Oracle ® Entitlements Server 10*g* (10.1.4.3)

SSM Installation and Configuration Guide

September 2008

ORACLE

SSM Installation and Configuration Guide, Oracle ® Entitlements Server

Introduction

Overview

Installing SSMs

Configuring SSMs Using ConfigTool

Silent Mode Installations

CHAPTER 1

Introduction

This section describes the contents and organization of this guide.

Note: Oracle Entitlements Server was previously known as BEA Aqualogic Enterprise Security. Some items, such as schema objects, paths, and so on may still use the term “ALES."

Document Scope and Audience

This document is a resource for system administrators, database administrators, and software developers who need to install and configure Oracle Entitlements Server Security Service Modules (SSMs). It contains information that is relevant during the design, staging, and production deployment phases of a software project.

Guide to this Document

This document is organized as follows:

- “Overview” on page 2-1 provides a summary of installation and configuration tasks.
- “Installing SSMs” on page 3-1 describes installation requirements, pre-installation tasks, and gives step-by-step instructions for installing SSMs.

- "Configuring SSMs Using ConfigTool" on page 4-1 describes how to configure the WLS, WLS 8.1, Java, and Web Service SSMs using the ConfigTool.
- "Configuring the Web Server SSM" on page 5-1 describes how to configure the Web Server SSM.
- "Configuring the Oracle SSM" on page 5-1 describes how to configure the Oracle SSM.
- "Configuring the WebSphere SSM" on page 6-1 describes how to configure the WebSphere SSM.
- "Configuring a Remote SSM and Proxy" on page 7-1 describes how to use a remote SSM by implementing an SSM proxy on an application client.
- "Configuring a Custom SSM" on page 8-1 describes how to create and configure a custom SSM.
- "Running an SSM Without an SCM" on page 9-1 provides criteria for deciding whether to provision the SSM with configuration updates using an SCM or an export file. It also gives pertinent instructions.
- "Silent Mode Installations" on page 10-1 describes how to perform silent-mode SSM installs. This can be useful when installing multiple SSMs on many machines.

Related Documentation

For information about installing and configuring the Administration Server, see *Installing the Administration Server*. For information about other aspects of Oracle Entitlements Server, see the following documents:

- *Introduction to Oracle Entitlements Server* provides overview, conceptual, and architectural information of this product.
- *Getting Started with Oracle Entitlements Server* provides a number of tutorials that show how to use the Entitlements Administration Application to secure application resources.
- *Securing OES Production Environments*—Contains information about security practices that should be considered when moving OES from a development to a production system.
- *Policy Managers Guide d*efines the policy model and describes how to manage, generate, import, and export policy data.

- *Programming Security for Java Applications* describes how to implement security in Java applications. It includes descriptions of the security service APIs and provides programming instructions.
- *Developing Security Providers* provides security vendors, administrators, and application developers with information needed to develop custom security providers.
- For API documentation in Javadoc format, see the documentation home page.

CHAPTER 2

Overview

Security Service Modules (SSMs) are installed on the machines hosting the applications to be secured. An SSM ties the secured application into Oracle Entitlements Server so that all administrative security activities are performed through the Administration Server.

The following out-of-box SSMs are available in this release:

- WLS SSM (for WebLogic 9.x/10.x)
- WLS 8.1 SSM (for WebLogic 8.1)
- Web Server SSM (for Microsoft IIS and Apache Web Server)
- Web Service SSM
- WebSphere SSM
- Oracle SSM

Installation Overview

The primary tasks for installing one or more SSMs is to run the SSM installation program and then perform the enrollment process which sets up secure communication with the Administration Server. The same SSM installer is used to install all out-of-box SSMs.

For instructions, see “Installing SSMs” on page 3-1.

Configuration Overview

After installing the SSM and performing the enrollment process, the SSM instance must be created and its initial configuration defined.

Tip: The term ‘configuration’ is being used broadly here to include initial policies and policy components (resources, identities, etc.) in addition to the SCM, SSM, and security providers.

There are a number of ways by which SSM instances are created and configured:

- For the WLS, WLS 8.1, Web Service, and Java SSMs, a utility called the ConfigTool can be used. This tool automates many tasks that must otherwise be performed manually. For more information about the tool and how to use it, see “Configuring SSMs Using ConfigTool” on page 4-1.
- The Websphere, Web Server, Oracle, and custom SSMs involve unique tasks that are described in chapters 5 through 8.
- All SSMs can be configured by manually defining the SSM’s configuration and the policies to enforce when securing an application. Detailed instructions are provided in a number of documents, particularly the Policy Manager’s Guide and the help systems for the Administration Console.

CHAPTER 3

Installing SSMs

This sections provides step-by-step instructions for installing and enrolling SSMs and performing additional post-installation tasks.

The following topics are described:

Installation Requirements

SSMs require certain software components to operate properly.

System Requirements

For system requirements on machines where which SSMs are installed, see the release notes.

Other Requirements

- The machine on which a SSM is installed must have a static IP address.
- On Windows, the file system must be NTFS (not FAT). To check the file system format, open Windows Explorer and right-click the hard drive on which the SSM is being installed and select **Properties**.
- Do *not* install the SSMs from a network drive. Download the installation file to a local drive on the machine and install it from there.

Installation Topologies

SSMs can be installed using a distributed or centralized deployment model:

- Distributed — SSMs are installed on the same machine as the secured application. The SSM obtains configuration information and policy updates directly from the Administration Server. This is the prevasive method of deploying SSMs.

  For this model, install the SSM on the application machine as described in "Installation Steps" on page 3-3.

- Centralized — An SSM proxy on the application client communicates with a centralized SSM on a separate machine. The proxy emulates an SSM security services on the application client and provides authorization caching, logging, and failover support.

  When using an SSM proxy, RMI or SOAP can be used for communication with the remote SSM. Instructions for setting up SSM proxies are located in "Configuring a Remote SSM and Proxy" on page 7-1.

SSM Upgrades

The SSM installer detects earlier versions and upgrades an existing SSMs using its configuration information. To perform an upgrade, follow this procedure:

1. Upgrade the Administration Server before upgrading any SSMs. SSMs can continue to run while the Administration Server is being upgraded.
2. Make sure you have read and delete permission for the SSMs files. You must be logged in as a member of the group used when the earlier version was installed.
3. If using an SCM on the SSM machine, shut it down.
4. Run the installation program, as described in "Installation Steps" on page 3-3.

When the SSM installer runs, it detects the earlier versions and uses its configuration information.

5. Respond to the prompts as required.

Installation Steps

To install an SSM:

1. Shut down any running programs.
2. Unzip the installation ZIP file.

   The file name is `OES10gR3_ssm_win32.zip` (Windows), `OES10gR3_ssm_solaris32.zip` (UNIX), or `OES10gR3_ssm_linux.zip` (Linux).
3. Launch the installation program as described in Table 3-1.

Table 3-1 SSM Installation Programs

Windows	Launch `OES10gR3_ssm_win.exe`
	Note: To generate a verbose installation log, add the following to the launch command:
	`-log=<logfile> -log_priority=debug`
	Example:
	`OES10gR3_ssm_win32.exe -log=D:\logs\oes_install.log -log_priority=debug`

Table 3-1 SSM Installation Programs

Solaris	1. Change the protection on the install file by entering: `chmod u+x OES10gR3_ssm_solaris32.bin`.
	2. Enter: `OES10gR3_ssm_solaris32.bin`
	Note: To generate a verbose installation log, add the following to the launch command:
	`-log=<logfile> -log_priority=debug`
	Example:
	`oes320ssm_solaris32.bin -log=/opt/logs/oes_install.log -log_priority=debug`

Linux	1. Change the protection on the install file by entering chmod u+x `OES10gR3_ssm_rhas_IA32.bin`. 2. Enter: `OES10gR3_ssm_rhas_IA32.bin` **Note**: To generate a verbose installation log, use same command string as described above for Solaris.

4. Complete the prompts using Table 3-2.

Table 3-2 SSM Installation Prompts

Window	Action
Welcome	Click **Next**.
Choose Home Directory	Accept the default location (recommended) or select a different one and click **Next**.
Choose Products and Components	Select the SSMs to install and click **Next**. Only installable components are listed. For example, if installing on WebLogic Server 9.2/10.0, the SSM for WebLogic 8.1 is not listed.
Choose Product Installation Directories	Accept the default or specify a different directory and click **Next**. If the directory you specify does not exist, the installation program will create it. If you have installed other SSMs, you will see **Installation Complete**. Otherwise, continue.

Table 3-2 SSM Installation Prompts

Centralized Configuration of Security Providers	Accept the default checkbox selection to use an SCM for distributing configuration data to the SSM or clear the checkbox to not use an SCM. For more information about this, see "Running an SSM Without an SCM" on page 9-1 for more information. **Note:** This window does not appear when installing only the WLS SSM.
Choose Network Interface	Select the IP address the SCM will use to listen for requests to provision configuration data and click **Next**.
Configure SCM	**SCM Logical Name** — (Applicable only if using an SCM) Enter a name to assign the SCM. This name must be used later as described in **"Define an SCM in the Database" on page 3-8**. **SCM Port** — (Applicable only if using an SCM) Accept the default or specify a different port used by the SCM to receive data from the Administration Server. The port cannot be used by any other server. **Primary Server URL** — Enter the Administration Server address in the format: `https://servername:7010`. **Backup Server URL** — Leave blank unless you have a second Administration Server installed, in which case enter its address using the same URL format.
Choose JDK	Accept the default selection or specify a different JDK and click **Next**.

5. On the **Installation Complete** window, click **Close**.

Enrollment

Note: This section does not apply to the Web Server SSM, which uses a different enrollment tool, as described in "Configuring the Web Server SSM" on page 5-1.

Enrollment is the process by which an OES component on a remote machine registers with the Administration Server. As part of this process, the SSM system exchanges security certificates with the Administration Server.

All components located under a `BEA_HOME` directory use the same set of keys located in `BEA_HOME/ales32-shared/keys`. Therefore, the enrollment process must be run once for any given `BEA_HOME`.

There are two enrollment modes:

- In *demo* mode, enrollment clients use *BEA_HOME*\ales32-shared\keys\DemoTrust.jks to verify the Administration Server's certificate from webserver.jks.

  When the client tries to enroll, the Administration Server presents its public certificate for verification to the client. This public certificate is signed by a trusted ALES Demo CA and bound to the server's hostname.

  The client will trust the certificate, because the DemoTrust.jks keystore has the same public certificate of the same trusted Demo CA that is in webserver.jks.

- In *secure* mode, the enrollment client uses the cacerts certificates file from the JDK installation to verify the Administration Server's certificate from webserver.jks.

  cacerts is a system-wide keystore that conatins CA certificates. For example, the file for the jrockit_150_11 JDK is in *BEA_HOME*\jrockit_150_11\jre\lib\security\cacerts

Certificates

Some certificates issued by CA authorities do not strictly comply with Certicom's Internet X.509 Public Key Infrastructure standard. To use these certificates, you must disable constraints extension checking by adding the following lines to enroll.bat|sh and unenroll.bat|sh located in the BEA_HOME/ales32-shared/bin directory.

```
if [ -f $JAVA_HOME/lib/security/cacerts ]; then

    JAVA_OPTIONS="-Dbea.home=$BEA_HOME
-Dwles.ssl.enforceConstraints=false -Dwles.ssl.verifyHostnames=yes
-Dwles.ssl.trustedCAKeyStore=$JAVA_HOME/lib/security/cacerts
-Dlog4j.configuration=file:./log4j.properties"

         else

    JAVA_OPTIONS="-Dbea.home=$BEA_HOME
-Dwles.ssl.enforceConstraints=false  -Dwles.ssl.verifyHostnames=yes
-Dwles.ssl.trustedCAKeyStore=$JAVA_HOME/jre/lib/security/cacerts
-Dlog4j.configuration=file:./log4j.properties"

         fileif [ "$1" = "demo" ]; then

    JAVA_OPTIONS="-Dbea.home=$BEA_HOME
-Dwles.ssl.enforceConstraints=false  -Dwles.ssl.verifyHostnames=no
-Dwles.ssl.trustedCAKeyStore=$ALES_SHARED_HOME/keys/DemoTrust.jks
-Dlog4j.configuration=file:./log4j.properties"

else
```

Enrollment Steps

To run the enroll tool, perform the following steps:

1. Make sure the Administration Server is running and configured for 1-way SSL. For further details, see *Securing OES Production Environments*.
2. If the SSM is using an SCM, make sure the SCM is running.
3. In the `BEA_HOME/ales32-shared/bin` directory, set the environment:

   ```
   set-env
   ```

4. Run the following script:

   ```
   enroll demo
   ```

5. When the Enrollment prompt appears, enter the Administration Server administrator username and password. (The defaults are `admin` and `password` respectively).
6. Enter and confirm the following passwords. You choose the passwords; they do not need to match the key passwords used when the Administration Server was installed.

   Private key password — Protects the identity of the components being enrolled.
   identity.jceks password — Protects the `identity.jceks` keystore.
   peer.jks password — Protects the `peer.jks` keystore.
   trust.jks password — Protects the `trust.jks` keystore.

For more information on `enroll` utility options, see Administrative Utilities in the *Administration Reference*.

Example of Running Enroll

```
D:\bea\ales32-shared\bin>set-env
D:\bea\ales32-shared\bin>enroll secure
================================================================================
Enrollment/Unenrollment Utility
================================================================================
Enter admin username :> admin
Enter admin password :>
Enter SSM private key password :>
Confirm SSM private key password :>
Enter password for identity.jceks :>
Confirm password for identity.jceks :>
Enter password for peer.jks :>
```

```
Confirm password for peer.jks :>
Enter password for trust.jks :>
Confirm password for trust.jks :>

Submitting enrollment request
Processing enrollment response
Updating trusted CA keystore
Updating peer keystore
```

Define an SCM in the Database

Use the Administration Console to define an SCM. When the ConfigTool sets up the initial security providers that will be used by the SSM to secure the application, this information will be maintained under this SCM.

Note: For step-by-step instructions on creating an SCM, see "Configuring a Service Control Manager" in the Administration Console's help system.

If the SSM will run using an SCM, the name of the SCM must match the **SCM Logical Name** entered when the SSM was installed. For details, see Table 3-2, "SSM Installation Prompts," on page 3-4.

You must define the SCM even if the SSM does not use it to obtain configuration data from the Administration Server. When this is the case, SCM will be the collection point for exporting configuration data to an XML file. For more information, see "Running an SSM Without an SCM" on page 9-1.

Run asipassword

Before configuring the SSM, use the asipassword utility to set the Administration Server's `admin` user password on the SSM machine. This password is required for secure communications between the SSM and the Administration Server.

To run the tool:

1. Change to the `BEA_HOME\ales32-shared\bin` directory and enter the following:

   ```
   asipassword admin <BEA_HOME>\ales32-shared\keys\password.xml
   <BEA_HOME>\ales32-shared\keys\password.key
   ```

   Example:

```
asipassword admin c:\bea\ales32-shared\keys\password.xml
c:\bea\ales32-shared\keys\password.key
```

2. When prompted for the 'alias' password, enter the Administration Server user's password. (The default password is *password*.)

Notes:

- The `password.xml` file does not exist until the enrollment process completes.

What's Next?

After installation, create and configure SSM instances as described in the following chapters:

- For the WLS SSM, WLS 8.1 SSM, Java SSM, and Web Service SSM, see "Configuring SSMs Using ConfigTool" on page 4-1.
- For the Web Server SSM, see "Configuring the Web Server SSM" on page 5-1.
- For the Oracle SSM, see "Configuring the Oracle SSM" on page 5-1.
- For the Websphere SSM, see "Configuring the WebSphere SSM" on page 6-1.
- For a custom SSM, see "Configuring a Custom SSM" on page 8-1.

CHAPTER 4

Configuring SSMs Using ConfigTool

This section describes how to configure the WLS, WLS 8.1, Java, and Web Service SSMs using the ConfigTool.

Prerequisites

- For the WLS SSM and WLS 8.1 SSM, create a Weblogic domain for the application that will be secured using the SSM. This is not required if the application is already using a domain.
- For WebLogic 9.x, 10.x running with a Sun JDK, it is recommended that you have at least 256MB of "PermGen" space. You can provide this by adding the following to JAVA_OPTION in `<domain-home>/bin/startWebLogic.sh|bat`:

  ```
  -XX:PermSize=128m -XX:MaxPermSize=256m
  ```

ConfigTool Overview

For the WLS, WLS 8.1, Web Service, and Java SSMs, this release provides a utility called the ConfigTool that automates a number of steps that must otherwise be performed manually. In

particular, the ConfigTool defines the SSM's initial configuration as well as a set of basic policies that can be added to or modified as required to secure the application.

Note: Since the WLS SSM uses WebLogic security providers, the ConfigTool adds these to the WebLogic server. They must be managed using the WebLogic console.

It is recommended that you generate an initial configuration with the ConfigTool and then use the Administration Console and Entitlements Management Tool to update or modify the policies as needed to secure the application.

When the ConfigTool runs, the information added depends on template files provided when the SSM is installed. These files are located in the SSM's `config` directory. For example, the template files used for configuring the Java SSM are located in `BEA_HOME\ales32-ssm\java-ssm\config\java-ssm\ales-policies`.

The data added by the ConfigTool depends on the type of SSM and is based on out-of-box policies that are provided when the SSM is installed. Table 4-1 provides a general description of the type of information added.

Table 4-1 Information Added by ConfigTool

Database Entries	Description
Security Service Module	An SSM is created and used to contain the security providers that make decisions about user requests in the protected application.
Security Providers	Creates a number of security providers that the SSM uses to secure the application. For example, the ConfigTool adds the following providers for a Web Service SSM: ASI Adjudicator Log4j Auditor ALES Identity Asserter Database Authenticator ASI Authorization Provider ALES Identity Credential Mappers ASI Role Mapper Provider **Note:** For the WLS SSM, the ConfigTool adds providers to WebLogic where they can be managed using the WebLogic console.

Table 4-1 Information Added by ConfigTool

Database Entries	Description
Organization **Note:** This is provided in OES 10gR3 Cumulative Patch 2 (CP2).	An Organization is created under RootOrg and named as specified in `myssm_config.properties`.
Identity Directory	The Identity Directory is used to define and manage Users and Groups for the protected application. The name to use for creating the Identity Directory is specified in `myssm_config.properties` prior to running the ConfigTool.
Policies	A number of default authorization and role mapping policies are added. Those added depend on the SSM type.

Before running the ConfigTool, a properties file must be updated to include names and other information you want the tool to use when adding the initial configuration and policies.

The tool has *check* (validate) and *process* options. In check mode, the tool verifies that the SSM instance can be created without error. In process mode, the tool actually creates the SSM instance and configuration. It is recommended that you first run with the check option to make sure that there are no errors.

ConfigTool Steps

The ConfigTool performs a number of steps that are not observable during execution. This section provides a detailed description of ConfigTool operations. These operations are performed in three stages:

- Collects and Builds Configuration Data
- Performs Preconfiguration Checks
- Makes Configuration Changes

Collects and Builds Configuration Data

The following steps are performed:

1. Reads the configuration information specified in the properties file. Confirms any default values that were not specified and prompts for any required data.

2. Builds a properties object with all the information.
3. Copies the policy files from the SSM's `/config/<SSM_TYPE>/ales-policies` into a temporary directory.
4. Substitutes all "@...@" values in the temp directory with data in the properties object.

Performs Preconfiguration Checks

The following steps are performed. If any check is not verified, it aborts and exit.

1. If custom.ant.script is enabled, it verifies the existence of the script file.
2. Verifies that enrollment was performed.
3. Verifies that asipassword was run
4. Verifies that the SSM instance does not exist.
5. Verifies that the ARME port is free.
6. Check connectivity to BLM server process on the Admin Server.
7. Check JDBC parameters by connecting to the database.
8. For all WebLogic domains, it verifies that the domain directory exists and that there are no ConfigTool backup files in the domain directory (this prevents affecting a domain is already secured).
9. For WebLogic 9.2 and later, it verifies the config.xml and that the domain is not running and then starts it. Then it verifies that WLST script can connect and login. Then it shuts down the domain

Makes Configuration Changes

The following steps are performed:

1. Uses the SSM's instance wizard (instancewizard.sh|bat) to create the SSM instance.
2. Uses policy loader and policyIX to load policies from temporary directory.
3. Uses the SetPassword tool to set the password for the Admin Server `admin` user.
4. For WebLogic domains, edits the StartWeblogic script in the domain, inserts OES JAR files to the CLASSPATH, and adds "JAVA_OPTIONS". It also copies the `security.properties` file.
5. For WebLogic 9.2 and later, it starts and verifies the WebLogic domain, creates a new security realm, creates and configures all required providers. It then switches the default realm to the new realm and shuts down the domain.

Configuration Steps

1. If using an SCM on the machine, make sure it is running.

2. Make a backup copy of `myssm_config.properties` located in the SSM's `adm` directory. Then open the file in a text editor and make the changes shown in Table 4-2.

Table 4-2 Properties File Modifications

Field	Description
wls.domain.dir	(WLS, WLS 8.1 SSM Only) The path to the WebLogic domain directory. **Note:** Use forward slashes. Example: `wls.domain.dir = BEA-HOME/user_projects/domains/App1_domain`
ssm.conf.id	A unique name for the SSM. Example: `ssm.conf.id = MyAppName`
db.password	The OES database user password. The name of the OES database user can be obtained by viewing `database.properties` in the Administration Server's `config` directory. The ConfigTool will prompt for this value if it is not specified in the properties file. For security purposes, it is recommended that you not store clear-text passwords in the properties file. Example: `db.password = <password>`
ales.admin.password	The OES administrator's password. The OES administrator's default user name and password is `admin` and `password` respectively. The ConfigTool will prompt for this value if it is not specified in the properties file. For security purposes, it is recommended that you not store clear-text passwords in the properties file. Example: `ales.admin.password = weblogic`
ssm.admin.name	The username required to boot the application or WebLogic domain secured by the SSM. For WebLogic domains, the default user name is `weblogic`. Example: `ssm.admin.name = weblogic`
ssm.admin.password	The password for the username above. For WebLogic domains, the default password is `weblogic`. Example: `ssm.admin.name = weblogic` The ConfigTool will prompt for this value if it is not specified in the properties file. For security purposes, it is recommended that you not store clear-text passwords in the properties file.

Table 4-2 Properties File Modifications

Field	Description
ssm.type	Specify the SSM type. One of the following: `java-ssm` — Java SSM `webservice-ssm` — Web Service SSM `wls8-ssm` — WebLogic 8.x domain `wls-ssm` — WebLogic 9.x or 10.x domain `wls-portal-ssm` — Portal-based domain in WebLogic 9.x/10.x `wls-alsb-ssm` — Oracle Service Bus-based domain in WebLogic 9.x/10.x Example: `ssm.type = wls-portal-ssm`
db.login	(REQUIRED ONLY IF THE ADMINISTRATION SERVER IS ON A SEPARATE MACHINE) The OES database user name. The user name user can be obtained by viewing `database.properties` in the Administration Server's `config` directory. Example: `db.login = alfred`
ales.admin.name	(REQUIRED ONLY IF THE ADMINISTRATION SERVER IS ON A SEPARATE MACHINE) The OES administrator's username. The ConfigTool will prompt for this value if it is not specified in the properties file. The administrator's default user name and password is `admin` and `password` respectively. Example: `ales.admin.name = admin`
ssm.instance.name	The name that will be assigned to the SSM instance. Example: `ssm.instance.name` = `MySsm`
ales.organization.scope **Note:** This is provided in OES 10gR3 Cumulative Patch 2 (CP2).	The name to be used for creating the Organization. Example: `ales.resource.root = MyOrg`
ales.identity.dir	A name that will be used to create the Identity directory containing the application's users and groups. Example: `ales.identity.dir = MyDir`

Table 4-2 Properties File Modifications

Field	Description
Database JDBC URL	(REQUIRED WHEN THE ADMINISTRATION SERVER IS ON A SEPARATE MACHINE) The JDBC connection string to the OES database. This varies by database type: Oracle — `jdbc:oracle:thin:@<server>:<port>:<sid>` Sybase — `jdbc:sybase:Tds:<server>:<port>` Sql Server — `jdbc:sqlserver://<server>:<port>` Pointbase — `jdbc:pointbase:server://<server>/ales` where: `<server>` — name or IP address of database machine `<port>` — port where the database listener is running `<sid>` — SID for oracle database Example for Oracle: `db.jdbc.url = jdbc:oracle:thin:@db_server:1521:db_sid`
Database JDBC Driver	(REQUIRED WHEN ADMINISTRATION SERVER IS ON A SEPARATE MACHINE) The database JDBC driver type. One of the following: Oracle — `oracle.jdbc.driver.OracleDriver` Sybase — `com.sybase.jdbc3.jdbc.SybDriver` Sql — `com.microsoft.sqlserver.jdbc.SQLServerDriver` Pointbase — `com.pointbase.jdbc.jdbcUniversalDriver` DB2 — `com.ibm.db2.jcc.DB2Driver`
arme.port	The ARME's port number that was specified when the SSM was installed, by default this is 8000. Example: `arme.port = 8000`
custom.ant.script	(Advanced Users Only) If desired, specify an Ant script that will be executed after the configuration is complete. Such a script could be used to add additional configuration information. Example: `custom.ant.script = /<dir_name>/CustomAntScript.xml`

3. Run `ConfigTool.bat -check myssm_config.properties` to ensure there are no errors.
4. Run `ConfigTool.bat -process myssm_config.properties`.

Add JDBC Driver to the Classpath

This section describes how to specify the location of the JDBC driver in the CLASSPATH environment variable. This is required if you are using a MS SQL, PointBase, or DB2 database and the WLS, WLS 8.1, Java, or Web Service SSM.

- Web Service SSM
- Java SSM
- WLS and WLS 8.1 SSMs

Notes:

- Due to license restrictions, the JDBC driver for MS SQL, PointBase, and DB2 are not included. They are available for download from the vendor websites.
- You must use the latest MS SQL 2005 JDBC driver with **all** versions of MS SQL.

Web Service SSM

To add the JDBC driver to the CLASSPATH, edit *INSTANCE_HOME*`/config/WLESws.wrapper.conf` and append the JDBC driver to the `wrapper.java.classpath` parameter.

Example:

```
wrapper.java.classpath.48=<BEA_HOME>/ales32-ssm/webservice-ssm/lib/sslclient.jar
wrapper.java.classpath.49=<BEA_HOME>/ales32-ssm/webservice-ssm/lib/pdsoap11.jar
wrapper.java.classpath.50=<BEA_HOME>/ales32-ssm/webservice-ssm/lib/antlr.jar
wrapper.java.classpath.51=../pbclient51.jar
```

Java SSM

To add the JDBC driver to the CLASSPATH, edit *INSTANCE_HOME*`/bin/set-env.bat` (or `set-env.sh`) and append the JDBC driver to the CLASSPATH environment variable.

Example:

```
set CLASSPATH=%CLASSPATH%;%INSTALL_HOME%\lib\antlr.jar
set CLASSPATH=%CLASSPATH%;%INSTALL_HOME%\lib\jaxrpc.jar
set CLASSPATH=%CLASSPATH%;f:\pbclient51.jar
```

WLS and WLS 8.1 SSMs

To add the JDBC driver to the CLASSPATH, edit the *INSTANCE_HOME*/bin/set-wls-env.bat (or set-wls-env.sh)file and append the JDBC driver location to the WLES_POST_CLASSPATH environment variable.

Example:

```
set
WLES_POST_CLASSPATH=%WLES_POST_CLASSPATH%;%INSTALL_HOME%\lib\jsafeJCE.jar
set WLES_POST_CLASSPATH=%WLES_POST_CLASSPATH%;%INSTALL_HOME%\lib\asn1.jar
set WLES_POST_CLASSPATH=%WLES_POST_CLASSPATH%;%INSTALL_HOME%\lib\certj.jar
set WLES_POST_CLASSPATH=%WLES_POST_CLASSPATH%;f:\pbclient51.jar
```

CHAPTER 5

Configuring the Oracle SSM

The Oracle SSM makes use of a feature in Oracle 10g called **Fine Grained Access Control** (FGAC). This allows an Oracle customer to define access policies that restrict access to database tables for DML operations.

FGAC is used to intercept DML queries on protected tables and filter the result sets based on user entitlements stored in OES. The Web Service SSM client library is used to invoke authorization queries.

This section describes how to configure and run the Oracle SSM.

Prerequisites

- Oracle 10g Release 2 (10.2.0.1.0) is installed and configured.
- The Administration Server is installed on a separate system.
- Web Service SSM is installed (but not running) on the same system where the Oracle SSM will run, as described in "Configuring SSMs Using ConfigTool" on page 4-1.
- On Linux/UNIX, the Oracle SSM and WebServices SSM must be installed by the main oracle user (by default this is 'oracle').

- The currently logged on user must belong to the `ora_dba` group on Windows or `dba` group on UNIX. This is required in order to connect as "system" user with "SYSDBA" role.
- The `ORACLE_HOME/ordata` directory must be writable. Files are added there by OES scripts.

Create and Configure the Oracle SSM

1. Make sure the Administration Server is running.
2. If the enrollment process has not been performed for the BEA_HOME containing the SSM, then:
   a. Run the enroll tool, as described in "Enrollment" on page 3-5. You can use demo mode.
   b. Include the password for system in the encrypted password.xml by running the following in the `ales32-shared/bin` directory:

      ```
      asipassword.bat|sh system ../keys/password.xml ../keys/password.key
      ```
3. Make sure the Web Service SSM is **not** running. Then launch `BEA_HOME/ales32_SSM/oracle-ssm/adm/instancewizard.cmd|sh` and complete the wizard fields as indicated in Table 5-1. These entries reflect the values specified in `BEA_HOME/ales32-ssm/oracle-ssm/examples/OracleSSM/build properties`.

Table 5-1 Oracle SSM Instance Wizard

Field	Description
Instance Name	oraclessm
SM WS Port	9000
SM WS Config ID	ora_ws_ssm
Location	Accept the default value.

4. In the new Oracle SSM instance's `bin` directory, use an editor to make the following changes in `set-env.bat|sh`:
   a. If required, update JAVA_HOME.
   b. Make sure that SQLPlus is in the system path using a line such as the following:

      ```
      set PATH=C:\oracle\product\10.2.0\db1\bin;%PATH%
      ```

5. In a command shell, set the following environment variables:

   ORACLE_HOME
   ORACLE_HOME/bin (must be the first element in path)
   ORACLE_SID (set to the SID of the database)

6. Execute `setupOracleSSM.bat|sh` in the shell window. Substitute the actual values for each field.

   ```
   setupOracleSSM.bat|sh
   -jdbc_url <JDBC_URL>
   -oracle_home <c:/oracle/products/10.2.0/db2>
   -db_sys_user <system>
   -db_sys_password <password>
   -ales_ssm_home <c:/bea/ales32-ssm>
   -ales_shared_home <c:/bea/ales32-shared>
   -ws_ssm_instance_dir <c:/bea/ales32-ssm/webservice-ssm/<instance>
   -db_user <ales_ora_user>
   -db_password <password>
   -load_example_table <true>
   ```

   Note: The `-db_user` value must not be the name of an existing user. This user is created when the script is run.

7. Open a shell window and change the directory to `ales32-ssm/oracle-ssm/examples/OracleSSM`.

8. Update `build.properties` and then execute `set-env.bat|sh`.

9. Run `ant dist config load`.

10. In Administration Console, perform the following steps:

    a. Go to the SSM Configuration of the Web Service SSM and click on **Authentication**> **FGACIdentityAsserter**.

    b. On the **Details** tab, enter the **Key** value of secret property defined in table `oes_oracle_ssm_properties`.

       Note: This value can be obtained from the database using the following SQL query:

       ```
       'select value from oes_oracle_ssm_properties where KEY='secret';')
       ```

    c. Navigate to the **Deployment** node and distribute the configuration.

11. Start the Web Service SSM instance.

12. To test the results, use SQLPlus to run the following queries:

a. Run query as sysdba to see the complete data set.

```
$ sqlplus sys/sys-password@oracle-listner as sysdba

SQL> select * from ales_ora_user.cust_payment_info;

FIRST    LAST       ORDER   CREDIT_CARD_NUMB
Jon       Oldfield   10001  5446959708812985
Chris     White      10002  5122358046082560
Alan      Squire     10003  5595968943757920
Mike      Anderson   10004  4929889576357400
Annie     Schmidt    10005  4556988708236902
Elliot    Meyer      10006  374366599711820
Celine    Smith      10007  4716898533036
Steve     Haslam     10008  340975900376858
Albert    Einstein    10009  310654305412389
```

b. Run query as ales_ora_user to see a subset of the data.

```
$ sqlplus ales_ora_user/password@oracle-listner

SQL> select * from cust_payment_info;

$ sqlplus ales_ora_user/password@oracle-listner

FIRST    LAST       ORDER   CREDIT_CARD_NUMB
Chris     White      10002  5122358046082560
```

Client Run-Result

This section shows sample results by using run.bat|sh (a sample JDBC client).

Note: Before using run.bat|sh, update BEA_HOME/ales32-ssm/oracle-ssm/examples/OracleSSM/client.properties to reflect your {jdbcUrl,schemaName,queryType,query} settings.

Listing 5-1 shows a sample test result for a queryType of select, update, and delete.

Listing 5-1 Sample Oracle Client Run Result

```
C:\buildTree\ales32-ssm\oracle-ssm\examples\OracleSSM>run
Properties loaded from file : ./Client.properties
Database URL : jdbc:oracle:thin:@192.168.200.10:1521:ORCL
User Name : smysore3
User Password : password
User (of database connection) : SMYSORE3
```

```
ClientIdentifier : smysore3
Query Type [select/update/delete] : select
Query : select * from cust_payment_info
Executing SELECT query...
Last Name, First Name : White,Chris

C:\buildTree\ales32-ssm\oracle-ssm\examples\OracleSSM>run
Properties loaded from file : ./Client.properties
Database URL : jdbc:oracle:thin:@192.168.200.10:1521:ORCL
User Name : smysore3
User Password : password
User (of database connection) : SMYSORE3
ClientIdentifier : smysore3
Query Type [select/update/delete] : update
Query : UPDATE cust_payment_info set first_name = 'Test' where
first_name='Alan'

Executing UPDATE query...
0 rows updated

C:\buildTree\ales32-ssm\oracle-ssm\examples\OracleSSM>run
Properties loaded from file : ./Client.properties
Database URL : jdbc:oracle:thin:@192.168.200.10:1521:ORCL
User Name : smysore3
User Password : password
User (of database connection) : SMYSORE3
ClientIdentifier : smysore3
Query Type [select/update/delete] : delete
Query : DELETE from cust_payment_info where first_name='Alan'
Executing DELETE query...
0 rows deleted
```

CHAPTER 6

Configuring the WebSphere SSM

This section describes how to configure and set up the WebSphere SSM. It also describes a simple application that demonstrates how to retrieve basic security services and use them for authentication and authorization.

Prerequisites

- Administration Server is installed and running.
- WebSphere 6.1 Application Server is installed, but not running.
- WebSphere SSM is installed on the WebSphere server machine.

WARNING: If the WLS or WLS 8.1 SSM is running on the same machine, the WebSphere SSM must be installed and run in a different BEA_HOME. During installation in the new BEA_HOME, be sure to enter different values for the SCM name.

Configuration Steps

1. After the WebSphere SSM is installed, make sure the following steps have been completed:

- “Enrollment” on page 3-5
- “Define an SCM in the Database” on page 3-8
- “Run asipassword” on page 3-8

2. Create the WebSphere SSM instance by running `BEA_HOME\ales32-ssm\websphere-ssm\adm\instancewizard.cmd`.

   In Windows, this can be done by opening the **Start** menu and selecting **Oracle Entitlements Server > Security Service Module > Websphere Security Service Module > Create New Instance.**

3. If you are using a MS SQL, PointBase, or DB2 database, the location of the JDBC driver must be specified by opening the instance's `set-env.bat|sh` in an editor and appending the JDBC driver to the CLASSPATH environment variable.

   Example:

   ```
   set CLASSPATH=%CLASSPATH%;%INSTALL_HOME%\lib\antlr.jar
   set CLASSPATH=%CLASSPATH%;%INSTALL_HOME%\lib\jaxrpc.jar
   set CLASSPATH=%CLASSPATH%;f:\pbclient51.jar
   ```

4. Set up the `JavaAPIExample` in the `BEA_HOME/ales32-ssm/websphere-ssm/examples` directory.

5. Start the SCM and run the JavaAPIExample using the defaults.

6. Copy the contents of `BEA_HOME\ales32-ssm\websphere-ssm\<instance>\config\websphere-server.xml` into `BEA_HOME/websphere-ssm/AppServer/profiles/AppSrv01/config/cells/terminatorNode01Cell/nodes/terminatorNode01/servers/server1/server.xml`.

   Note: Make sure the xml blocks are inserted in the correct section.

7. Start the WebSphere Server by running the following script:

   BEA_HOME`/websphere-ssm/AppServer/profiles/AppSrv01/bin/startServer.bat|sh`.

8. Set JAVA_HOME in `BEA_HOME\ales32-ssm\websphere-ssm\examples\PolicyQueryWebApp\set-env.bat|sh`.

9. Run `BEA_HOME\ales32-ssm\websphere-ssm\examples\PolicyQueryWebApp\set-env.bat|sh`.

10. Run `ant all` to build the example.

11. Log in to the WebSphere Server console and deploy `/dist/PolicyQueryApp.war`.

12. Open a browser window and go to the deployed PolicyQueryApp application. For example:

```
http://<myhost>:9080/PolicyQueryApp/index.jsp
```

13. When you accept all the defaults and click on **Submit**, you should get the following on the `access.jsp` page:

```
Your Inputs
  user: system
  privilege: buy
  resource: store/book
  attributes: canbuy=yes;attrname=value
Evaluation Results
  Allowed.
Response Attributes
  No response attribute is returned!
```

CHAPTER 7

Configuring a Remote SSM and Proxy

This chapter describes the necessary steps for securing an application using a remote SSM.

Overview

When OES security calls using the Java API have been implemented in a custom application, SSM proxy services can be set up on the application host and used to communicate with a remote SSM on another machine. The SSM proxy provides local security services, including caching, logging, and failover support.

RMI or SOAP can be used as the transport mechanism for communication between the SSM proxy and the remote SSM. (In XACML terminology, the *SSM proxy* and *remote SSM* are be analogous to *PDP Proxy* and *PDP* respectively.)

Creating the Remote SSM

To set up a remote SSM:

1. On the remote machine, install and enroll the Web Services SSM as described in "Installing SSMs" on page 3-1.

   Note: For this release, there is no separate install for the RMI SSM. It is available only by installing and instantiating a Web Services SSM.

2. Create a Web Service SSM or RMI SSM instance.

   To create a Web Service instance, see "Configuring SSMs Using ConfigTool" on page 4-1.

   To create an RMI instance:

   a. Run `BEA_HOME\ales32-ssm\webservice-ssm\adm\create_rmi_ssm.bat | .sh`. This creates the `BEA_HOME\ales32-ssm\rmi-ssm` directory.

   b. `In BEA_HOME\ales32-ssm\rmi-ssm\adm` launch `instancewizard.cmd` and complete the wizard prompts.

Configuring the SSM Proxy

To set up the SSM proxy on the client application:

1. On the remote SSM, copy the directory indicated below and its files to a directory on the application client.

   Web Service SSM— `BEA_HOME\ales32-ssm\webservice-ssm\<instance>\pdpproxy`
   RMI SSM—
   `BEA_HOME\ales32-ssm\webservice-ssm\<instance>\rmi-ssm\pdpproxy`

2. Also on the remote SSM, copy the trust keystore located in `BEA_HOME\ales32-shared\keys` to the same directory on the application client containing the Jar files.

   Note: The demonstration trust keystore provided during installation is named `DemoTrust.jks`. This can be used in development and testing environments.

3. Make sure the directory on the application client containing the Jars and keystore file are included in the SSM client Classpath.

   To specify the SSM proxy location, use a system property in the command line, such as: `-Dpdp.configuiration.properties.location=D:\pdpproxy\PDPClientConfiguration.properties`.

4. Open `PDPProxyConfiguration.properties` in a text editor and set the values as described in Table 7-1.

Table 7-1 PDPProxyConfiguration.properties Settings

Setting	Description

ProxyID	An arbitrary SSM proxy ID. This name must be unique in the OES deployment.
	Examples:
	`ProxyId=SSM-RMI-Client1`
	`ProxyId=SSM-WS-Client1`

SSMConfigID	The configuration ID of the remote SSM. This configuration can be managed using the Administration Console. Example: `SSMConfigID=asiadmin`
PDPTransport	This setting determines whether RMI or SOAP is used as transport protocol for communicating with the remote SSM. Possible values: `RMI`, `WS`, or `JAVA` Example: `PDPTransport=RMI`

PDPAddress	A comma-separated list of the url of the primary remote SSM and port number and one or more failover SSMs.
	For RMI, the URL must begin with `rmi://`; for SOAP, it must begin with `http://`. To use SSL, these must be `rmis://` or `https://`.
	Example:
	`PDPAddress=rmi://acme01:9300,rmi://acme02:9300`
	`PDPAddress=https://acme01:9300,https://acme02:9300`

RequestTimeoutMilliSecs	Number of milliseconds to timeout a request is the remote SSM is not responding. Example: `RequestTimeoutMilliSecs=10000`
FailureRertyCount	Number of attempts to make before failing over to failover SSM. Example: `FailureRertyCount=3`
FailbackTimeoutMilliSecs	Number of milliseconds after failover to wait before attempting reaccess to the primary remote SSM. Example: `FailbackTimeoutMilliSecs=180000`

Table 7-1 PDPProxyConfiguration.properties Settings

Setting	Description
TrustStore	The fully-qualified path to the trust keystore copied in step 2 above. This provides one-way SSL connections with the remote SSM. Use forward slashes. **NOTE:** The PDPAddress parameter must begin with `rmis://` or `https://`. Example: `c:/bea3_2/ales32-shared/keys/<filename>.jks`
SynchronizationIntervalMilliSecs	The interval (in milliseconds) used by the proxy to poll the remote SSM to determine if the authorization cache should be synchronized. Example: `SynchronizationIntervalMilliSecs=60000`

5. The proxy files copied to the application client contains the Apache Log4j package. For logging SSM proxy activities, configure the log4j.properties with appenders to enable proxy debugging.

Configuring Caching, Logging, and Failover

How caching, logging, and failover are configured depends on the whether it is the remote SSM or the SSM proxy.

Caching

Caching configuration on the remote SSM is determined by settings established on the authorization provider being used by the SSM. This is managed using the Administration Console.

The SSM proxy uses the same caching configuration established on the remote SSM. In addition, the **SynchronizationIntervalMilliSecs** setting in `PDPProxyConfiguration.properties` determines how often to poll the remote SSM to see if re-synchronization is necessary.

The cache on the SSM proxy cannot be flushed from the Administration Console. However, the SSM proxy cache is automatically flushed when whenever there is a new policy distribution on the SSM (subject to the **SynchronizationIntervalMilliSecs** parameter). In addition, if this is not sufficient, the ability to flush the cache may be added to the application client using the following methods of the **AuthorizationService** class:

- **flushCache** — flushes the cache completely
- **flushCacheByUser** — flushes only data associated with a given identity (user).

These methods can be invoked from either the SSM or the SSM proxy. They will simultaneously flush both the client and server cache.

Logging

Logging configuration on the remote SSM is determined by settings established on the auditing provider being used by the SSM. This is managed using the Administration Console.

On the SSM proxy, logging can be implemented using the Apache Log4j package (log4j.jar) that is included in the `pdpproxy` directory that was copied from the SSM instance.

Failover

Failover support for an SSM proxy can be implemented by deploying one or more failover SSMs and then pointing to those SSMs using the **PDPAddress** setting in `PDPProxyConfiguration.properties`. The first specified SSM will be the primary SSM; the second will be the failover SSM.

Adding New Assertion Types (Web Services SSM)

To add support for new identity assertion types to the Web Services SSM:

1. Create a new Java class as a holder for the identity assertion. Note that the new holder class must belong to the `com.bea.security.ssmws.credentials` namespace. In this procedure, we use a class named `com.bea.security.ssmws.credentials.TestCredHolderImpl` and a custom identity assertion type named `TestIA`. See Figure 7-1 for an example.
2. Add the JAR file containing the holder class to the Web Service SSM's classpath. To do this, modify the `WLESws.wrapper.conf` configuration file located in `BEA_HOME/ales32-ssm/webservice-ssm/instance-name/config`. For example, if the holder class is contained in a file named `ssmwsCustomAssertion.jar`, add a line like this to `WLESws.wrapper.conf`:

   ```
   wrapper.java.classpath.40=C:/bea/ales32-ssm/webservice-ssm/lib/ssmwsCustomAssertion.jar
   ```

   Note: The `wrapper.java.classpath` lines must increment sequentially.

3. Modify the mapping file for incoming messages. Mapping for incoming messages is controlled by the `castor.xml` file in the `BEA_HOME/ales32-ssm/webservice-ssm/lib/com/bea/security/ssmws/soap` directory. Add an entry like the following inside the <mapping> XML element:

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

4. Modify the mapping file for outgoing messages. Mapping for incoming messages is controlled by the `castor.xml` file in the `BEA_HOME/ales32-ssm/webservice-ssm/lib/com/bea/security/ssmws/credentials` directory. Add an entry like the following inside the <mapping> XML element

```
<class name="com.bea.security.ssmws.credentials.TestCredHolderImpl">
<map-to cst:xml="TestIA"
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>
```

5. To log SOAP messages received and sent by the Web Services SSM, make the following changes to the SSM instance's `config/log4j.properties` file:

   a. Change `log4j.appender.A1.Threshold=ERROR` to `log4j.appender.A1.Threshold=DEBUG`

   b. Add the following entry: `log4j.logger.com.bea.security.ssmws.server=DEBUG`

When the Web Services SSM is started, it will use the new holder implementation and the mapping entries to convert back and forth between the token's XML and Java representations.

Figure 7-1 Sample Identity Assertion Holder Class

```
public class TestCredHolderImpl implements CredentialHolder
{
private String m_cookie;
public static final String m_Type = "TestIA";

public void setCookie(String cookie)
{
    m_cookie = cookie;
}
public String getCookie()
```

```
{
    return m_cookie;
}
public Object getObject()
{
    return getCookie();
}
public void setObject(Object cred)
{
    setCookie((String)cred);
}
public String getType()
{
    return TestCredentialHolderImpl.m_Type;
}
public String getAsString()
{
    return m_cookie;
}
}
```

CHAPTER 8

Configuring a Custom SSM

This section describes how to create and configure a custom SSM.

A custom SSM can be created by making a copy of an existing SSM and then making a few modifications before running the ConfigTool. This is a fairly straight-forward process, because all configuration information is stored in modifiable text files. Once the custom SSM is created, the same files can be used to replicate it across multiple systems.

The out-of-box SSMs that can be copied for this purpose are the following:

- WLS SSM
- WLS 8.1 SSM
- Java SSM
- Web Service SSM

Creating a Custom SSM

1. Determine the out-of-box SSM that most closely resembles what you need. Then select that SSM's directory and copy it to a directory to hold the custom SSM.

   For example, copy *BEA_HOME*\ales32-ssm\java-ssm to *BEA_HOME*\ales32-ssm\custom-ssm.

2. Rename the custom-ssm\config\java-ssm directory to custom-ssm\config\custom-ssm.

3. Change to the `BEA_HOME\ales32-ssm\custom-ssm\config\custom-ssm\ales-policies` directory and modify the default policies as follows:

   a. Add the following line to the `subject` file:

   ```
   //user/@ales.identity.dir@/@my.custom.user@/
   ```

   b. Add the following line to the `rule` file:

   ```
   grant( //role/Administrators, @ales.resource.root@,
   //user/@ales.identity.dir@/@my.custom.user@/) if true;
   ```

4. Change to the `BEA_HOME\ales32-ssm\custom-ssm\config\custom-ssm` directory and add the following line to `all-params.properties`:

   ```
   my.custom.user = string, Enter the name of the custom user
   ```

5. Make a backup copy of `BEA_HOME\ales32-ssm\custom-ssm\adm\myssm_config.properties`. Then make the following changes to `myssm_config.properties`.

   a. Set the `ssm.type` to `custom-ssm`.

   b. Define `my.custom.user` to a username.

   c. Modify other values as needed.

6. Run `ConfigTool.bat -check myssm_config.properties` to check the properties file.

7. Run `ConfigTool.bat -process myssm_config.properties`.

Replicating a Custom SSM

After creating a custom SSM as described above, perform the following steps to replicate it on another system:

1. Copy the `custom-ssm` directory from the source to the destination system.

2. If the SSM was installed to use an SCM, start the SCM.

3. If the enrollment process has not been performed for the BEA_HOME on the destination system, run the enrollment program as described in "Enrollment" on page 3-5.

4. If the custom SSM is based on the WLS or WLS 8.1 SSM, create a domain for the application to be secured.

5. If needed, update `myssm_config.properties` with the domain name, the correct path, and other variables.
6. Run the ConfigTool on the destination system.

CHAPTER 9

Running an SSM Without an SCM

This section provides information and instructions for running an SSM without an SCM.

Overview

An SCM is responsible for storing and maintaining the configuration data for all SSMs running on a machine. An SSM receives its configuration data from the SCM at startup and whenever a configuration change is made and distributed from the Administration Server. The SCM receives and caches the updated information, and provides it to the SSM when it is restarted.

Tip: The term 'configuration' is used in its restrictive sense here and refers only to the SCM, SSM, and the SSM's security providers. It does not refer to policy data.

An SSM can run without an SCM by obtaining its configuration information from data that is exported from the OES database using the PolicyIX tool. This tool allows you to export configuration data to an XML file that is read by the SSM when it is restarted.

Notes:

- The PolicyIX tool can extract both policy and configuration information from the database. In this context, it is used to extract configuration information only.
- Information in this section does not apply to the WLS SSM, which uses configuration information maintained in the WebLogic 9.x/10.x Administration Console. It does not use either an SCM or configuration data exported from the OES database.

Choosing How to Run the SSM

Use the following criteria when deciding whether to use an SCM or exported configuration data:

- A running SCM provides the ability to centrally manage all SSM configurations on a machine. This is extremely useful when running multiple SSMs and configuration changes are common.
- The SCM is an additional process that must be installed and maintained. This may be unneeded if configuration changes are relatively rare.
- When using an XML file, a manual export must be performed whenever a configuration change is made in the database. This may prove cumbersome, particularly when frequent configuration changes are made.
- Once an SSM is set up to obtain configuration data from an XML file, it cannot be switched to use an SCM. The SSM must be removed and then reinstalled.
- An SCM configuration must be maintained in the database whether or not an SCM is used on the SSM machine. The SCM configuration is the collection point for the SSM's configuration data that is exported from the database to the XML file.

Installing An SSM Without An SCM

During the SSM installation process, the **Centralized Configuration of Security Providers** window displays, as shown in Figure 9-1. When you clear the **Allow centralized configuration...** checkbox, the SSM will not use an SCM.

Figure 9-1 Disabling an SCM

Exporting Configuration Data

Perform the following steps to export an SSM's configuration data using the PolicyIX tool:

Note: Complete information about PolicyIX commands is provided in the PolicyIX section of the *Administration Reference*.

1. In the `BEA_HOME/ales32-admin/bin` directory, enter the following command:

   `policyIX.bat <exportID> -exportConfig policyIX_config.xml`

   where <*exportID*> is the name of the SSM configuration to export.

   Two files will be generated in the `bin` directory: `wles.securityrealm.xml` and `wles.securityrealm.xml.sig`.

2. Copy the two files to the SSM instance's `bin` directory.

   For example, for an WLS 8.1 SSM instance name of `WLS8Domain`, copy the files to `BEA_HOME/ales32-ssm/wls8/WLS8Domain/bin`.

3. Restart the SSM and ignore the instructions about starting the SCM.

4. Repeat these steps whenever the SSM's configuration is updated in the Administration Server.

Disabling an SCM

The following procedure illustrates how to disable the SCM for a specific SM.

1. Stop SCM.
2. Export the SSM configuration as documented in the previous section, Exporting Configuration Data.

   In the `BEA_HOME/ales32-admin/bin` directory, enter the following command:

```
policyIX.bat <exportID> -exportConfig policyIX_config.xml
```

   where <*exportID*> is the name of the SSM configuration to export. `wles.securityrealm.xml` and `wles.securityrealm.xml.sig` will be generated in the `bin` directory.
3. Copy the two files to the directory in which you start the SSM.
4. Restart the SSM.

   During SSM initialization, the authorization engine first attempts to retrieve configuration data from XMLConfiguration and, second, SCMConfiguration. XMLConfiguration will find the configuration files as copied under the directory in which you start the SSM.

Silent Mode Installations

This section describes how to install SSMs using silent mode installation.

Note: See **Silent Mode Installations** for instructions on how to silently install the Oracle Entitlements Server Administration Server.

Silent Mode Overview

In silent mode, the SSM installer reads inputs from an XML file rather than prompt you to enter these values one at a time. Silent-mode installs displays no GUI windows and completes without user intervention.

To perform a silent-mode install, you first set up the XML template file and then run the installer with an option that tells it to read that file.

- "XML File" on page 10-1
- "Launch the Installer in Silent-Mode" on page 10-3

XML File

When an SSM is installed in regular (non-silent) mode, a silent-mode XML file is created. This file can be modified and used for subsequent silent mode installs. The created file is *BEA_HOME*/ales32-ssm/<ssmtype>/adm/silent_install_ssm.xml. Use the information in Table 10-1 to modify the file for your purposes.

Table 10-1 Silent-Mode XML File Entries

Data Element Name	Description	Default or Sample Value
BEAHOME	BEA_HOME directory	C:\bea
COMBO_INSTALL_DIR	Directory within BEA_HOME to install the SSM	<BEA_HOME>\ales32-wls-ssm
SCM_INSTALL_DIR	Directory within BEA_HOME to install the SCM. **Note:** Do not enter if not using an SCM.	<BEA_HOME>\ales32-scm
COMPONENT_PATHS	Specifies the SMs to install, separated by the pipe (|) character. Possible component selections are: — OES_SM_COMBO/OES SM for Java — OES_SM_COMBO/OES SM for Web Service — OES_SM_COMBO/OES SM for IIS — OES_SM_COMBO/OES SM for Apache — OES_SM_COMBO/OES SM for WLS8.1 — OES_SM_COMBO/OES SM for WLS — OES_SM_COMBO/OES SM for Oracle — OES_SM_COMBO/OES SM for WebSphere	
SCM_JAVA_HOME	Java home for SCM.	<BEA_HOME>\jrockit150_06
WLS8_SSM_JAVA_HOME	Java home for the WLS 8.1 SSM.	<BEA_HOME>\jrockit150_06
WLS9_SSM_JAVA_HOME	Java home for the WLS SSM.	<BEA_HOME>\jrockit150_06
JAVA_SSM_JAVA_HOME	Java home for the Java SSM.	<BEA_HOME>\jrockit150_06
IIS_SSM_JAVA_HOME	Java home for the Web Server SSM on Microsoft IIS.	<BEA_HOME>\jrockit150_06
APACHE_SSM_JAVA_HOME	Java home for the Web Server SSM on Apache.	<BEA_HOME>\jrockit150_06
WEBSPHERE_SSM_JAVA_HOME	Java home for the Websphere SSM.	<BEA_HOME>\jrockit150_06

Data Element Name	Description	Default or Sample Value
ORACLE_SSM_JAVA_HOME	Java home for the Oracle SSM.	<BEA_HOME>\jrockit150_06
WEBSERVICE_SSM_JAVA_HOME	Java home for the Web Service SSM.	<BEA_HOME>\jrockit150_06
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
ENTERPRISE_DOMAIN_NAME	Should always be `asi`.	asi
SCM_NAME	The name to assign the Service Control Manager. **Note:** Do not enter if not using an SCM.	testscm
SCM_PORT	Port used by the SCM to receive configuration and policy data from the Administration Server; may not be used by any other server. **Note:** Do not enter if not using an SCM.	7005
SCM_PRIMARY_ADMIN_URL	The address used by the Administration Server.	https://lancer:7010/
SCM_BACKUP_ADMIN_URL	The address used by a secondary (backup) Administration Server, if you have one. Optional.	https://dancer:7010/

Launch the Installer in Silent-Mode

To run the SSM installation in silent mode, use one of the following commands:

- For Windows platforms:

  ```
  OES10gR3ssm_win32.exe -mode=silent -silent_xml=<path_to_silent.xml>
  ```

- For the Sun Solaris platform:

  ```
  OES10gR3ssm_solaris32.bin -mode=silent -silent_xml=<path_to_silent.xml>
  ```

- For the Red Hat Advanced Server Linux platforms:

  ```
  OES10gR3ssm_rhas_IA32.bin -mode=silent -silent_xml=<path_to_silent.xml>
  ```