodo-runtime.jar
%INSTALL_HOME%\lib\providers\ales\jdo.jar
%INSTALL_HOME%\lib\providers\ales\openjpa.jar
%INSTALL_HOME%\lib\providers\ales\commons-lang-2.1.jar
%INSTALL_HOME%\lib\providers\ales\jta-spec1_0_1.jar
%INSTALL_HOME%\lib\providers\ales\openjpa.jar
%INSTALL_HOME%\lib\providers\ales\commons-collections-3.2.jar
%INSTALL_HOME%\lib\providers\ales\commons-pool-1.3.jar
%INSTALL_HOME%\lib\providers\ales\serp.jar
```

3. Add the `BEA_HOME` and `BEA_HOME\ales32-ssm\java-ssm\<instance>\config` directories into the Eclipse classpath.

   Note: If Eclipse reports, "*Cannot not nest the directory inside library <BEA-HOME>* ", copy `license.bea` into the instances `config` directory.

4. Set the following run configurations in the project:

- Working directory — `BEA_HOME\ales30-ssm\java-ssm\example\JavaAPIExample\build\config`
- VM arguments based on `%JAVA-OPTIONS%` of `set-env.bat`.

Troubleshooting

Note the following error conditions and resolution:

AXIS SOAP compatibility issue:
'java.lang.NoSuchFieldError: RPC'
The issue is caused by AIXS SOAP stack compatibility between different AIXS version. Remove `pdsoap1.jar` from class path.

License Check:
'Got exception in reading the license file'.
Make sure `license.bea` file is in class path.

XML Parsing:
'java.lang.NoSuchMethodException: org.apache.axis.encoding.ser.ArraySerializerFactory. create(java.lang.Class, javax.xml.namespace.QName)'
Only the jar files set in `set-env.bat` should be included in the project. Remove all other jar files. Files like the following are not needed and should be removed:

- `com.bea.core.common.security.opensaml2_4.0.0.0.jar`,
- `com.bea.core.xml.beaxmlbeans_2.2.0.0.jar`
- `javax.xml.stream_1.0.0.0.jar`
- `xml-apis.jar`

CHAPTER 9

Debugging SSL Connectivity

The SSL Diagnosis Tool can help to debug SSL connectivity issues when using Oracle Entitlements Server (OES); for example 'BAD_CERTIFICATE'. The tool checks the OES SSL configuration in on the Security Module (SM) side and displays detailed SSL handshake information. This document contains information on how to use the tool.

Using the SSL Diagnosis Tool

The SSL Diagnosis Tool should be executed from the SM directory located in `ales32-shared/bin`. Run the script as follows:

```
ssldiagnosis.bat|sh <demo|secure>
```

Choose the Demo option to check SSL certificates created by the demo CA certificate from `DemoTrust.jks` key store.

Choose the Secure option to check SSL certificates created by using the CA certificate from the `cacerts file` in the *`BEA_HOME`*`/jdk-version/jre/lib/security` directory.

Running the SSL Diagnosis Tool

Use the following procedure to run the SSL Diagnosis Tool. Ensure that the OES Administration Server is running before beginning this procedure.

1. Open a terminal window.
2. Change to the *`BEA_HOME`*`/ales32_shared/bin` directory.

3. Run `ssldiagnosis.bat|sh demo`.
4. Enter the Administration Server administrator username and password at the enrollment prompt.
   The default values are *admin* and *password* respectively.
5. Check DEMO CA.
   The default password for demo CA is *password* and the default CA alias name is *alesdemoca*.
6. Check OES Certificates in keystore files.
7. Check OES components.
   PD and SCM belongs to SM: give directory of any SSM. For example, Java-SSM location value is BEA_HOME/ales32-ssm/java-ssm

To Display SSL Handshake Information

To display additional debug messages, set the following properties for the OES Administration Server based on the container in which it is running.

On Tomcat

1. Modify the `WLESTomcat.conf` file by setting the following property:
   `-Djavax.net.debug=ssl`
2. Modify the `log4j.properties` file by setting the following property:
   `log4j.logger.com.bea.security.ssl = debug`

On WebLogic Server

1. Modify the `WLESWebLogic.conf` file by setting the following property:
   `-Dssl.debug=true -Dweblogic.StdoutDebugEnabled=true`
2. Modify the `log4j.properties` file by setting the following property:
   `log4j.logger.com.bea.security.ssl = debug`

CHAPTER 10

Debugging Policies

This document desscribes how to set an SSM instance's logs to record debugging-level events having to do with authentication, role mapping, and authorization.

Overview

When policy outcomes are other than expected, it may be useful to enable policy debugging so that the SSM's logs will capture all events related to policy decisions. The logged information may policy-related details, such as failed authentications, missing group memberships, incorrect role assignments, and others.

Caution: SSM performance can be severely impacted when debug flags are enable. In production environments turn on debug flags only when necessary.

Enabling Policy Debugging

Policy debugging is enabled by changing settings in the SSM instance's `log4j.properties` files. To do this:

1. In the SSM instance's `config` directory, open `log4j.properties` in an editor.
2. To turn on authentication debugging, uncomment the following line:

   ```
   log4j.logger.com.bea.security.providers.authentication = DEBUG
   ```

3. To turn on role mapping and authorization debugging, uncomment the following lines:

   ```
   log4j.logger.com.bea.security.providers.rolemapper = DEBUG
   log4j.logger.com.bea.security.providers.authorization = DEBUG
   log4j.logger.com.wles.util.DebugStore=DEBUG
   ```

4. Restart the SSM.

Event Logs

This section describes common policy-related events that may be captured when in debugging mode.

Authentication

For authentication events, check for the events shown in Table 10-1:

Table 10-1 Authentication Events

Event	Description
Usernames	Usernames should match those supplied to the SSM. Usernames are logged as follows: `DBMSAtnLoginModuleImpl - Login username: <username>`
Identity Directory	Check that the identity directory name is correct. Directory names are logged as follows: `DefaultDBMSPluginImpl - Formatted User: //user/<directory>/<username>/`

Table 10-1 Authentication Events

Event	Description
Authentications	The following message indicates a successful authentication: `DBMSAtnLoginModuleImpl - Authenticated User <username>`
Groups	The following message indicates a user's group memberships: `odbms.DBMSAtnLoginModuleImpl - Groups Found: <list-of-groups>`

Role Mapping

For role mapping events, check for the events shown in Table 10-2:

Table 10-2 Role Mapping Events

Event	Description
Roles	The following entry denotes the entry point for evaluating the roles. <*username*> is the user name supplied to the application and <*resource*> is the name of the queried resource. `BoolEvaluator - Query roles entered for //user/asi/<username>/://app/policy/<resource>`
Role Policies	Make sure all relevant policies are evaluated. The following is a sample logged event: `BoolEvaluator - evaluateGrantDenyRoles: evalute grant policy: 3600:grant (//role/<role>,//app/policy/<resource>, //user/<username>/)`
Constraint Evaluations	The following is a sample message indicating a constraint evaluation: `BoolEvaluator - constraint evaluation result is: true`
Roles Granted	The following is a sample message indicating a role assignment: `BoolEvaluator - Role //role/<role> was granted`

Authorization

For authorization events, check for the events shown in Table 10-3:

Table 10-3 Authorization Events

Event	Description

Authorization	The following entry indicates the authorization policy evaluated:
	`BoolEvaluator - evaluateRuleArray():`
	`Evaluate policy: 3401:grant`
	`(//priv/buy,//app/policy/javaapi_app/store/book,`
	`//role/borrower)`

Constraint Evaluations	The following is a sample message indicating a constraint evaluation: `BoolEvaluator-constraint evaluation result: true`

Roles Granted	The following is a sample message indicating a authorization policy evaluation:
	`BoolEvaluator - authEvalWorker:`
	`evalute with roles return GRANT`

Sample Log Messages

========== Policy Evaluation Info ==========
RequestResource is: //app/policy/<resource>
UserInfo:
Name: //user/<identity-directory>/<user-name>
Groups: //sgrp/<identity-directory>/<group-name>

Resource Present: true
Roles Granted: //role/<granted-roles>

Role Mapping Policies:
1. Result: true; Policy Type: grant
Role: //role/<requested-role>
Resource: //app/policy/<resource>
Subject: //user/<identity-directory>/<user-name>

```
    Constraints: (some-variable = "some-value")
    Evaluated Attributes and Functions:
            some-variable(identity) = some-value

ATZ Policies:
1. Result: true; Policy Type: grant
    Privilege: //priv/<requested-privilege>
    Resource: //app/policy/<resource>
    Subject: //role/<granted-role>
    Constraints: NONE

        ========== Policy Evaluation Info ==========
```

Debug API for Java-SSM

The following 2 API calls are specific to Java-SSM only. To enable policy debugging, open `BEA_HOME/ales30-ssm/java-ssm/<instancename>/jssm/config/WLESarme.properties` in an editor and set the following:

```
SsmPolicyTrace=true
```

Note: Enabling debugging weakens security, because OES policy evaluations will be visible to Java programs. For example, a malicious Java program could make "_Debug()" calls to gain information about policies.

To capture debugging data:

1. Create a DebugInfo object as follows:

   ```
   DebugInfo debugInfo = new DebugInfo();
   ```

2. Call getRoles_Debug() to obtain role assignments, as follows:

   ```
   getRoles_Debug(AuthenticIdentity ident, RuntimeResource resource,
   RuntimeAction action, AppContext context, DebugInfo debugInfo)
   ```

3. Call isAccessAllowed_Debug() to obtain information about the policies used to reach a decision, as follows:

   ```
   isAccessAllowed_Debug(AuthenticIdentity ident, RuntimeResource
   resource, RuntimeAction action, AppContext context, DebugInfo
   debugInfo)
   ```

4. Print the DebugInfo object to console:

   ```
   System.out.println(debugInfo.toString());
   ```

For further information, see the sample provided in the `BEA_Home/ales32-ssm/java-ssm/examples/JavaAPIExample` directory.

CHAPTER 11

Running Multiple Administration Servers on One Machine

Running multiple OES Administration Servers on one machine may be useful in development environments. This can be done by installing the additional servers in separate BEA_HOMES and then modifying the Window's service names of the added server and SCM to unique values.

Instructions

Follow these instructions:

1. To host the additional OES Administration Server, install an additional WebLogic or Tomcat server in a new BEA_HOME.

   Note: In these instructions, the new BEA_HOME is *BEA_HOME2*.

2. Install the additional OES Administration Server.

   Note: During installation, be sure to clear the **Install Database Schema** checkbox.

3. After installation, open `BEA_HOME2/SCM_INSTALL/bin/WLESscm.bat` in an editor and change the SCM service name in line 13 to a unique name. For example:

   ```
   set SERVICENAME="ALES Service Control Manager2"
   ```

4. Open `BEA_HOME2/SCM_INSTALL/config/WLESscm.conf` in an editor and specify unique values as described below:

   line 145 — `wrapper.ntservice.name=SCM2`

   line 148 — `wrapper.ntservice.displayname=ALES Service Control Manager2`

   line 151 — `wrapper.ntservice.description=ALES Service Control Manager2`

5. Change to BEA_HOME2/SCM_INSTALL/bin and run the following:

   ```
   WLESscm.bat register
   ```

   If the registration is successful, the new service will appear in the Window's Services applet.

6. Open BEA_HOME2/ales32-admin/bin/WLESadmin.bat and change line 13 to specify the same value set in the service name from step 3 above. For example:

   ```
   set SCMSERVICENAME="ALES Service Control Manager2"
   ```

7. Do one of the following:

   a. If using WebLogic server, open BEA_HOME2/ales32-admin/bin/WLESWebLogic.bat and modify line 13 as shown below:

      ```
      set SERVICENAME=ALES.WLS2.<host>
      ```

   b. If using Tomcat, modify line 13 in BEA_HOME2/ales32-admin/bin/WLESTomcat.bat as follows:

      ```
      set SERVICENAME=ALES.TOMCAT2.<host>
      ```

8. Do one of the following:

   a. If using WebLogic server, open BEA_HOME2/ales32-admin/config/WLESWebLogic.conf and modify the following lines as shown:

      line 186 — `wrapper.ntservice.name=ALES.WLS2.<host>`

      line 189 — `wrapper.ntservice.displayname=ALES WLS2.<host>`

      line 192 — `wrapper.ntservice.disscription=ALES WeblogicServer - WLS2.<host>`

   b. If using Tomcat, modify the following lines in BEA_HOME2/ales32-admin/config/WLESTomcat.conf:

      line 191 — `wrapper.ntservice.name=ALES.TOMCAT2.<host>`

      line 194 — `wrapper.ntservice.displayname=ALES TOMCAT2.<host>`

      line 197 — `wrapper.ntservice.description=ALES Tomcat Server2`

9. Go to BEA_HOME2/ales32-admin/bin and run the following:

   ```
   WLESadmin.bat register
   ```

If the registration is successful, the new service will appear in the Window's Services applet.

10. Remaining in `BEA_HOME2/ales32-admin/bin`, run the following to install the initial policies:

    ```
    install_ales_schema.bat <dbuser> <dbuser_password>
    ```

CHAPTER 12

Pointing the Administration Server to a Different Database

In the event of an unplanned database failure in a development environment, it may be desirable to point the Administration Server to a different database — referred to in this document as the 'target' database. This can be done by modifying information in a number of Administration Server files. These instructions are found in the following sections.

Requirements

The instructions in this section assume that the ALES schema is installed on the target database and that the target database is the same database product (for example, Oracle 10).

Instructions

1. Make sure the target database has the ALES schema installed.
2. Modify the necessary Administration Server files as described in Table 12-1.

Table 12-1 Changes to Administration Server Files

File	Property	Description
ales32_admin\config\ database.properties	ConnectionURL	Change the JDBC connection string to the target database in the following format: jdbc:oracle:thin:@<*host*>:<*port*>:<*sid*> <*host*>— name/IP of the database machine <*port*>—database listener port number <*sid*>—database SID Example: javax.jdo.option.ConnectionURL: jdbc:oracle:thin:@acmedb.acme.com:1521:orcl
	ConnectionUserName	If necessary, change the name of the OES database user in the target database. **Note:** Not necessary if the target database uses the same database username (default = *admin*). Example: javax.jdo.option.ConnectionUserName: admin
	ConnectionPassword	Add the following line after the ConnectionUserName entry: javax.jdo.option.ConnectionPasssord: <*dbuser_password*> where <*dbuser_password*> is the password of the OES db user in the target database (default = *password*). **Notes:** • The existing file does not contain this line. • These instructions are for development environments so the password may be stored in clear text if local security policy permits. If not, use the following procedure to encrypt the password and specify the encrypted value. **a.** Define the ConnectionPassword value with the clear text password and complete this procedure through step 3 to ensure that the server starts. **b.** Use asipassword to encrypt the password. **c.** Once the server starts, login to the asi console and reset the DatabaseAuthenticator with the new password. **d.** Remove the clear text password from database.properties. **e.** Continue the parent procedure from step 4.

Table 12-1 Changes to Administration Server Files

File	Property	Description
ales32_admin\config\ admin_install.properties	db.login	If necessary, change the name of the OES database user in the target database. **Note:** Not necessary if the target database uses the same username for the OES database user (default = *admin*). Example: `db.login = admin`
	db.jdbc.url	Change the JDBC connection string to the target database in the following format: jdbc:oracle:thin:@<*host*>:<*port*>:<*sid*> <*host*>— name/IP of the database machine <*port*>—database listener port number <*sid*>—database SID Example: javax.jdo.option.ConnectionURL: jdbc:oracle:thin:@acmedb.acme.com:1521:orcl

Table 12-1 Changes to Administration Server Files

File	Property	Description
ales32_admin\config\ asiadmin.xml	DatabaseAuthenticator properties: config__database_user_login config__database_user_password	If necessary, change the name and password of the OES database user in the target database within the <object id="DatabaseAuthenticator"> definition. To encrypt the password, use ales32-admin\bin\encrypt_password.bat|sh. **Note:** Not necessary if the target database uses the same username and password for the OES database user (default = *admin, password*). Example: <property name="config__database_user_login"> <value>"admin"</value> </property> <property name="config__database_user_password"> <value>"{salt}g7fVAbmt8rHELa1XZ5X0xbHUMCSlN2f/"</value> </property>
	DatabaseAuthenticator property: config__j_d_b_c_connection_u_r_l	Change the JDBC connection string to the target database. Example: <property name="config__j_d_b_c_connection_u_r_l"> <value>"jdbc:oracle:thin:@acmedb.acme.com:1521:orcl"</value> </property>

Table 12-1 Changes to Administration Server Files

File	Property	Description
ales32_admin\asiDomain\config\config.xml	<sec:authentication-provider xmlns:ext="http://www.bea.com/ns/weblogic/90/security/extension" xsi:type="ext:database-authenticatorType">	If the Administrator Server container is WebLogic Server 9.0 or later, modify the jdbc connection string, database user name, and password. To achieve password encryption, enter the password in clear text and restart WebLogic server. The restart should encrypt the password. **Note:** Not necessary to change the username and password if the target database uses the same values (default = *admin*, *password*). Examples of the pertinent definitions are shown below: <ext:jdbc-connection-url>jdbc:oracle:thin:@acmedb.acme.com:1521:orcl</ext:jdbc-connection-url> <ext:database-user-login>admin</ext:database-user-login> <ext:database-user-password-encrypted>{3DES}u6iQ9mLdY1xy6mLiSJrXyg== </ext:database-user-password-encrypted>

3. If necessary, use the asipassword command to add the OES database user on the target database to `password.xml`.

   This is not necessary if the target database uses the same username and password for the OES database user (default = *admin*, *password*).

   For information about asipassword, see the *Administration Reference* guide.

4. If the administration server is running on Tomcat or Websphere, run the `propagateInitialCache.bat` batch processing file found under the **`ales32-admin\bin`** directory.

5. Start the SCM and Administration Server.

6. Log into the Administration Console and modify the *asiadmin* SSM configuration's Database Authenticator to point to the target database.