BEA AquaLogic Enterprise Security™®

SSM Installation and Configuration Guide

Version: 3.0
Document Revised: December 2007

Introduction

Overview

Installing SSMs

Configuring SSMs Using ConfigTool

Configuring the Web Server SSM

Configuring the Oracle SSM

Configuring the WebSphere SSM

Configuring a Custom SSM

Running an SSM Without an SCM

Silent Mode Installations

Introduction

This section describes the contents and organization of this guide.

- “Document Scope and Audience” on page 1-1
- “Guide to this Document” on page 1-1
- “Related Documentation” on page 1-2
- “Contact Us!” on page 1-3

Document Scope and Audience

This document is a resource for system administrators, database administrators, and software developers who need to install and configure BEA AquaLogic Enterprise Security™ Security Service Modules. It contains information that is relevant during the design, staging, and production deployment phases of a software project.

It is assumed that readers understand Web technologies and have a general understanding of the Microsoft Windows or UNIX operating systems. Prior to using this document, you should have a general understanding of the principal components and architecture of BEA AquaLogic Enterprise Security. Read the *Introduction to BEA AquaLogic Enterprise Security* for conceptual information that is helpful in understanding how the product works.

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 6-1 describes how to configure the Oracle SSM.
- “Configuring the WebSphere SSM” on page 7-1 describes how to configure the WebSphere 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 AquaLogic Enterprise Administration Application, see *Installing the Administration Server*.

For information about other aspects of ALES, see the following documents:

- *Introduction to BEA AquaLogic Enterprise Security* provides overview, conceptual, and architectural information for AquaLogic Enterprise Security.
- Getting Started with ALES 3.0 provides a number of tutorials that show how to use the Entitlements Management tool to secure application resources.
- *Policy Managers Guide d*efines the ALES 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.
- *Programming Security for Web Services* describes how to implement security in web servers.

- *Developing Security Providers for BEA AquaLogic Enterprise Security* provides security vendors, administrators, and application developers with information needed to develop custom security providers.
- *Javadocs for Java API* provides reference documentation for the Java API provided in this release.
- *Wsdldocs for Web Services API* provides reference documentation for the Web Services APIs provided in this release.
- *Javadocs for Security Service Provider Interfaces* provides reference documentation for the Security Service Provider Interfaces.
- *Javadocs for BLM API* provides reference documentation for the Business Logic Manager (BLM) Application Programming Interfaces.

Contact Us!

Your feedback on BEA documentation is important to us. Send questions or comments to docsupport@bea.com. In the message, please indicate the software name and version, as well as the title and date of the documentation. Your comments will be reviewed by the BEA documentation professionals.

If you have technical questions about this version of BEA AquaLogic Enterprise Security or experience problems installing and running the product, contact BEA Customer Support through BEA Web Support at http://www.bea.com or by using the contact information provided on the Customer Support Card included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Name, e-mail address, phone and fax numbers
- Company name and address
- Machine type and authorization codes
- Product name and version
- Description of the problem and the content of pertinent error messages.

Overview

BEA AquaLogic Enterprise Security includes components named Security Service Modules (SSMs) that are installed on the machines hosting the applications to be secured. An SSM ties the secured application into ALES so that all administrative security activities are performed through the Administration Server.

The following out-of-box SSMs are available in this release of AquaLogic Enterprise Security.

- 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 ALES documents, particularly the Policy Manager’s Guide, Getting Started Tutorials, and the help systems for the Administration Console and Entitlements Management Tool.

Distribution

BEA AquaLogic Enterprise Security products are distributed and installed using the BEA Installation and Distribution System, which provides a complete framework for the following:

- Distribution of BEA products by download from the BEA web site.
- ALES Installation and uninstallation, including documentation.

Web Distribution

To download the product from the BEA web site, contact BEA Sales at http://www.bea.com/framework.jsp?CNT=sales1.htm&FP=/content/about/contact/.

An SSM installer package is available for each supported operating system. The same package is used to install all SSM types.

Installing SSMs

This sections provides step-by-step instructions for installing and enrolling Security Service Modules (SSM), and performing some additional post-installation tasks.

The following topics are described:

Installation Requirements

The ALES Security Service Modules require certain software components to operate properly.

System Requirements

Table 3-1 lists the system requirements for machines on which SSMs are installed.

Table 3-1 System Requirements

SSM	Platform Version(s)	Windows 2000, 2003[1]	Solaris 8, 9, 10	RHAS[2] 3.0, 4.0	Suse[3] 9.2, 10.0	AIX[4] 5.3
Web Service	MS .NET 1.1 and 2.0[5] WebLogic Workshop 9.0, 10.0 Studio 3.0	Yes	Yes	Yes	Yes	Yes
BEA WebLogic Platform	WLS/P 8.1 SP5, SP6 WLS/P[6] 9.2 MP2, 10.0 MP1 WLI 9.2 MP2	Yes	Yes	Yes	Yes	No
BEA AquaLogic Products	ALDSP 2.5, 3.0[7] ALSB 2.6, 3.0[8] ALBPM 6.0	Yes	Yes	Yes	Yes	No
IBM WebSphere	WebSphere App Server 6.1	Yes	Yes	Yes	Yes	Yes
Java	Sun JVM 1.4.2, 5.0, 6.0 JRockit 1.4.2, 5.0, 6.0 IBM JDK 1.4.2, 5.0, 6.0	Yes	Yes	Yes	Yes	Yes
Web Servers	Apache IIS Web Server	Yes Yes	Yes No	Yes No	Yes No	No No

[1] Windows 2000 SP4 and higher, Windows 2003 R2 and higher.

[2] RedHat Advanced Server.

[3] Suse Linux is supported on both 32-bit and 64-bit hardware.

[4] AIX SSM support will be delivered post-GA as a CP to ALES 3.0.

[5] NET Web Services client on Windows 2000 and 2003 only.

[6] Works with WLS configured to use either the Sun JVM or the JRockit JVM that ship with the 9.x or 10.x version of the server. JRockit JVM supported on Intel hardware only.

[7] ALDSP 2.5 running on WLS 8.1.x, ALDSP 3.0 running on WLS 10.0 MP1.

[8] ALSB 2.6 running on WLS 9.2, ALSB 3.0 running on WLS 9.2 MP1 and WLS 10.0 MP1.

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 software distribution to a local drive on the machine and install it from there.

Installation Topology

Although it is possible to install the ALES Administration Server and SSMs on the same system, this is not recommended in a production environment. This document assumes that SSMs and the Administration Server are being installed on separate machines.

SSM Upgrades

ALES 3.0 includes a utility to upgrade from ALES 2.2 and 2.6. 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 ALES files. You must be logged in as a member of the group used when the earlier ALES 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.
5. When the SSM installer runs, it detects the earlier versions and uses its configuration information. Respond to the prompts as required.

Installation Steps

To install an SSM:

1. Shut down any running programs.
2. Locate and launch the installation program. This varies by operating system, as shown in Table 3-2.

Table 3-2 SSM Installation Programs

Windows	Launch `ales300ssm_win.exe`
	Note: To generate a verbose installation log, add the following to the launch command:
	`-log=<logfile> -log_priority=debug`
	Example:
	`ales300ssm_win32.exe -log=D:\logs\ales_install.log -log_priority=debug`

Solaris	1. Change the protection on the install file by entering: `chmod u+x ales300ssm_solaris32.bin`.
	2. Enter: `ales300ssm_solaris32.bin`
	Note: To generate a verbose installation log, add the following to the launch command:
	`-log=<logfile> -log_priority=debug`
	Example:
	`ales300ssm_solaris32.bin -log=/opt/logs/ales_install.log -log_priority=debug`

Linux	1. Change the protection on the install file by entering chmod u+x `ales300ssm_rhas_IA32.bin`. 2. Enter: `ales300ssm_rhas_IA32.bin` **Note**: To generate a verbose installation log, use same command string as described for Solaris.

3. Complete the prompts using Table 3-3.

Table 3-3 SSM Installation Prompts

Window	Action
Welcome	Click **Next**.
BEA License Agreement	Select **Yes** and then click **Next**.
Choose BEA Home Directory	Accept the default location (recommended) or specify a different one and click **Next**.

Table 3-3 SSM Installation Prompts

Choose Components	Select the SSMs to install and click **Next**. Only installable components are listed. For example, if installing on WebLogic Server 9.2, 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 ALES products, you will see **Installation Complete**. Otherwise, continue.
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 Service Control Manager Directory	Accept the default directory where the SCM will be installed or specify a different one. Then click **Next**.
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 as described in **"Define a SCM in the ALES 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 Java JDK for the SSM	Accept the default selection or specify a different JDK and click **Next**.
Choose Java JDK for the SCM	Accept the default selection or specify a different JDK and click **Next**.

4. 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 ALES component on a remote machine registers with the Administration Server. As part of this process, the SSM system exchanges security certificates with the ALES Administration Server.

All ALES components located under a `BEA_HOME` directory use the same set of keys located in `BEA_HOME/ales30-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\ales30-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 ALES 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/ales30-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 that the Administration Server is running and configured for 1-way SSL. For further details, see SSL for Production Environments.
2. If the SSM is using an SCM, make sure the SCM is running.
3. In the `BEA_HOME/ales30-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 `system` and `weblogic` 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 *ALES Administration Reference*.

Example of Running Enroll

```
D:\bea\ales30-shared\bin>set-env
D:\bea\ales30-shared\bin>enroll secure
======================================================================
AquaLogic Enterprise Security Enrollment/Unenrollment Utility
======================================================================
Enter admin username :> system
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 a SCM in the ALES Database

Use the Administration Console to define an SCM in the ALES database. 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 in the ALES database, 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. Otherwise, the name can be of your choosing. For details, see Table 3-3, "SSM Installation Prompts," on page 3-4.

You must define the SCM even if the SSM does not use an SCM 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, you must use the asipassword utility to set the Administration Server's `system` user password on the SSM machine. This password is required to secure communications between the SSM and the Administration Server.

To run the tool:

1. Change to the `BEA_HOME\ales30-shared\bin` directory and enter the following:

   ```
   asipassword system <BEA_HOME>\ales30-shared\keys\password.xml
   <BEA_HOME>\ales30-shared\keys\password.key
   ```

   Example:

   ```
   asipassword system c:\bea\ales30-shared\keys\password.xml
   c:\bea\ales30-shared\keys\password.key
   ```

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

Notes:

- The `password.xml` file does not exist until after the enrollment process is performed.

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 6-1.
- For the Websphere SSM, see “Configuring the WebSphere SSM” on page 7-1.
- For a custom SSM, see “Configuring a Custom SSM” on page 8-1.

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, 10.0 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 version of ALES 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 `C:\bea\ales30-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
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.
Resources	The resource root that will serve as the parent resource of the application resources to which authorization policies will be assigned. The name to use for creating the resource root 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 loads policies from temporary directory.
3. Uses the SetPassword tool to set the password for the Admin Server `system` user.
4. For WebLogic domains, edits the StartWeblogic script in the domain, inserts ALES JAR files to the CLASSPATH, and adds ALES "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 (ALES and others). 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 in the ALES system. Example: `ssm.conf.id = MyAppName`
db.password	The ALES database user password. The name of the ALES 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 ALES administrator's password. The ALES administrator's default user name and password is `system` and `weblogic` 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 `aldsp-ssm` — ALDSP-based domain in WebLogic 8.x `wls-ssm` — WebLogic 9.x or 10.x domain `wls-portal-ssm` — Portal-based domain in WebLogic 9.x/10.x `wls-alsb-ssm` — ALSB-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 ALES 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 ALES administrator's username. The ConfigTool will prompt for this value if it is not specified in the properties file. The ALES administrator's default user name and password is `system` and `weblogic` respectively. Example: `ales.admin.name = system`
ssm.instance.name	The name that will be assigned to the SSM instance. Example: `ssm.instance.name` = `MySsm`
ales.resource.root	The name that will be used to create the root resource under which the application resources to be secured will be defined. The root resource must be preceded by `//app/policy/` Example: `ales.resource.root` = `//app/policy/MyApp`
ales.identity.dir	A name that will be used to create the Identity directory that will be used to define and manage the application's users and groups. Example: `ales.identity.dir = MyDir`

Table 4-2 Properties File Modifications

Field	Description
Database JDBC URL	(REQUIRED ONLY IF THE ADMINISTRATION SERVER IS ON A SEPARATE MACHINE) The JDBC connection string to the ALES 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 ONLY IF THE 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, ALES does not include the JDBC driver for MS SQL, PointBase, or DB2, but 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=F:/bea/ales30-ssm/webservice-ssm/lib/sslclient.j
ar
wrapper.java.classpath.49=F:/bea/ales30-ssm/webservice-ssm/lib/pdsoap11.ja
r
wrapper.java.classpath.50=F:/bea/ales30-ssm/webservice-ssm/lib/antlr.jar
wrapper.java.classpath.51=F:/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
```

Configuring the Web Server SSM

This section describes tasks you must perform after installing the Web Server SSM. This SSM supports Microsoft IIS and Apache Web Server.

The following tasks are described:

Requirements

The Web Server SSM uses the Web Service SSM. Before performing the tasks described in this chapter, you must configure the Web Service SSM as described in "Configuring SSMs Using ConfigTool" on page 4-1.

Creating a Web Server SSM Instance

To create a Web Server SSM instance:

1. Start the Instance Wizard:
   - On Windows, click Start > Programs > BEA AquaLogic Enterprise Security > *<Type of Security Service Module>* > Create New Instance.
   - On UNIX, if you are using X-windows**,** go to *BEA_HOME*/ales30-ssm/*<ssm-type>*/adm and enter: instancewizard.sh. If you are not using X-windows, use a console based installer.
2. In the **Instance Name** field, enter an instance name. Then specify Web Service SSM port number and click **Next**.
3. In the **Location** field, accept the default location or specify a different directory and click **Next**.
4. When wizard completes, click **Done**.

Notes:

- After creating an Apache Web Server SSM, you must add the Apache user to the asiusers group. This gives the Administration Server the permissions required to access the Apache Web Server SSM instance and deploy the security policy and the security configuration.
- For the IIS Web Server SSM, Table 5-1 indicates information that is added to the Microsoft Windows Registry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\BEA Systems\ALES\IIS Module\3.0
```

Table 5-1 IIS SSM Registry Entries

Value Name	Description/Setting
ALES_HTTP_SERVER	The IIS SSM instance's config directory.
ALES_INSTALL_DIR	The SSM's installation directory.
ALES_LOG_LEVEL	The log level. The default level is 2 (INFORMATIONAL).

Enrolling the Web Server SSM Instance

Enrollment is the process by which an ALES component such as an SCM or SSM registers with an Administration Server.

As part of this process, the SSM system exchanges security certificates with the associated ALES Administration Server.

You must have the Administration Server running prior to enrolling the Security Service Module.

Note: While you can use the demonstration digital certificate in a development environment, you should never use it in a production environment.

To enroll the Security Service Module:

1. Open a command window and go to the Security Service Module instance `/adm` directory: `BEA_HOME/ales30-ssm/<`*`ssm-type`*`>/instance/`*`instancename`*`/adm`, where *`instancename`* is the name you assigned to the instance when you created it.

2. Run the following script:

   ```
   enroll demo
   ```

3. Enter the `admin` username and password. This is the username and password of the Security Administrator doing the enrollment (if you used the default values and have not yet changed them, the default username is `system` and the password is `weblogic`).

4. Enter and confirm the following passwords:

   – **Private key password**—This password protects the identity of the Security Service Module that you are creating.

   – **identity.jceks password**—This password protects the `ssl\identity.jks` keystore. This keystore contains the identities for all the components you are enrolling.

   – **peer.jks password**—This password protects the `ssl\peer.jks` keystore. This keystore contains the certificates of components with which this Security Service Module can communicate.

   – **trust.jks password**—This password protects the `ssl\trust.jks` keystore. This keystore contains the AquaLogic Enterprise Security CA certificate used for enrollment.

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

Start the Web Service SSM

To start an instance of the Web Service SSM on Windows:

- Click Start > Programs > BEA AquaLogic Enterprise Security > Security Service Module > Web Service Security Service Module> *instancename* > Start Web Service (console mode). The Start Web Service command windows appears and indicates that the Web Service SSM started.

To start an instance of the Web Service SSM on UNIX:

- Open another command prompt, cd to `BEA_HOME/ales30-ssm/webservice-ssm/instance/<`*`instancename`*`>/bin` and enter `WLESws.sh start`, where <*instancename*> is the name of the Web Service SSM.

Create and Bind the Corresponding SSM

This section describes how to:

1. Use the Administration Console to create a corresponding SSM.
2. Specify which security providers it will use.
3. Bind it to the SCM in the Administration Console.

Create a Corresponding SSM in the Administration Console

The Web Service SSM and the Web Server SSM are installed as a combo. The Web Service SSM instance you created on the remote system has a Configuration ID that will correspond to the SSM you create in the Administration Console. (The Web Server SSM does **not** have a Configuration ID.)

That is, in your ALES environment you will have two SSMs with the same Configuration ID: one (the instance) on the system where the SSM is installed, and a corresponding one on the Administration Server system.

Perform the following steps:

1. In the Administration Console, expand the Security Configuration node in the left pane, and click Unbound Configurations. The Unbound Security Service Module Configurations page displays.
2. Click Create a New Security Service Module Configuration. The Edit Security Service Module Configuration page displays.
3. In the Configuration ID text box, enter the matching Configuration ID for the SSM.
4. In the Configuration Version dropdown box, select `Java SSM 3.0, WS SSM 3.0`.
5. Click Create.

Add Security Providers

After you create the SSM, the Providers tab becomes active. The tab for a given provider type provides additional information. At a minimum you must have one authorization provider for each SSM.

Click the Providers tab and configure the desired providers.

Bind Everything Together

1. Click on the SCM that you previously configured for this SSM. The Edit a Service Control Manager Configuration page displays.
2. Click on the Binding tab and bind your new SSM configuration to the SCM.

Configuring the Web Server Environmental Binding

The Web Server Environmental Binding configuration procedures vary depending on the type of Web Server SSM you are configuring. AquaLogic Enterprise Security supports two Web server SSMs that require configuration of the Web Server Environmental Binding: the Microsoft IIS Web Server SSM and the Apache Web Server SSM.

Microsoft IIS Web Server

To configure the environmental binding for Microsoft IIS Web Server, perform the following tasks:

- "Configuring the Microsoft IIS Web Server Binding Plug-In File" on page 5-5
- "Configuring the NamePasswordForm.acc File for the IIS Web Server" on page 5-10
- "Deploying and Testing the IIS Web Server Sample Application" on page 5-10

Configuring the Microsoft IIS Web Server Binding Plug-In File

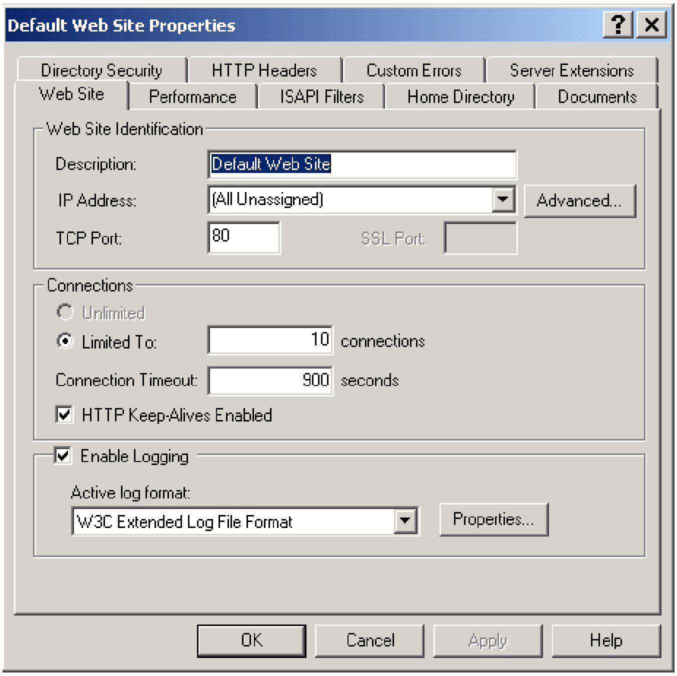
Note: This task assumes you have created an instance of the IIS Web Server SSM.

The IIS Web Server Binding plug-in file is named `wles_isapi.dll`. This file is located in the *BEA_HOME*`\ales30-ssm\iis-ssm\lib` directory.

To configure the Microsoft IIS Web Binding plug-in, perform the following steps:

1. To open the Internet Information Services Manger, click Start>Settings>Control Panel, select Administrative Tools, and double-click Internet Services Manager. The Internet Information Services Window appears.
2. In the left-hand pane, expand the machine node, right click Default Web Site, and select Properties. The Default Web Site Properties dialog box appears (see Figure 5-1).

Figure 5-1 IIS Web Site Properties Dialog



Web Site Identification

Description: Default Web Site

IP Address: (All Unassigned)

TCP Port: 80 SSL Port:

Connections

○ Unlimited

● Limited To: 10 connections

Connection Timeout: 900 seconds

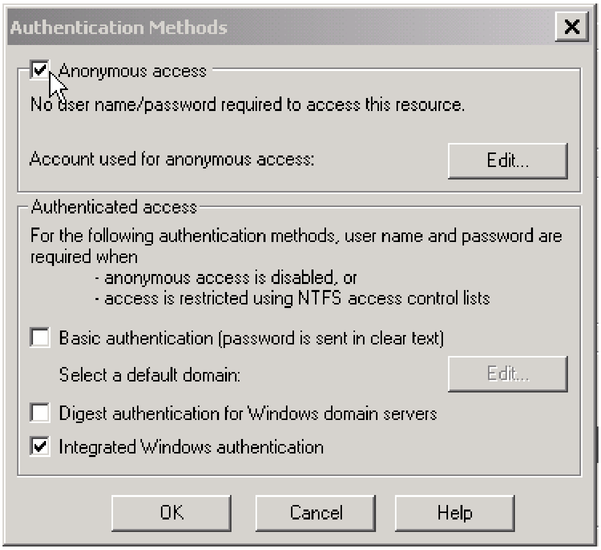
☑ HTTP Keep-Alives Enabled

☑ Enable Logging

Active log format: W3C Extended Log File Format

3. Click the ISAPI Filters tab, click the Add button, assign a name to the ISAPI filter, use the Browse button to add the `wles_isapi.dll` file (which is located in *BEA_HOME*`\ales30-ssm\iis-ssm\lib` directory), and click OK.
4. Click the Directory Security tab. The Authentication Methods dialog appears (see Figure 5-2).

Figure 5-2 Authentication Methods Dialog



Authentication Methods

☑ Anonymous access

No user name/password required to access this resource.

Account used for anonymous access:

Authenticated access

For the following authentication methods, user name and password are required when

- anonymous access is disabled, or
- access is restricted using NTFS access control lists

□ Basic authentication (password is sent in clear text)

Select a default domain:

□ Digest authentication for Windows domain servers

☑ Integrated Windows authentication

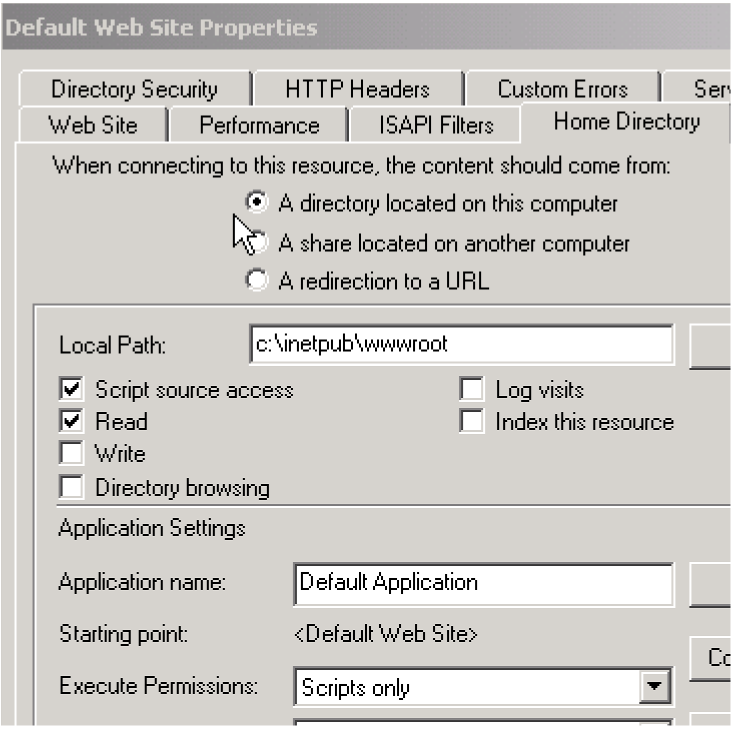
5. Click the Edit button for Anonymous Access, check the Anonymous username, and, if necessary, change the username and password to ensure that the Anonymous user has `Read` and `Read/Execute` permissions on the following directories:

   ```
   BEA_HOME\ales30-ssm\iis-ssm\lib
   BEA_HOME\ales30-ssm\iis-ssm\instance\iisssmdemo\ssl
   BEA_HOME\ales30-ssm\iis-ssm\instance\iisssmdemo\config
   ```

6. If you put the `NamePasswordForm.acc` file in a virtual directory, repeat the previous step for the virtual directory as well.

7. Return to the Default Web Site Properties dialog box (see Figure 5-1) and click the Home Directory tab. The Home Directory dialog appears (see Figure 5-3).

Figure 5-3 IIS Web Site Home Directory Dialog



Default Web Site Properties

When connecting to this resource, the content should come from:

● A directory located on this computer
○ A share located on another computer
○ A redirection to a URL

Local Path: c:\inetpub\wwwroot

☑ Script source access
☑ Read
☐ Write
☐ Directory browsing
☐ Log visits
☐ Index this resource

Application Settings

Application name: Default Application

Starting point: <Default Web Site>

Execute Permissions: Scripts only

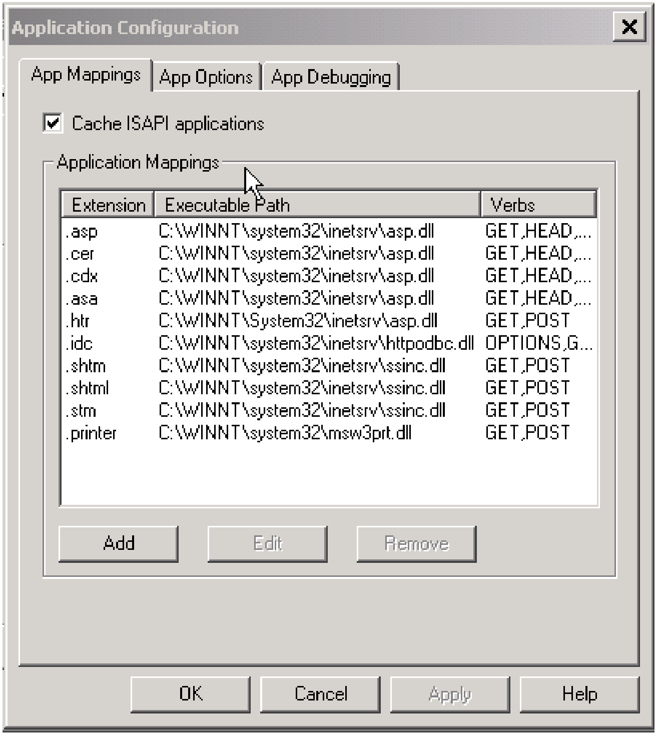
8. Verify that the property settings match the information in Table 5-2 and click Apply and OK.

Table 5-2 Home Directory Setting

Property	Setting
Local Path	`c:\inetpub\wwwroot`
Application name	Default Application
Execute Permissions	Scripts Only

9. Click the Configuration button. The Application Configuration dialog appears (see Figure 5-4).

Figure 5-4 IIS Web Site Application Configuration Dialog



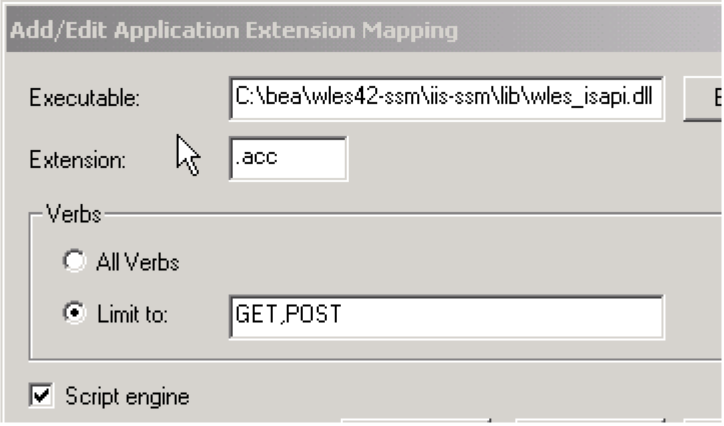
☑ Cache ISAPI applications

Application Mappings

Extension	Executable Path	Verbs
.asp	C:\WINNT\system32\inetsrv\asp.dll	GET,HEAD,...
.cer	C:\WINNT\system32\inetsrv\asp.dll	GET,HEAD,...
.cdx	C:\WINNT\system32\inetsrv\asp.dll	GET,HEAD,...
.asa	C:\WINNT\system32\inetsrv\asp.dll	GET,HEAD,...
.htr	C:\WINNT\System32\inetsrv\asp.dll	GET,POST
.idc	C:\WINNT\system32\inetsrv\httpodbc.dll	OPTIONS,G...
.shtm	C:\WINNT\system32\inetsrv\ssinc.dll	GET,POST
.shtml	C:\WINNT\system32\inetsrv\ssinc.dll	GET,POST
.stm	C:\WINNT\system32\inetsrv\ssinc.dll	GET,POST
.printer	C:\WINNT\system32\msw3prt.dll	GET,POST

10. Click the Add button. The Add/Edit Application Extension Mapping Dialog appears (see Figure 5-5).

Figure 5-5 IIS Web Site Add/Edit Application Extension Mapping Dialog



Add/Edit Application Extension Mapping

Executable: C:\bea\wles42-ssm\iis-ssm\lib\wles_isapi.dll

Extension: .acc

Verbs

○ All Verbs

● Limit to: GET,POST

☑ Script engine

11. Use the Browse button to add the `wles_isapi.dll` file to the Executable field, fill in the other fields as shown in Figure 5-5, and click OK.

12. Click OK to close the remaining windows.

13. Right click the Default Web Site again and start the Default Web Site. (Stop the Web Site first if necessary.)

14. Re-open the Default Web Site Properties dialog box and select the ISAPI Filters tab. The IIS Web Server Binding Plug-in status shows a green arrow to indicate that the IIS Web Server Binding Plug-in is loaded. If the green arrow is not displayed, add the `wles_isapi.dll` file again and start the IIS Web Server.

    Note: Be sure to start the IIS Web server with IIS SSM after you have started the Web Service SSM and ARME.

Configuring the NamePasswordForm.acc File for the IIS Web Server

Configure the `NamePasswordForm.acc` file for the IIS Web Server as follows:

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

Deploying and Testing the IIS Web Server Sample Application

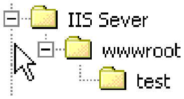
To set up the sample web application, perform the following steps:

Note: The Web Service SSM must be started before you perform this task because the filter and extension attempt to connect to the Web Service SSM when they are loaded by the Web server.

1. Set up the `IIS Server/wwwroot/test` directory as shown in Figure 5-6 and copy the following files to the `test` directory:
   - `NamePasswordForm.acc`
   - `foo.html`
   - `atnfailure.html`
   - `atzfailure.html`

   Note: The `NamePasswordForm.acc` file is provided in the *`BEA_HOME`*`\ales30-ssm\iis-ssm\instance\<`*`instancename`*`>\templates` directory. The `foo.html`, `atnfailure.html` and `atzfailure.html` files are not provided in the product installation kit. You should use your own versions of these files.

Figure 5-6 Deploying the Sample Application on the IIS Web Server



2. Start the IIS Web Server, open a browser and go to `http://<machine_name_with DNS_suffix>:80/test/foo.html`.
3. You are redirected to `NamePasswordForm.acc.`
4. Enter the system username/password (a default system username and password was set when you installed the Administration Application) and click OK. You are granted access to `foo.html`.

Apache Web Server

To configure the Apache Web Server, perform the following tasks:

- "Downloading and Installing the Apache Web Server" on page 5-12
- "Configuring the ALES Module" on page 5-12
- "Configuring the NamePasswordForm.html File for the Apache Web Server" on page 5-13

- "Deploying and Testing the Apache Web Server Sample Application" on page 5-13

Downloading and Installing the Apache Web Server

To download and install the Apache Web Server software, perform the following steps:

1. Go to the Apache download web site at `http://httpd.apache.org/download.cgi` and download and install the software.

2. Verify the following two modules are included in the installation:

   – *`ServerRoot`*`/modules/mod_include.so`

   – *`ServerRoot`*`/modules/mod_ssl.so`

   where *`ServerRoot`* is the Apache installation directory.

   Note: The Apache Web Server Security Service Module (SSM) requires that the above two modules be included in the Apache installation; otherwise the Secure Sockets Layer (SSL) and the Security Assertion Markup Language (SAML) server-server include (SSI) related functions will not work.

   Note: You may build your own 2.0.x version of the Apache Web Server with the above mentioned modules. If the modules are built into Apache, there may be no such files.

Configuring the ALES Module

The ALES module contains only one file. For Windows, the file name is `mod_wles.dll`. For Sun Solaris and Linux, the file name is `mod_wles.so.`

To install and configure the ALES module:

1. Open the *`ServerRoot`*`/conf/httpd.conf` file and add a `LoadModule` directive. There are several `LoadModule` directives in the LoadModule section of the `httpd.conf` file. Add the following line to the end of the LoadModule section:

   ```
   LoadModule wles_module <APACHE_SSM_HOME>/lib/mod_wles.so
   ```

   where <*`APACHE_SSM_HOME`*> is the Apache Web Server SSM installation directory.

   For example:

   For Windows systems:

   ```
   LoadModule wles_module c:\bea\ales30-ssm\apache-ssm\lib\mod_wles.dll
   ```

   ```
   For UNIX systems:
   ```

   ```
   LoadModule wles_module
   /home/tiger/bea/ales30-ssm/apache-ssm/lib/mod_wles.so
   ```

2. Add a `WLESConfigDir` directive right after the above `LoadModule` directive as follows:

   ```
   <IfModule mod_wles.cpp>
   WLESConfigDir <APACHE_SSM_HOME>/instance/<instance_name>/config
   </IfModule>
   ```

   where the `config` directory is the directory that contains the `default.properties` file.

   Note: In the `IfModule` condition, be sure to specify `mod_wles.cpp`, not `mod_wles.c`.

3. To make sure your server works properly, configure the `ServerName`. For example:

   ```
   ServerName www.yourservername.com:8080
   ```

4. Change the Group directive to have the Apache Web Server running as the `asiusers` group so it can read the `mod_wles` file and other required files:

   ```
   Group asiusers
   ```

5. Edit the `envvars` file in the `ServerRoot/bin` directory, appending the directory where `mod_wles.so` resides to the default `LD_LIBRARY_PATH`, so that the file looks like this:

   ```
   LD_LIBRARY_PATH="/www/apache/lib:$LD_LIBRARY_PATH:<APACHE_SSM_HOME>/lib"
   ```

   Note: This step ensures that the Apache Web Server can load the dependency libraries for the `mod_wles` file.

6. Use the Apache `ctl` script to start or restart Apache Web Server in the `ServerRoot/bin` directory.

Configuring the NamePasswordForm.html File for the Apache Web Server

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

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

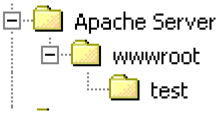
Deploying and Testing the Apache Web Server Sample Application

To set up the sample web application, perform the following steps:

1. Set up the `Apache Server/wwwroot/test` directory as shown in Figure 5-7 and copy the following files to the `test` directory:
   - `NamePasswordForm.html`
   - `foo.html`
   - `atnfailure.html`
   - `atzfailure.html`

Note: The `NamePasswordForm.html` file is provided in the `*BEA_HOME*\ales30-ssm\apache-ssm\instance\<*instancename*>\templates` directory. The `foo.html`, `atnfailure.html` and `atzfailure.html` files are not provided in the product installation kit. You should use your own versions of these files.

Figure 5-7 Deploying the Sample Application on the Apache Web Server



2. Start the Apache Web Server, open a browser, and go to `http://<hostmachine.cookiedomain>:8088/test/foo.html`.
3. You are redirected to `NamePasswordForm.html`
4. Enter the system username/password (a default system username and password was set when you installed the Administration Application) and click OK. You are granted access to `foo.html`.

Configuring Web Server SSM Properties

The Web Server SSM has a configuration file named `default.properties`. All configuration settings for the Web Server SSM instance are defined in this file. This file is pre-configured and placed in the proper location for you.

If you want to edit the `default.properties` file for your particular environment, refer to the parameters descriptions in the following sections:

- "Session Settings" on page 5-15
- "Authentication Settings" on page 5-15
- "Role Mapping Settings" on page 5-20
- "Credential Mapping Settings" on page 5-21
- "Naming Authority Settings" on page 5-22
- "Logging Level Setting" on page 5-23
- "Environment Variables Accessible Using CGI" on page 5-23

Session Settings

The AquaLogic Enterprise Security services are stateless services; it is the calling Web Service client that is responsible for determining session related information. In addition, in a web environment, a session does not necessarily end with an explicit logout, so session termination must be inferred from a lack of activity.

Table 5-3 describes the settings used to manage session behavior. You use these settings to configure the Web server session related behavior for the security configuration to which it applies.

Table 5-3 Session Settings

Session Setting	Description
`session.inactivity.timeout`	The number of seconds of inactivity that causes a session to expire. Default value: 600 seconds (10 minutes)
`session.absolute.timeout`	The number of seconds an active session is allowed to be available before it expires and the user is forced to re-authenticate. If this setting is set to zero, then established active sessions can continue indefinitely. Default value: 3600 seconds (60 minutes)
`session.cookie.name`	The name of the session cookie. Default value: ALESIdentityAssertion.
`session.forcedlogoffURL`	The name of the URL that, when accessed, forces the session to logoff.

Authentication Settings

Table 5-4 describes the settings that you use to configure the Web server authentication behavior for the security configuration to which it applies. Also, for information on mapping JAAS Callbacks, see "Mapping JAAS Callback Type to Form and Form Fields" on page 5-17.

Table 5-4 Authentication Settings

Authentication Setting	Description
`authentication.precedence`	An ordered, comma-separated list of types of identity creation. If identity information is available from multiple types of identity transfers, this list determines which identity to use. The valid identity type is: • FORM — credential information collected from an authentication provider using forms. Default value: `FORM`
`authentication.initialForm`	Specifies the first form presented for form-based authentication.
`authentication.<callback type>[<prompt>] = <field>,<form URL>`	Given a question, this setting specifies what field on what form will answer that question. Notice that the <prompt> is shown as optional. However, the prompt is required if there are multiple callbacks of the same type, because there is no other way for the SSM to distinguish identical callback types. The prompt is obtained from the callback by calling the `getPrompt()` method, but it is not used in the display of the form. If the prompt setting is missing, then the Web Server SSM attempts to answer the callbacks in the order of the settings. If the order does not match the order of the providers, then authentication fails. For more information on using this setting, see "Mapping JAAS Callback Type to Form and Form Fields" on page 5-17.
`authentication.onatnfailure`	If authentication fails, and this setting is set to a URL, then rather than issuing a 401 Authentication Failed, the user will be redirected to the specified URL.
`authentication.onatzfailure`	If authorization fails and this setting is set to a URL, then rather than issuing a 403 Permission Denied, the user is redirected to the specified URL.

Table 5-5 describes the different types of authentication callbacks that are supported by the Web Server SSM.

Table 5-5 Authentication Callback Type Descriptions

Authentication Callback Type	Description
`authentication.nameCallback`	The form template responsible for collecting a name for a name callback. This form must exist in the same directory as the post handler.
`authentication.passwordCallback`	The form template is responsible for collecting a password for a password callback. This form must exist in the same directory as the post handler.
`authentication.choiceCallback`	The form template is responsible for collecting a choice for a choice callback. This form must exist in the same directory as the post handler.
`authentication.confirmationCallback`	The form template is responsible for collecting a confirmation for a confirmation callback. This form must exist in the same directory as the POST handler.
`authentication.textInputCallback`	The form template is responsible for collecting some text input for a text input callback. This form must exist in the same directory as the post handler.

Mapping JAAS Callback Type to Form and Form Fields

There are two required and one optional configuration settings that specify what form and what field contain the information required to satisfy the authentication callbacks. The credential gathering form must use an HTTP `POST` method to specify this information. Listing 5-1 shows an example of how to use the `POST` method in the credential gathering form.

Listing 5-1 Example of Using the POST Method in the Credential Gathering Form

```
<FORM METHOD=POST ACTION="LoginNamePwdTextIn.html">
<!--#AUTHSTATE -->
<TABLE BGCOLOR="C0C0C0"><TR><TD>
<TABLE BGCOLOR="#FFFFFF">
<TR><TD COLSPAN="2" BGCOLOR="#C0C0C0">Please Login</TD></TR>
<TR><TD COLSPAN="2">User Name&nbsp;</TD><TR>
<TR><TD><!--#PROMPT --></TD><TD><INPUT NAME="username"></TD></TR>
```

```
<TR><TD COLSPAN="2">Password&nbsp;</TD><TR>
<TR><TD><!--#PROMPT.1--></TD><TD><INPUT TYPE=
             PASSWORD NAME="password"></TD></TR>
<TR><TD COLSPAN="2">Input Text&nbsp;</TD><TR>
<TR><TD><!--#PROMPT --></TD><TD><INPUT NAME="textinput"></TD></TR>
<TR><TD COLSPAN="2">&nbsp;</TD><TR>
<TR><TD COLSPAN="2" ALIGN="CENTER"><INPUT TYPE="SUBMIT"
VALUE="OK"></TD><TR>
</TABLE>
</TD></TR></TABLE>
</FORM>
```

The form field defines the HTTP POST data name that results from a submitted form.

The settings have the following format:

```
authentication.<callback type>[<prompt>] = <field>:<form URL>
```

Given a question, this setting specifies what field on what form will answer that question. Notice that the <prompt> is shown as optional. However, if there are multiple callbacks of the same type, the <prompt> is required because there is no other way for the Web Server SSM to distinguish identical callback types. The <prompt> is obtained from the callback by calling the getPrompt() method, but it is not used in the display of the form. If the <prompt> setting is missing, then the Web Server SSM attempts to answer the callbacks in the order of the settings. If the order does not match the order of the authentication providers, then authentication fails.

The supported callback types are: nameCallback, passwordCallback, textInputCallback, textOutputCallback.

Table 5-6 provides examples of callback usage and more information on each supported callback type.

Table 5-6 Authentication Callback Usage Examples

Authentication Callback Types	Example/Discussion

Name and password callbacks	`authentication.nameCallback[]=username: /ales/NamePasswordForm.htm`
	`authentication. passwordCallback []= password: /ales/NamePasswordForm.htm`

Name, password, and textInput callbacks	`authentication.initialForm=/test/NamePasswordForm.html`
	`# username/password`
	`authentication.nameCallback[]=username:/test/ NamePasswordForm.html`
	`authentication.passwordCallback[]=password:/test/ NamePasswordForm.html`
	`# username/password/textInput`
	`authentication.nameCallback[]=username:/test/ LoginNamePwdTextIn.html`
	`authentication.passwordCallback[]=password:/test/ LoginNamePwdTextIn.html`
	`authentication.textInputCallback[]=textinput:/test/ LoginNamePwdTextIn.html`
	In this example the user will be prompted for username/password. The authentication provider then prompts for the user's mother's maiden name. The Web Server SSM redirects to `QuestionForm.htm` and knows from what field to get the information.

Table 5-6 Authentication Callback Usage Examples (Continued)

Authentication Callback Types	Example/Discussion

Name, password, and textInput callbacks	`authentication.nameCallback[]=username: /ales/NamePasswordForm.htm`
	`authentication. passwordCallback []= password: /ales/NamePasswordForm.htm`
	`authentication. textInputCallback ["maiden name"]=maiden_name: /ales/ QuestionForm.htm`
	`authentication. textInputCallback ["social security number"]=maiden_name: /ales/ QuestionForm.htm`
	In this example two providers require username/password callbacks, a third provider requires a textInputCallback for mother's maiden name, and a fourth provider requires a textInputCallback for a Social Security number: The prompts distinguish between the two textInputCallbacks.
	Note: The textInputCallback prompt requires quotes only if it contains embedded strings.

TextOutput Callback	The `textOutputCallback` is used to display a message to the user. Because the Web Server SSM does not create or update forms, if it gets a textOutputCallback, it redirects it to the form URL and adds the field as a query string argument and the message value. The application that processes the URL is responsible for parsing the query string and displaying the message.
Language callback	Language callbacks are handled internally by the Web server; the user is never prompted, so no configuration is needed. The user's browser Accept-Language header is checked for the preferred language it supports and that locale is returned to the authentication provider. If the user's browser has no Accept-Language header, the system default locale is used.

Role Mapping Settings

Table 5-7 describes the settings that you use to configure the Web server role mapping behavior for the policy domain to which it applies.

Table 5-7 Role Mapping Settings

Role Mapping Setting	Description
`rolemapping.enable`	If set to `true`, then roles are injected into the request stream as a comma separated list.
`rolemapping.name`	The name of the variable in which to put the roles. The default is: `ALES_ROLES.`

Credential Mapping Settings

Table 5-8 describes the settings that you use to configure the Web server credential mapping behavior for the policy domain to which it applies.

Table 5-8 Credential Mapping Settings

Credential Mapping Setting	Description
`credentialmapping.enable`	If set to `true`, then credentials for each request are injected into the request stream.
`credentialmapping.credtypes`	List of credential types to ask for in this policy domain. Only credentials that are mapped and that are supported by configured Credential Mapping provider are returned for a specific request. Therefore, asking for a credential does not guarantee that it is there. For example, to configure credential mapping to support the password for the database server, perform the following steps: • Set `credentialmapping.credtypes` to: `"credentialmapping.credtypes=DBPASSWORD"` • On the Details tab of the Database Credential Mapping provider in the Web Service SSM, set the `Allowed Types` parameter to `DBPASSWORD`. **Note:** The Database Credential Mapper provider provides identity credentials. An identity credential is the same as a `PasswordCredential` in Java. Others credentials, such SAML assertions, ALES Identity Assertions IA, and so on, are identity assertions. They are the same as a `GenericCredential` in Java. The Web Service SSM can have only one identity credential defined, but many identity assertions.
`credentialmapping.prefix`	Prefix to prepend to credential names, for example `CRED`.

Naming Authority Settings

Table 5-9 describes the settings that you use to configure the Web Server SSM naming authority.

Table 5-9 Naming Authority Settings

Setting	Description
`namingauthority.resource`	Specifies the naming authority for the resource. The naming authority is configured in the Web Service SSM.
`namingauthority.action`	Specifies the action naming authority.
`namingauthority.audit`	Specifies the audit naming authority.
webservice.registry.url	Specifies the URL of the Web Services Registry Service. For example: `http://localhost:8000/ServiceRegistry`

Logging Level Setting

Table 5-10 describes the settings that you use to configure the Web Server SSM naming authority.

Table 5-10 Logging Level Setting

Setting	Description
`log.level`	Specifies the logging level for the log4j Auditing provider.

Environment Variables Accessible Using CGI

The Web Server Security Service Module (SSM) tool kit enables you to access user environment variables using Common Gateway Interface (CGI).

Although security is embedded within the web server itself, requiring no special programming (if the user does not have access, your code will never run), a security administrator may want to use CGI to access and modify environment variables passed in by the Web Server Security Service Module. In order to customize the application according to the details of the security being enforced, a web application may access several environmental values in order to provide a more integrated user experience.

You can use CGI to access the following environment variables:

- ALES_IDENTITY — An authentication environment variable. It is available to a CGI programmer after a user successfully authenticates. This variable contains the username of the user, if available. It specifies the name of the HTTP header that will be added. The value of the variable is a list of the identity principals, including username and groups.
- ALES_DECISIONTIME — An authorization environment variable. It is available to a CGI programmer after a user is authorized to access a secure resource. It contains the date and time this authorization decision was rendered and has this format: "Monday June 23 15:14:21 EDT 2003"
- ALES_ROLES — A role environment variable that stores a list of roles calculated for the user.
- Credential Environment Variable — Table 5-12 describes the credential that is injected into the request stream when the user is authenticated. A CGI application can use this variable to access an LDAP store or database with an appropriate credential, rather than hard coding usernames and passwords. The prefix to this credential variable is configurable, although CRED is the default. Different credential types are handled differently, but the general format of the variable is: CRED_{NAME}={VALUE}

Table 5-11 Credential Environment Variables

Environment Variable	Description
Password Credentials	Password credentials conform to the format javax.resource.spi.security.PasswordCredential. The ManagedConnectionFactory element of this class is ignored. This credential type is rendered in the CGI environment as: {PREFIX}_PASSWORD={NAME}:{PASSWORD} where PREFIX is the configured prefix, NAME is the username, and PASSWORD is the password as a string. This name must match the requested credential type from credentialmapping.credtypes. For example: CRED_PASSWORD=system:weblogic

Table 5-12 Credential Environment Variables

Environment Variable	Description
Password Credentials	Password credentials conform to the format `javax.resource.spi.security.PasswordCredential`. The `ManagedConnectionFactory` element of this class is ignored. This credential type is rendered in the CGI environment as: `{PREFIX}_PASSWORD={NAME}:{PASSWORD}` where `PREFIX` is the configured prefix, `NAME` is the username, and `PASSWORD` is the password as a string. This name must match the requested credential type from `credentialmapping.credtypes`. For example: `CRED_PASSWORD=system:weblogic`

Configuring the Oracle SSM

The Oracle SSM makes use of a feature in Oracle 10g called **Fine Grained Access Control** (FGAC). FGAC allows an Oracle customer to define access policies to 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 ALES. 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.
- ALES Admin is installed on one system.
- Web Service SSM is installed and on another system, as described in "Configuring SSMs Using ConfigTool" on page 4-1. The Web Service SSM needs to be installed and configured properly before you install the Oracle SSM.

  As an alternative, you can use the `*BEA_HOME*\ales30-ssm\webservice-ssm\examples\JavaWebServiceClient` example to quickly set up a Web Service SSM, and then use the resulting SSM to configure Oracle SSM.
- The example program, the Oracle SSM, and the Web Service SSM instance must be on the same system.

- The currently logged on user must belong to the `ora_dba` group on Windows or `dba` group on Unix.

  For instance, if the currently logged on user is 'joe' then 'joe' needs to be in the `ora_dba` or `dba` group, as appropriate.

  This is required in order to connect as "system" user with "SYSDBA" role.

Steps to Create and Configure the Oracle SSM

1. Make sure all Administration Server and WebService SSM services have been started.
2. If the enrollment process has not been performed for the BEA_HOME that this SSM belongs to, then:

   a. Run the enroll tool, as described in "Enrollment" on page 3-6. You can use demo mode.

   b. Include the password for system in the encrypted password.xml by running the following in the `ales30-shared/bin` directory:

   ```
   asipassword.bat|sh system ../keys/password.xml ../keys/password.key
   ```

3. Create an Oracle SSM instance that matches what is listed in `*ALES30_SSM*/oracle-ssm/examples/OracleSSM/build.properties`.

   To do this, use `*ALES30_SSM*/oracle-ssm/adm/instancewizard.cmd|sh`.

4. Open a shell window and change directory to `*ORACLE_SSM_INSTANCE*/bin`.
5. If required, update JAVA_HOME in `*ORACLE_SSM_INSTANCE*/bin/set-env.bat|sh`.
6. Execute `*ORACLE_SSM_INSTANCE*/bin/setupOracleSSM.bat|sh` in the shell window. Substitute your 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/ales30-ssm>

        -ws_ssm_instance_dir
<c:/bea/ales30-ssm/webservice-ssm/instance/ssmws>
```

```
-db_user <ales_ora_user>

-db_password <password>

-load_example_table <true>
```

Note: If a password is not provided, the tool prompts for one. The password entry does not echo.

Default values are assigned for keys/properties when values are unspecified.

7. Open a shell window and change the directory to `ales30-ssm/oracle-ssm/examples/OracleSSM`.
8. Update `build.properties` and then execute `set-env.bat|sh`.
9. Run `ant dist config load`.
10. In ALES Administration Console, perform the following steps:
    a. Go to SSM Configuration of the Web Service SSM and click on Authentication -> FGACIdentityAsserter.
    b. On the Details tab page enter the Key value which is the value of secret property defined inside ORACLE_HOME\ssm-properties\oracle-ssm.properties.

       Click Apply.
    c. Go to Deployment. On Configuration tab page, distribute configuration.
    d. Restart Web Service SSM instance.
11. Update `ales30-ssm/oracle-ssm/examples/OracleSSM/Client.properties` to reflect your {jdbcUrl,schemaName,queryType,query} settings
12. Run `run.bat|sh` to execute client.

Sample Oracle Client Run-Result

Listing 6-1 shows a sample test result for a queryType of select, update, and delete.

Listing 6-1 Sample Oracle Client Run Result

```
C:\buildTree\ales30-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\ales30-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\ales30-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
```

Configuring the WebSphere SSM

This section describes how to configure and set up the WebSphere SSM. It also contains a simple Policy Query Web Application that shows how to retrieve basic security services, and use them to do authentication and authorization.

- Prerequisites
- Configuration Steps

Prerequisites

- ALES 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 3-6

   "Define a SCM in the ALES Database" on page 3-8

“Run asipassword” on page 3-9

2. Create the WebSphere SSM instance by running `BEA_HOME\ales30-ssm\websphere-ssm\adm\instancewizard.cmd`.

   In Windows, this can be done by opening the **Start** menu and selecting **BEA AquaLogic Enterprise Security** > **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 `*INSTANCE_HOME*/bin/set-env.bat` (or `set-env.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/ales30-ssm/websphere-ssm/examples` directory.

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

6. Copy the contents of `BEA_HOME\ales30-ssm\websphere-ssm\instance\<InstanceName>\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 into 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\ales30-ssm\websphere-ssm\examples\PolicyQueryWebApp\set-env.bat|sh`.

9. Run `BEA_HOME\ales30-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 the `/dist/PolicyQueryApp.war`.

12. Open a new 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!
    ```

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 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*\ales30-ssm\java-ssm` to `*BEA_HOME*\ales30-ssm\custom-ssm`.

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

3. Change to the `BEA_HOME\ales30-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\ales30-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\ales30-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 3-6.

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.

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 ALES 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 ALES database.

Choosing How to Run the SSM

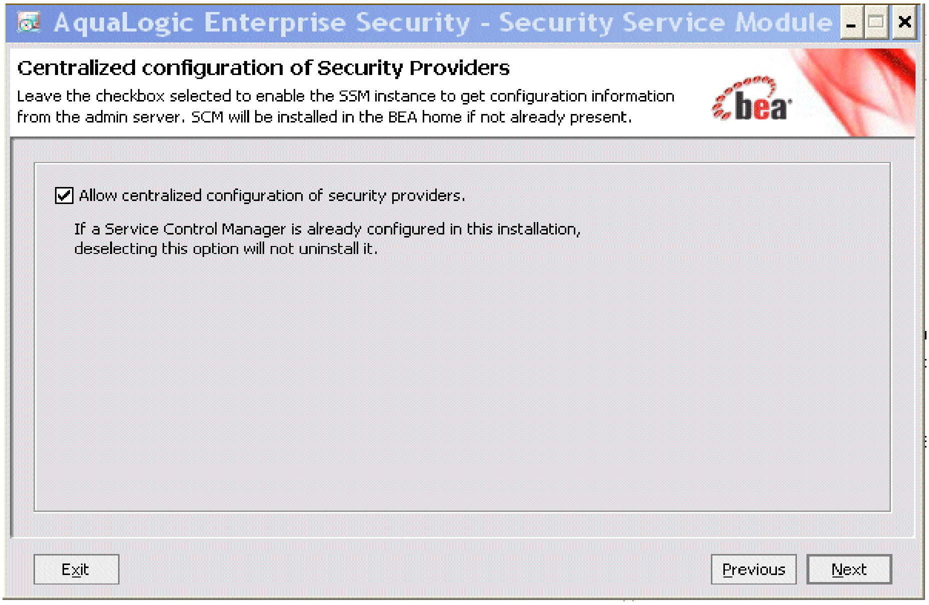
Use the following criteria when deciding whether to use an SCM or exported configuration data:

- 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


AquaLogic Enterprise Security - Security Service Module

Centralized configuration of Security Providers

Leave the checkbox selected to enable the SSM instance to get configuration information from the admin server. SCM will be installed in the BEA home if not already present.

☑ Allow centralized configuration of security providers.

If a Service Control Manager is already configured in this installation, deselecting this option will not uninstall it.

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/ales30-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/ales30-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.

Silent Mode Installations

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

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/ales30-ssm/<ssmtype>/adm/silent_install_ssm.xml`. Use the information in Table 10-2 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 in which to install the Administration Server	C:\bea
USER_INSTALL_DIR	Directory within BEA_HOME directory in which to install the SSM	C:\bea\ales30-wls-ssm
SCM_INSTALL_DIR	Directory within BEA_HOME directory in which to install the Service Control Manager. **Note:** Do not enter if not using an SCM.	C:\bea\ales30-scm
COMPONENT_PATHS	Specifies the SSMs to install, separated by the pipe (|) character. Possible component selections are: • ALES SSM COMBO/ALES SSM for Java • ALES SSM COMBO/ALES SSM for Web Service • ALES SSM COMBO/ALES SSM for IIS • ALES SSM COMBO/ALES SSM for Apache • ALES SSM COMBO/ALES SSM for WLS8.1 • ALES SSM COMBO/ALES SSM for WLS • ALES SSM COMBO/ALES SSM for Oracle • ALES SSM COMBO/ALES SSM for WebSphere	
SCM_JAVA_HOME	Java home for SCM.	d:\bea\jrockit150_06
WLS8_SSM_JAVA_HOME	Java home for the WLS 8.1 SSM.	d:\bea\jrockit150_06
WLS9_SSM_JAVA_HOME	Java home for the WLS SSM.	d:\bea\jrockit150_06
JAVA_SSM_JAVA_HOME	Java home for the Java SSM.	d:\bea\jrockit150_06
IIS_SSM_JAVA_HOME	Java home for the Web Server SSM on Microsoft IIS.	d:\bea\jrockit150_06
APACHE_SSM_JAVA_HOME	Java home for the Web Server SSM on Apache.	d:\bea\jrockit150_06
WEBSPHERE_SSM_JAVA_HOME	Java home for the Websphere SSM.	d:\bea\jrockit150_06
ORACLE_SSM_JAVA_HOME	Java home for the Oracle SSM.	d:\bea\jrockit150_06

Data Element Name	Description	Default or Sample Value
WEBSERVICE_SSM_JAVA_HOME	Java home for the Web Service SSM.	d:\bea\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

Table 10-2 Silent Installation Configuration

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

- For Windows platforms:

  ```
  ales300ssm_win32.exe -mode=silent -silent_xml=<path_to_silent.xml>
  ```

- For the Sun Solaris platform:

  ```
  ales300ssm_solaris32.bin -mode=silent -silent_xml=<path_to_silent.xml>
  ```

- For the Red Hat Advanced Server Linux platforms:

  ```
  ales300ssm_rhas_IA32.bin -mode=silent -silent_xml=<path_to_silent.xml>
  ```