Sun Java System Access Manager 7.1 2006Q4 Postinstallation Guide

Beta

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819–5899–05
July 2006

060721@15490

Contents

Figures

Tables

Examples

Preface

The *Sun Java System Access Manager 7.1 2006Q4 Postinstallation Guide* provides configuration information for Sun Java™ System Access Manager.

Access Manager is a component of the Sun Java Enterprise System (Java ES), a set of software components that provide services needed to support enterprise applications distributed across a network or Internet environment.

Who Should Use This Book

This book is intended for system administrators and system integrators who are responsible for installing and configuring Access Manager.

Before You Read This Book

Readers should be familiar with the following components and concepts:

- Access Manager technical concepts, as described in the *Sun Java System Access Manager 7.1 2006Q4 Technical Overview*.
- Deployment platform: Solaris™ or Linux operating system
- Web container that will run Access Manager: Sun Java System Application Server, Sun Java System Web Server, BEA WebLogic, or IBM WebSphere Application Server
- Technical concepts: Lightweight Directory Access Protocol (LDAP), Java technology, JavaServer Pages™ (JSP) technology, HyperText Transfer Protocol (HTTP), HyperText Markup Language (HTML), and eXtensible Markup Language (XML)

Related Books

Related documentation is available as follows:

- "Access Manager Core Documentation" on page 16
- "Sun Java Enterprise System Product Documentation" on page 16

Access Manager Core Documentation

The Access Manager core documentation set contains the following titles:

- The *Sun Java System Access Manager 7.1 2006Q4 Release Notes*will be available online after the product is released. It gathers an assortment of last-minute information, including a description of what is new in this current release, known problems and limitations, installation notes, and how to report issues with the software or the documentation.
- The *Sun Java System Access Manager 7.1 2006Q4 Technical Overview* provides an overview of how Access Manager components work together to consolidate access control functions, and to protect enterprise assets and web-based applications. It also explains basic Access Manager concepts and terminology.
- The *Sun Java System Access Manager 7.1 2006Q4 Postinstallation Guide* (this guide) provides procedural information for configuring Access Manager after you run the Java ES installer.
- The *Sun Java System Access Manager 7.1 2006Q4 Deployment Planning Guide*provides planning and deployment solutions for Access Manager based on the solution life cycle.
- The *Sun Java System Access Manager 7.1 2006Q4 Administration Guide* describes how to use the Access Manager console as well as manage user and service data via the command line interface.
- The *Sun Java System Access Manager 7.1 2006Q4 Federation and SAML Administration Guide* provides information about the Federation module based on the Liberty Alliance Project specifications. It includes information on the integrated services based on these specifications, instructions for enabling a Liberty-based environment, and summaries of the application programming interface (API) for extending the framework.
- The *Sun Java System Access Manager 7.1 2006Q4 Developer's Guide* provides information about customizing Access Manager and integrating its functionality into an organization's current technical infrastructure. It also contains details about the programmatic aspects of the product and its API.
- The *Sun Java System Access Manager 7.1 2006Q4 C API Reference* provides summaries of data types, structures, and functions that make up the public Access Manager C APIs.
- The *Sun Java System Access Manager 7.1 2006Q4 Java API Reference* provides information about the implementation of Java packages in Access Manager.
- The *Sun Java System Access Manager 7.1 2006Q4 Performance Tuning Guide* provides information about how to tune Access Manager and its related components for optimal performance.
- The *Sun Java System Access Manager Policy Agent 2.2 User's Guide* provides an overview of the policy functionality and the policy agents available for Access Manager.

Sun Java Enterprise System Product Documentation

For useful information for related products, see the following documentation collections on the Sun Java Enterprise System documentation web site (`http://docs.sun.com/prod/entsys.05q4`):

- Sun Java System Directory Server:

http://docs.sun.com/coll/1316.1

- Sun Java System Web Server:

  http://docs.sun.com/coll/1308.1

- Sun Java System Application Server:

  http://docs.sun.com/coll/1310.1

- Sun Java System Message Queue:

  http://docs.sun.com/coll/1307.1

- Sun Java System Web Proxy Server:

  http://docs.sun.com/coll/1311.1

Related Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (http://www.sun.com/documentation/)
- Support (http://www.sun.com/support/)
- Training (http://www.sun.com/training/)

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–1 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX® system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–2 Shell Prompts

Shell	Prompt
C shell	`machine_name%`
C shell for superuser	`machine_name#`
Bourne shell and Korn shell	`$`
Bourne shell and Korn shell for superuser	`#`

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions.

To share your comments, go to `http://docs.sun.com` and click Send comments. In the online form, provide the document title and part number. The part number is a seven-digit or nine-digit number that can be found on the title page of the book or at the top of the document.

For example, the title of this book is *Sun Java System Access Manager 7.1 2006Q4 Postinstallation Guide*, and the part number is 819–5899–05.

◆ ◆ ◆ CHAPTER 1

Getting Started

The *Sun Java™ System Access Manager 7.1 2006Q4 Postinstallation Guide* includes information about configuring Access Manager after installation. Usually, you perform postinstallation tasks only a few times. For example, you might want to deploy an additional instance of Access Manager instance or configure Access Manager for session failover.

For information about tasks that you perform on a regular basis, such as backing up Access Manager or directory data, see the *Sun Java System Access Manager 7.1 2006Q4 Administration Guide*.

Topics in this chapter include:

Overview of the Installation Process

For a new installation, install the first instance of Access Manager and other Sun Java Enterprise System (Java ES) components by running the Java ES installer. Information about the installer includes:

Getting the Java ES Installer

The Java ES installer is available in a media kit containing CDs or a DVD, as web download, on a pre-installed system, or from a file server on your network.

For more information, see the *Sun Java Enterprise System 2005Q4 Installation Guide for UNIX*.

Installation Modes

You can run the Java ES installer in the following modes:

- Graphical mode: An interactive wizard guides you through a series of choices on installation pages on a graphical workstation.
- Text-based mode: An interactive command-line installer prompts you for responses in a terminal window.
- Silent mode: The installer reads input from a state file, which is a text file containing name-value pairs of configuration information. You create a state file by running the installer with the `-no` and `-saveState` options. Then, you edit the state file for the specific host server where you plan to install the various Java ES components. Using a state file is useful for installing multiple instances on different host servers.

Installer Configuration Options

When you run the Java ES installer, you can select either of these configuration options for Access Manager as well as other Java ES components:

- Configure Now: You configure Access Manager and the various Java ES components when you run the installer by choosing options (or using default values). Not all Java ES components support this option.
- Configure Later: When you run the Java ES installer, you specify only minimal configuration values. Then, you later configure the specific components by running a script or using an administration console. Access Manager provides the `amconfig` script and `amsamplesilent` file for postinstallation configuration.

  If you plan to use BEA WebLogic Server or IBM WebSphere Application Server as the Access Manager web container, you must choose the Configure Later option when you install Access Manager.

For detailed information about the Java ES installer on UNIX systems, see the *Sun Java Enterprise System 2006Q4 Installation Guide for UNIX*.

Access Manager `amconfig` Script and `amsamplesilent` file

The Java ES installer installs the Access Manager `amconfig` script and silent configuration input file (`amsamplesilent`) in the following directory, depending on your platform:

- Solaris systems: *AccessManager-base* `/SUNWam/bin`
- Linux systems: *AccessManager-base*`/identity/bin`

AccessManager-base represents the Access Manager base installation directory. On Solaris systems, the default base installation directory is `/opt`, and on Linux systems, it is `/opt/sun`.

The amconfig script is a top-level script that reads configuration variables in the amsamplesilent file (or copy of the file) and then calls other scripts as needed to perform the specific Access Manager configuration.

The amsamplesilent is an ASCII text file that contains Access Manager configuration variables in the following format:

```
variable-name=value
```

For example:

```
DEPLOY_LEVEL=1
NEW_INSTANCE=true
SERVER_HOST=amhost.example.com
...
```

Before you run the amconfig script, copy (and rename, if you wish) the amsamplesilent file, and then edit the variables in the file based on your system environment and the configuration you want to perform.

For a list of the variables you can set in a configuration script input file, see the Access Manager Sample Configuration Script Input File.

The format of the amsamplesilent file does not follow the same format or necessarily use the same variable names as a Java Enterprise System silent installation state file.

Caution – Variables in the amsamplesilent file (or copy of the file) can specify sensitive data such as administrator passwords. Make sure to secure the file as appropriate for your deployment.

The amconfig script reads the configuration variables in the amsamplesilent file (or a copy of the file) to perform various operations. For more information, see Chapter 2.

◆ ◆ ◆ CHAPTER 2

Running the Access Manager `amconfig` Script

Sun Java™ System Access Manager provides the `amconfig` script and the silent configuration input file (`amsamplesilent`) to perform various postinstallation configuration operations. This chapter includes these topics:

Overview of the `amconfig` Script and `amsamplesilent` File

After you run the Java Enterprise System installer, the Access Manager `amconfig` script and silent configuration input file (`amsamplesilent`) is available in the following directory, depending on your platform:

- Solaris systems: *AccessManager-base*`/SUNWam/bin`
- Linux systems: *AccessManager-base*`/identity/bin`

AccessManager-base represents the Access Manager base installation directory. On Solaris systems, the default base installation directory is `/opt`, and on Linux systems, it is `/opt/sun`.

Use the `amconfig` script and `amsamplesilent` file (or a copy of the file) to perform these functions:

- Configure an Access Manager instance that you installed by running the Java ES installer in Configure Later mode.
- Deploy and configure additional instances of Access Manager.
- Reconfigure or redeploy an Access Manager instance.
- Deploy and configure specific Access Manager components, including:
  - Access Manager Console
  - Access Manager client SDK

- Distributed Authentication UI server
- Federation Manager
- Support for Microsoft Active Directory

- Generate an Access Manager WAR file that you can deploy on other host servers.
- Uninstall Access Manager instances and components that you deployed using the amconfig script.

Access Manager amconfig Script

The amconfig script reads the silent configuration input file (amsamplesilent or a copy) and then calls other scripts in silent mode, as needed, to perform the requested operation.

To set configuration variables, copy and rename the amsamplesilent file. Then, set the variables in the file for the operation you want to perform. For an example of a silent configuration input file, see "Example Configuration Script Input File" on page 44.

To run the amconfig script, use this syntax:

```
amconfig -s input-file
```

where:

-s runs amconfig in silent mode.

The *input-file* is the silent configuration input file that contains the configuration variables for the operation you want to perform. For more information, see "Access Manager amsamplesilent File Configuration Variables" on page 27.

Several considerations for running the amconfig script are:

- You must be running as superuser (root).
- Specify the full path to the amsamplesilent file (or copy of the file). For example:

```
# cd /opt/SUNWam/bin
# ./amconfig -s ./amsamplesilent
```

or

```
# ./amconfig -s /opt/SUNWam/bin/amsamplesilent
```

Note – In the Access Manager 7.1 2006Q4 release, the following scripts are not supported:

- `amserver` with the create argument
- `amserver.`*instance*

Also, by default `amserver start` starts only the authentication `amsecuridd` and `amunixd` helpers. The `amsecuridd` helper is available only on the Solaris OS SPARC platform.

Access Manager `amsamplesilent` File Configuration Variables

This silent configuration input file (`amsamplesilent`) contains the following configuration variables:

Deployment Mode Variable

The required `DEPLOY_LEVEL` variable determines the operation you want the `amconfig` script to perform.

TABLE 2–1 Access Manager DEPLOY_LEVEL Variable

Operation	DEPLOY_LEVEL Variable Value and Description
Install	1 = Full Access Manager installation for a new instance (default) 2 = Install Access Manager console only 3 = Install Access Manager SDK only 4 = Install SDK only and configure the container 5 = Install Federation Management module only 6 = Install server only 7 = Install Access Manager and configure the container for deploying with Portal Server **Caution** DEPLOY_MODE=7 is intended only for deploying Access Manager with Portal Server. 8 = Configure or redeploy Distributed Authentication UI server only 9 = Configure or redeploy Access Manager client SDK only 10 = Generate an Access Manager WAR file For some deployments, you might want to install the console only and server only on a single host server using different web containers. First, run the Java ES installer to install all Access Manager subcomponents using the Configure Later option. Then, run the amconfig script to configure both the console and server instances.
Uninstall (unconfigure)	11 = Full uninstall 12 = Uninstall console only 13 = Uninstall SDK only 14 = Uninstall SDK only and unconfigure the container 15 = Uninstall Federation Management module 16 = Uninstall server only 17 = Uninstall Access Manager and unconfigure the container when deployed with Portal Server. **Caution** DEPLOY_MODE=17 is intended only when Access Manager is deployed with Portal Server. 18 = Uninstall Distributed Authentication UI server only 19 = Uninstall Access Manager client SDK only

TABLE 2–1 Access Manager DEPLOY_LEVEL Variable *(Continued)*

Operation	DEPLOY_LEVEL Variable Value and Description
Re-install (also referred to as re-deploy or re-configure)	21 = Redeploy all (console, password, services, and common) web applications. 26 = Undeploy all (console, password, services, and common) web applications.

Access Manager Configuration Variables

This section describes the Access Manager configuration variables.

TABLE 2–2 Access Manager Configuration Variables

Variable	Description
AM_REALM	Indicates the Access Manager mode: ■ enabled: Access Manager operates in Realm Mode, with Access Manager 7.1 2006Q4 and features and console. ■ disabled: Access Manager operates in Legacy Mode, with Access Manager 6 2005Q1 features and console. Default: enabled **Caution –** Access Manager Realm Mode is enabled by default. If you are deploying Access Manager with Portal Server, Messaging Server, Calendar Server, Delegated Administrator, or Instant Messaging, you must select Legacy Mode (AM_REALM=disabled) before you run the amconfig script.
BASEDIR	Base installation directory for Access Manager packages. Default: PLATFORM_DEFAULT For Solaris systems, PLATFORM_DEFAULT is /opt For Linux systems, PLATFORM_DEFAULT is /opt/sun
SERVER_NAME	Name of the host on which the Access Manager server (/amserver) has been or will be deployed.
SERVER_HOST	Fully qualified host name of the system where Access Manager is running (or will be installed). For a remote SDK installation, set this variable to the host where Access Manager is (or will be) installed and not the remote client host. This variable should match the counterpart variable in the web container configuration. For example, for Application Server 8, this variable should match AS81_HOST.

TABLE 2–2 Access Manager Configuration Variables *(Continued)*

Variable	Description
SERVER_PORT	Access Manager port number. Default: 58080 For a remote SDK installation, set this variable to the port on the host where Access Manager is (or will be) installed and not the remote client host. This variable should match the counterpart variable in the web container configuration. For example, for Application Server 8, this variable should match AS81_PORT.
ADMIN_PORT	Port on which the administration instance will listen for connections. Default values are: ■ Web Server 7 2006Q4: 8989 ■ Application Server: 4849 ■ BEA WebLogic Server: 7001 ■ IBM WebSphere Application Server: 9080
SERVER_PROTOCOL	Server protocol: http or https. Default: http For a remote SDK installation, set this variable to the protocol on the host where Access Manager is (or will be) installed and not the remote client host. This variable should match the counterpart variable in the web container configuration. For example, for Application Server 8, this variable should match AS81_PROTOCOL.
CONSOLE_HOST	Fully qualified host name of the server where the console is installed. Default: Value provided for the Access Manager host
CONSOLE_PORT	Port of the web container where the console is installed and listens for connections. Default: Value provided for the Access Manager port
CONSOLE_PROTOCOL	Protocol of the web container where the console is installed. Default: Same as the server protocol
CONSOLE_REMOTE	Set to true if the console is remote from the Access Manager services. Otherwise, set to false. Default: false
DS_HOST	Fully qualified host name of Directory Server.
DS_PORT	Directory Server port. Default: 389.
DS_DIRMGRDN	Directory manager DN: the user who has unrestricted access to Directory Server. Default: "cn=Directory Manager"

TABLE 2–2 Access Manager Configuration Variables *(Continued)*

Variable	Description
DS_DIRMGRPASSWD	Password for the directory manager See the note about special characters in the description of "Access Manager Configuration Variables" on page 29.
ROOT_SUFFIX	Initial or root suffix of the directory. You must make sure that this value exists in the Directory Server you are using. See the note about special characters in the description of "Access Manager Configuration Variables" on page 29.
ADMINPASSWD	Password for the administrator (amadmin). Must be different from the password for amldapuser. **Note**: If the password contains special characters such as a slash (/) or backslash (\\), the special character must be enclosed by single quotes ('). For example: ADMINPASSWD='\\\\\\\\\\####///' However, the password cannot have a single quote as one of the actual password characters.
AMLDAPUSERPASSWD	Password for amldapuser. Must be different from the password for amadmin. See the note about special characters in the description of "Access Manager Configuration Variables" on page 29.
CONSOLE_DEPLOY_URI	URI prefix for accessing the HTML pages, classes and JAR files associated with the Access Manager Administration Console subcomponent. Default: /amconsole
SERVER_DEPLOY_URI	URI prefix for accessing the HTML pages, classes, and JAR files associated with the Identity Management and Policy Services Core subcomponent. Default: /amserver
PASSWORD_DEPLOY_URI	URI that determines the mapping that the web container running Access Manager will use between a string you specify and a corresponding deployed application. Default: /ampassword
COMMON_DEPLOY_URI	URI prefix for accessing the common domain services on the web container. Default: /amcommon
DISTAUTH_DEPLOY_URI	URI prefix for accessing content associated with the Distributed Authentication web application.
CLIENT_DEPLOY_URI	URI prefix for accessing content associated with the Client SDK.

TABLE 2–2 Access Manager Configuration Variables *(Continued)*

Variable	Description
COOKIE_DOMAIN	Names of the trusted DNS domains that Access Manager returns to a browser when it grants a session ID to a user. At least one value should be present. In general, the format is the server's domain name preceded with a period. Example: .example.com
JAVA_HOME	Path to the JDK installation directory. Default: /usr/jdk/entsys-j2se. This variable provides the JDK used by the command line interface's (such as amadmin) executables. The version must be 1.4.2 or later.
AM_ENC_PWD	Password encryption key: String that Access Manager uses to encrypt user passwords. Default: none. When the value is set to none, amconfig will generate a password encryption key for the user, so a password encryption will exist for the installation that is either specified by the user or created through amconfig . **Important**: If you are deploying multiple instances of Access Manager or the remote SDK, all instances must use the same password encryption key. When you deploy an additional instance, copy the value from the am.encryption.pwd property in the AMConfig.properties file for the first instance.
PLATFORM_LOCALE	Locale of the platform. Default: en_US (US English)
NEW_OWNER	New owner for the Access Manager files after installation. Default: root
NEW_GROUP	New group for the Access Manager files after installation. Default: other For a Linux installation, set NEW_GROUP to root.
PAM_SERVICE_NAME	Name of the PAM service from the PAM configuration or stack that comes with the operating system and is used for the Unix authentication module (normally other for Solaris or password for Linux). Default: other.
XML_ENCODING	XML encoding. Default: ISO-8859-1
NEW_INSTANCE	Specifies whether the configuration script should deploy Access Manager to a new user-created web container instance: ■ true = To deploy Access Manager to a new user-created web container instance other than an instance that already exists. ■ false = To configure the first instance or re-configure an instance. Default: false
SSL_PASSWORD	Is not used in this release.

Web Container Configuration Variables

The WEB_CONTAINER variable specifes the Access Manager web container. For the supported versions of each web container, see the Sun Java System Access Manager 7.1 2006Q4 Release Notes.

TABLE 2–3 Access Manager WEB_CONTAINER Variable

WEB_CONTAINER Value	Web Container
WS	Sun Java System Web Server 7 2006Q4
WS6	"Sun Java System Web Server 6.1 SP5" on page 34
AS8 (default)	"Sun Java System Application Server 8.1" on page 35
WL8	"BEA WebLogic Server 8.1" on page 36
WAS5	"IBM WebSphere Application Server 5.1" on page 37

Sun Java System Web Server 7 2006Q4

This section describes the configuration variables for Web Server 7 2006Q4.

TABLE 2–4 Web Server 7 2006Q4 Configuration Variables

Variable	Description
WS_INSTANCE	Name of the Web Server instance on which Access Manager will be configured or deployed. The value should correspond to a directory beneath the WS61_HOME value. Defaults: Web Server 6.1: https-$SERVER_HOST Web Server 7: https-config1
WS_HOME	Web Server instance directory. Defaults: Solaris systems: /var/opt/SUNWwbsvr7 Linux systems: /var/opt/sun/webserver7/$WS_INSTANCE
WS_PROTOCOL	Protocol (http or https) used by the Web Server instance. Default: $SERVER_PROTOCOL
WS_HOST	Fully qualified domain name on which the Web Server instance is listening for connections. Default: $SERVER_HOST If you are configuring a Distributed Authentication UI server, set WS_HOST to the same value as the DISTAUTH_HOST variable.
WS_PORT	Port on which WS_INSTANCE will listen for connections. Default: 80 (SERVER_PORT variable)

TABLE 2–4 Web Server 7 2006Q4 Configuration Variables *(Continued)*

Variable	Description
WS_ADMINPORT	Port on which the Web Server administration instance will listen for SSL connections. Default: 8989 (ADMIN_PORT variable)
WS_ADMIN	User ID of the Web Server administrator. Default: "admin"
WS_ADMINPASSWD	Password for the Web Server administrator. Default: Same value as the amadmin password (ADMINPASSWDS variable)

Sun Java System Web Server 6.1 SP5

This section describes the configuration variables for Web Server 6.1 2005Q4 SP5 in the silent configuration input file.

TABLE 2–5 Web Server 6.1 Configuration Variables

Variable	Description
WS61_INSTANCE	Name of the Web Server instance on which Access Manager will be deployed or un-deployed. Default: https-*web-server-instance-name* where *web-server-instance-name* is the Access Manager host ("Access Manager Configuration Variables" on page 29 variable)
WS61_HOME	Web Server base installation directory. Default: /opt/SUNWwbsvr
WS61_PROTOCOL	Protocol used by the Web Server instance set by the "Sun Java System Web Server 6.1 SP5" on page 34 variable where Access Manager will be deployed: http or https. Default: Access Manager protocol ("Access Manager Configuration Variables" on page 29 variable)
WS61_HOST	Fully qualified host name for the Web Server instance ("Sun Java System Web Server 6.1 SP5" on page 34 variable). Default: Access Manager host instance ("Access Manager Configuration Variables" on page 29 variable)
WS61_PORT	Port on which Web Server listens for connections. Default: Access Manager port number ("Access Manager Configuration Variables" on page 29 variable)
WS61_ADMINPORT	Port on which the Web Server Administration Server listens for connections. Default: 8888

TABLE 2–5 Web Server 6.1 Configuration Variables *(Continued)*

Variable	Description
WS61_ADMIN	User ID of the Web Server administrator. Default: "admin"

Sun Java System Application Server 8.1

This section describes the configuration variables for Application Server 8.1.

TABLE 2–6 Application Server 8.1 Configuration Variables

Variable	Description
AS81_HOME	Path to the directory where Application Server 8.1 is installed. Default: /opt/SUNWappserver/appserver
AS81_PROTOCOL	Protocol used by the Application Server instance: http or https. Default: Access Manager protocol ("Access Manager Configuration Variables" on page 29 variable)
AS81_HOST	Fully qualified domain name (FQDN) on which the Application Server instance listens for connections. Default: Access Manager host ("Access Manager Configuration Variables" on page 29 variable)
AS81_PORT	Port on which Application Server instance listens for connections. Default: Access Manager port number ("Access Manager Configuration Variables" on page 29 variable)
AS81_ADMINPORT	Port on which the Application Server administration server listens for connections. Default: 4849
AS81_ADMIN	Name of the user who administers the Application Server administration server for the domain into which Application Server is being displayed. Default: admin
AS81_ADMINPASSWD	Password for the Application Server administrator for the domain into which Application Server is being displayed. See the note about special characters in the description of "Access Manager Configuration Variables" on page 29.
AS81_INSTANCE	Name of the Application Server instance that will run Access Manager. Default: server

TABLE 2–6 Application Server 8.1 Configuration Variables *(Continued)*

Variable	Description
AS81_DOMAIN	Path to the Application Server directory for the domain to which you want to deploy this Access Manager instance. Default: domain1
AS81_INSTANCE_DIR	Path to the directory where Application Server stores files for the instance. Default: /var/opt/SUNWappserver/domains/domain1
AS81_DOCS_DIR	Directory where Application Server stores content documents. Default: /var/opt/SUNWappserver/domains/domain1/docroot
AS81_ADMIN_IS_SECURE	Specifies whether the Application Server administration instance is using SSL: ■ true: Secure port is enabled (HTTPS protocol). ■ false: Secure port is not enabled (HTTP protocol). Default: true (enabled) In ampsamplesilent, there is an additional setting that specified whether the application server administration port is secure: ■ true: The application server administration port is secure (HTTPS protocol). ■ false: The application server administration port is not secure (HTTP protocol). Default: True (enabled).

BEA WebLogic Server 8.1

This section describes the configuration variables for BEA WebLogic Server 8.1 in the silent configuration input file.

TABLE 2–7 BEA WebLogic Server 8.1 Configuration Variables

Variable	Description
WL8_HOME	WebLogic home directory. Default: /usr/local/bea
WL8_PROJECT_DIR	WebLogic project directory. Default: user_projects
WL8_DOMAIN	WebLogic domain name. Default: mydomain
WL8_CONFIG_LOCATION	Parent directory of the location of the WebLogic start script.
WL8_SERVER	WebLogic server name. Default: myserver
WL8_INSTANCE	WebLogic instance name. Default: /usr/local/bea/weblogic81 ($WL8_HOME/weblogic81)

TABLE 2–7 BEA WebLogic Server 8.1 Configuration Variables *(Continued)*

Variable	Description
WL8_PROTOCOL	WebLogic protocol. Default: http
WL8_HOST	WebLogic host name. Default: Host name of the server
WL8_PORT	WebLogic port. Default: 7001
WL8_SSLPORT	WebLogic SSL port. Default: 7002
WL8_ADMIN	WebLogic administrator. Default: "weblogic"
WL8_PASSWORD	WebLogic administrator password. See the note about special characters in the description of "Access Manager Configuration Variables" on page 29.
WL8_JDK_HOME	WebLogic JDK home directory. Default: "BEA WebLogic Server 8.1" on page 36 /jdk142_04

IBM WebSphere Application Server 5.1

This section describes the configuration variables for IBM WebSphere Application Server 5.1 in the silent configuration input file.

TABLE 2–8 IBM WebSphere Application Server 5.1 Configuration Variables

Variable	Description
WAS51_HOME	WebSphere home directory. Default: /opt/WebSphere/AppServer
WAS51_JDK_HOME	WebSphere JDK home directory. Default: /opt/WebSphere/AppServer/java
WAS51_CELL	WebSphere cell. Default: host-name value
WAS51_NODE	WebSphere node name. Default: host name of the server where WebSphere is installed. Default: hostname value
WAS51_INSTANCE	WebSphere instance name. Default: server1
WAS51_PROTOCOL	WebSphere protocol. Default: http
WAS51_HOST	WebSphere host name. Default: Hostname of the server
WAS51_PORT	WebSphere port. Default: 9080
WAS51_SSLPORT	WebSphere SSL port. Default: 9081
WAS51_ADMIN	WebSphere administrator. Default: "admin"
WAS51_ADMINPORT	WebSphere administrator port. Default: 9090

Directory Server Configuration Variables

For the versions of Directory Server supported by Access Manager 7.1 2006Q4, see the Sun Java System Access Manager 7.1 2006Q4 Release Notes. This section describes the Directory Server configuration variables.

TABLE 2–9 Directory Server Configuration Variables

Variable	Description
DIRECTORY_MODE	Directory Server modes: 1 = Use for a new installation of a Directory Information Tree (DIT). 2 = Use for an existing DIT. The naming attributes and object classes are the same, so the configuration scripts load the installExisting.ldif and umsExisting.ldif files. The configuration scripts also update the LDIF and properties files with the actual values entered during configuration (for example, BASE_DIR, SERVER_HOST, and ROOT_SUFFIX). This update is also referred to as "tag swapping," because the configuration scripts replace the placeholder tags in the files with the actual configuration values. 3 = Use for an existing DIT when you want to do a manual load. The naming attributes and object classes are different, so the configuration scripts do not load the installExisting.ldif and umsExisting.ldif files. The scripts perform tag swapping (described for mode 2). You should inspect and modify (if needed) the LDIF files and then manually load the LDIF files and services. 4 = Use for an existing multi-server installation. The configuration scripts do not load the LDIF files and services, because the operation is against an existing Access Manager installation. The scripts perform tag swapping only (described for mode 2) and adds a server entry in the platform list. 5 = Use for an existing upgrade. The scripts perform tag swapping only (described for mode 2). Default: 1
USER_NAMING_ATTR	User naming attribute: Unique identifier for the user or resource within its relative name space. Default: uid
ORG_NAMING_ATTR	Naming attribute of the user's company or organization. Default: o
ORG_OBJECT_CLASS	Organization object class. Default: sunismanagedorganization
USER_OBJECT_CLASS	User object class. Default: inetorgperson
DEFAULT_ORGANIZATION	Default organization name. Default: none

Distributed Authentication UI Server Configuration Variables

This section describes the Distributed Authentication UI server configuration variables in the silent configuration input file.

TABLE 2–10 Distributed Authentication UI Server Configuration Variables

Variable	Description
DISTAUTH_PROTOCOL	Protocol (http or https) used by the web container instance on which the Distributed Authentication UI server has been or will be deployed. Default: http
DISTAUTH_HOST	Fully qualified host name where the Distributed Authentication UI server is located. Default: distAuth_sample.com
DISTAUTH_PORT	Port on DISTAUTH_HOST on which the Distributed Authentication UI server has been or will be deployed. Default: 80
APPLICATION_USER	User name for the application. Default: username
APPLICATION_PASSWD	Password of the user for the application. Default: none
AM_ENC_SECRET	Password encryption secret key from the server. Default: none
AM_ENC_LOCAL	Password encryption key. Default: none
DEBUG_LEVEL	Level for the debug service. Values can be: error, warning, or message. Default: error
DEBUG_DIR	Directory where the debug files will be created. Default: Solaris systems: /var/opt/SUNWam/logs Linux systems: /var/opt/sun/identity/logs

Microsoft Active Directory Configuration Variables

The following table describes the Microsoft Active Directory configuration variables.

TABLE 2–11 Microsoft Active Directory Configuration Variables

Variable	Description
CONFIG_AD	Specifies true or false whether Active Directory will be used as the configuration data store. Default: false
CONFIG_SERVER	Fully qualified host name of the Active Director server. Default: Value of the DS_HOST variable

TABLE 2–11 Microsoft Active Directory Configuration Variables *(Continued)*

Variable	Description
CONFIG_PORT	Port of the Active Director server. Default: Value of the DS_PORT variable
CONFIG_ADMINDN	Administrator DN used to bind to Active Directory. Default: "cn=dsameuser,ou=DSAME Users"
CONFIG_ADMINPASSWD	Password for the Administrator DN (CONFIG_AMADMIN). Default: Value of the ADMINPASSWD variable

Access Manager Deployment Scenarios

After you have installed the first instance of Access Manager using the Java Enterprise System installer, you can deploy and configure additional Access Manager instances by editing the configuration variables in the silent configuration input file and then running the amconfig script.

This section describes the following scenarios:

- "Deploying Additional Instances of Access Manager" on page 40
- "Configuring and Reconfiguring an Instance of Access Manager" on page 42
- "Uninstalling Access Manager" on page 43
- "Uninstalling All Access Manager Instances" on page 43

Deploying Additional Instances of Access Manager

Before you can deploy a new instance of Access Manager, you must create and start the new web container instance using the administration tools for the web container. For information, refer to the specific web container documentation:

- For Web Server, see http://docs.sun.com/coll/1308.1
- For Application Server, see http://docs.sun.com/coll/1310.1

The steps described in this section only apply to an Access Manager instance that has been installed with the Configure Now option. If you are planning to use WebLogic or WebSphere as web containers, you must use the Configure Later option when installing Access Manager. See Chapter 2, Installing and Configuring Third-Party Web Containers for more information.

Deploying an Additional Access Manager Instance

This section describes how to deploy an additional Access Manager instance on a different host server and update the Platform Server List.

To Deploy an Additional Access Manager Instance

1. **Log in as an administrator, depending on the web container for the instance. For example, if Web Server 6.1 will be the web container for the new instance, log in either as superuser (root) or as the user account for the Web Server Administration Server.**

2. **Copy the `amsamplesilent` file to a writable directory and make that directory your current directory. For example, you might create a directory named `/newinstances`.**

   Tip Rename the copy of the `amsamplesilent` file to describe the new instance you want to deploy. For example, the following steps use an input file named `amnewws6instance` to install a new instance for Web Server 6.1.

3. **Set the following variables in the new `amnewws6instance` file:**

   ```
   DEPLOY_LEVEL=1
    NEW_INSTANCE=true
   ```

   Set other variables in the `amnewws6instance` file as required for the new instance you want to create. For a description of these variables, refer to the tables in the following sections:

   - "Access Manager Configuration Variables" on page 29
     - "Web Container Configuration Variables" on page 33
     - "Directory Server Configuration Variables" on page 38

       Important All Access Manager instances must use the same value for the password encryption key. To set the AM_ENC_PWD variable for this instance, copy the value from the `am.encryption.pwd` property in the `AMConfig.properties` file for the first instance.

       In case you might need to uninstall this instance later, save the `amnewws6instance` file.

4. **Run the `amconfig`, specifying the new `amnewws6instance` file. For example, on Solaris systems:**

   ```
   # cd opt/SUNWam/bin/
    # ./amconfig -s ./newinstances/amnewws6instance
   ```

   The `-s` option runs the `amconfig` script in silent mode.

   The `amconfig` script calls other configuration scripts as needed, using variables in the `amnewws6instance` file to deploy the new instance.

To Update the Platform Server List

When you crate an additional container instance, you must update the Access Manager Platform Server list to reflect the addition of the container(s).

1. **Log in to the Access Manager Console as the top-level administrator.**

2. **Click on the Service Configuration tab.**

3. **Click on the Platform service.**

4 **Enter the following information for the new instance in the Server List:**

protocol://*fqdn*:*port*|*instance-number*

The instance number should be the next available number that is not in use.

5 **Click Add.**

6 **Click Save.**

Configuring and Reconfiguring an Instance of Access Manager

You can configure an instance of Access Manager that was installed with the Configure Later option or reconfigure the first instance that was installed using Configure Now option in the Java Enterprise System installer by running the `amconfig` script.

For example, you might want to reconfigure an instance to change the Access Manager owner and group.

▼ To Configure or Reconfigure an Instance of Access Manager

1 **Log in as an administrator, depending on the web container for the instance. For example, if Web Server 6.1 is the web container, log in either as superuser (root) or as the user account for Web Server Administration Server.**

2 **Copy the silent configuration input file you used to deploy the instance to a writable directory and make that directory your current directory. For example, to reconfigure an instance for Web Server 6.1, the following steps use an input file named `amnewinstanceforWS61` in the `/reconfig` directory.**

3 **In the `amnewinstanceforWS61` file, set the DEPLOY_LEVEL variable to one of the values described for a "Deployment Mode Variable" on page 27 operation. For example, set DEPLOY_LEVEL=21 to reconfigure a full installation.**

4 **In the `amnewinstanceforWS61` file, set the NEW_INSTANCE variable to false:**

```
NEW_INSTANCE=false
```

5 **Set other variables in the `amnewinstanceforWS61` file to reconfigure the instance. For example, to change the owner and group for the instance, set the NEW_OWNER and NEW_GROUP variables to their new values.**

For a description of other variables, refer to the tables in the following sections:

- "Access Manager Configuration Variables" on page 29
  - "Web Container Configuration Variables" on page 33
  - "Directory Server Configuration Variables" on page 38

6 **Run the `amconfig` script, specifying your edited input file. For example, on Solaris systems:**

```
# cd opt/SUNWam/bin/
 # ./amconfig -s ./reconfig/amnewinstanceforWS61
```

The `-s` option runs the script in silent mode. The `amconfig` script calls other configuration scripts as needed, using variables in the `amnewinstanceforWS61` file to reconfigure the instance.

Uninstalling Access Manager

You can uninstall an instance of Access Manager that was installed by running the `amconfig` script. You can also temporarily unconfigure an instance of Access Manager, and unless you remove the web container instance, it is still available for you to re-deploy another Access Manager instance later.

▼ To Uninstall an Instance of Access Manager

1 **Log in as an administrator, depending on the web container for the instance. For example, if Web Server 6.1 is the web container, log in either as superuser (root) or as the user account for Web Server Administration Server.**

2 **Copy the silent configuration input file you used to deploy the instance to a writable directory and make that directory your current directory. For example, to unconfigure an instance for Web Server 6.1, the following steps use an input file named `amnewinstanceforWS61` in the `/unconfigure` directory.**

3 **In the `amnewinstanceforWS61` file, set the DEPLOY_LEVEL variable to one of the values described for an "Deployment Mode Variable" on page 27 operation. For example, set DEPLOY_LEVEL=11 to uninstall (or unconfigure) a full installation.**

4 **Run the `amconfig` script, specifying your edited input file. For example, on Solaris systems:**

```
# cd opt/SUNWam/bin/
 # ./amconfig -s ./unconfigure/aminstanceforWS61
```

The `-s` option runs the script in silent mode. The `amconfig` script reads the `amnewinstanceforWS61` file and then uninstalls the instance.

The web container instance is still available if you want to use it to re-deploy another Access Manager instance later.

Uninstalling All Access Manager Instances

This scenario completely removes all Access Manager instances and packages from a system.

▼ To Completely Remove Access Manager From a System

1 **Log in as or become superuser (root).**

2 **In the input file you used to deploy the instance, set the DEPLOY_LEVEL variable to one of the values described for an "Deployment Mode Variable" on page 27 operation. For example, set DEPLOY_LEVEL=11 to uninstall (or unconfigure) a full installation.**

3 **Run the `amconfig` script using the file you edited in "Uninstalling All Access Manager Instances" on page 43. For example on Solaris systems:**

```
# cd opt/SUNWam/bin/
# ./amconfig -s ./newinstances/amnewws6instance
```

The `amconfig` script runs in silent mode to uninstall the instance.

Repeat these steps for any other Access Manager instances you want to uninstall, except for the first instance, which is the instance you installed using the Java Enterprise System installer.

4 **To uninstall the first instance and remove all Access Manager packages from the system, run the Java Enterprise System uninstaller. For information about the uninstaller, refer to the** *Sun Java Enterprise System 2006Q4 Installation Guide for UNIX***.**

Example Configuration Script Input File

The following section includes an example of an Access Manager configuration script input file for deployment with WebLogic 8.1.

```
DEPLOY_LEVEL=1
BASEDIR=/opt
SERVER_HOST=ide-56.example.company.com
SERVER_PORT=7001
SERVER_PROTOCOL=http
CONSOLE_HOST=$SERVER_HOST
CONSOLE_PORT=$SERVER_PORT
CONSOLE_PROTOCOL=$SERVER_PROTOCOL
CONSOLE_REMOTE=false
DS_HOST=ide-56.example.company.com
DS_PORT=389
DS_DIRMGRDN=”cn=Directory Manager”
DS_DIRMGRPASSWD=11111111
ROOT_SUFFIX=”dc=company,dc=com”
ADMINPASSWD=11111111
AMLDAPUSERPASSWD=00000000
CONSOLE_DEPLOY_URI=/amconsole
SERVER_DEPLOY_URI=/amserver
PASSWORD_DEPLOY_URI=/ampassword
COMMON_DEPLOY_URI=/amcommon
COOKIE_DOMAIN=.iplanet.com
JAVA_HOME=/usr/jdk/entsys-j2se
AM_ENC_PWD=””
```

```
PLATFORM_LOCALE=en_US
NEW_OWNER=root
NEW_GROUP=other
XML_ENCODING=ISO-8859-1
NEW_INSTANCE=false
WEB_CONTAINER=WL8
WL8_HOME=/export/bea8
WL8_PROJECT_DIR=user_projects
WL8_DOMAIN=mydomain
WL8_CONFIG_LOCATION=$WL8_HOME/$WL8_PROJECT_DIR/domains
WL8_SERVER=myserver
WL8_INSTANCE=/export/bea8/weblogic81
WL8_PROTOCOL=http
WL8_HOST=ide-56.example.company.com
WL8_PORT=7001
WL8_SSLPORT=7002
WL8_ADMIN=”weblogic”
WL8_PASSWORD=”11111111”
WL8_JDK_HOME=$WL8_HOME/jdk142_04
DIRECTORY_MODE=1
USER_NAMING_ATTR=uid
ORG_NAMING_ATTR=o
ORG_OBJECT_CLASS=examplemanagedorganization
USER_OBJECT_CLASS=inetorgperson
DEFAULT_ORGANIZATION=
Sample Configuration Script Input File for WebLogic 8.1.x
```

◆ ◆ ◆ CHAPTER 3

Deploying Multiple Access Manager Instances

Deploying multiple Access Manager instances on different host servers, with each instance accessing the same Directory Server, includes these steps:

Running the Java Enterprise System (Java ES) Installer

Install the first Access Manager instance on a host server by running the Java ES installer. Considerations for running the installer include:

- When you run the installer, you can also install other Java ES components such as Directory Server, Message Queue, and either Web Server or Application Server as the Access Manager web container.
- After installation, the `amconfig` script and the `amsamplesilent` configuration file are available in the following directory, depending on your platform:
  - Solaris systems: *AccessManager-base* `/SUNWam/bin`
  - Linux systems: *AccessManager-base*`/identity/bin`

  Where: *AccessManager-base* represents the Access Manager base installation directory. On Solaris systems, the default base installation directory is `/opt`, and on Linux systems, it is `/opt/sun`.
- When you run the installer, specify either the Configure Now or Configure Later option.
  - Configure Now: You configure Access Manager and the various Java ES components when you run the installer by choosing options (or default values). Not all Java ES components support this option.
  - Configure Later: When you run the Java ES installer, you specify only minimal configuration values. Then, you later configure the specific components by running a script or using an administration console. Access Manager provides the `amconfig` script and `amsamplesilent` file for postinstallation configuration.

- If you want to use an existing Directory Server that already contains user data, check ”Yes” for “Is Directory Server provisioned with user data?”.
- To use BEA WebLogic Server or IBM WebSphere Application Server as the web container, you must choose the Configure Later option when you install Access Manager, as follows:
  1. Install BEA WebLogic Server or IBM WebSphere Application Server by following the respective BEA or IBM product documentation.
  2. Install Access Manager by running the installer with the Configure Later option.
  3. Configure Access Manager for the web container by setting variables in the amsamplesilent configuration file (or a copy of the file) and then running the amconfig script.

For information about running the installer, see the Sun Java Enterprise System 2006Q4 Installation Guide for UNIX.

Configuring Access Manager Using the amconfig Script

To configure or re-configure an Access Manager instance, set variables in the amsamplesilent file (or a copy of the file) and run the amconfig script.

▼ To Configure Access Manager Using the amconfig Script

1 Login as (or become) superuser (root).

2 Copy and edit the amsamplesilent file.

a. Copy the amsamplesilent file to a writable directory and make that directory your current directory. For example, you might create a directory named /newinstances.

b. Rename the copy of the amsamplesilent file to describe the new instance you want to configure. For example, if you plan to create a new Access Manager instance for Web Server 6.1, you might rename the file to amwebsvr6.

c. Set the variables in the amwebsvr6 file to configure or reconfigure the new instance. For example, to configure Access Manager in Realm mode:

```
AM_REALM=enabled
DEPLOY_LEVEL=1
NEW_INSTANCE=false
WEB_CONTAINER=WS6 # Web Server is the web container
DIRECTORY_MODE=4  # Directory Server is provisioned with user data
AM_ENC_PW=password-encryption-key-value-from-the-first-Access-Manager-instance
...
```

Considerations for setting variables in the amsamplesilent file:

- If you are using non-default naming attributes and object classes, specify the custom values as appropriate for the user naming and organization naming attributes and object classes. Also, all deploy URIs (SERVER_DEPLOY_URI, CONSOLE_DEPLOY_URI, PASSWORD_DEPLOY_URI, and COMMON_DEPLOY_URI) for the web applications must match the previous installation.
- Use the same password encryption key as the first instance, as described in following Caution.

Caution – In a multiple server deployment that shares the same Directory Server, all Access Manager instances must use the same value for the password encryption key.

If you run the Java ES installer to install Access Manager on subsequent (second, third, and so on) servers in a multiple server deployment, the installer generates a new random password encryption key for each server. Therefore, when you run the installer on a subsequent server, use the encryption key value from the first Access Manager instance, which you can copy from the am.encryption.pwd attribute in the AMConfig.properties file and set as follows:

- Configure Now option. Replace the new random encryption key generated by the installer with the encryption key value from the first instance.
- Configure Later option. Set the AM_ENC_PWD variable in the copy of the amsamplesilent file with the encryption key value from the first instance before you run the amconfig script.

However, if you need to change the password encryption key for an Access Manager instance, see Changing the Password Encryption Key in the *Access Manager Deployment Planning Guide*.

3 Run the amconfig script.

For example, on Solaris systems with Access Manger installed in the default directory, run amconfig using the new amwebsvr6 file as the configuration input file:

```
# cd /opt/SUNWam/bin/
# ./amconfig -s ./newinstances/amwebsvr6
```

Specify the full path to the amsamplesilent file (or copy of the file).

The amconfigscript reads the variables in the amwebsvr6 file and then runs in silent mode (-s option) to configure Access manager for the web container.

For more information about the amsamplesilent file and running the amconfig script, see Chapter 2.

4 In case you might need to reconfigure or uninstall this instance later, save the new amwebsvr6 file.

Adding Additional Instances to the Platform Server List and Realm/DNS Aliases

When you install multiple instances of Access Manager on different host servers, the additional instances are not added to the platform server list or the realm/DNS aliases. You must explicitly add the values for the additional Access Manager instances.

▼ To Add Additional Instances to the Platform Server List and Realm/DNS Aliases

1. **Log in to the Access Manager 7.1 2006Q4 Console as** `amadmin` **on the first Access Manager host server.**
2. **In the Access Manager Console, click** `Configuration`, `System` `Properties`, **and then** `Platform`.
3. **Add each additional Access Manager instance to the Platform Server List under** `Instance` `Name`:
   a. **In the Platform Server List under** `Instance` `Name` **Name, click** `New`.
   b. **In** `New` `Server` `Instance`, **add the Server and Instance Name. For example:**
      - Server: `http://amserver2.example.com:80`
      - Instance Name: `02`
   c. **Click** `OK` **to add the instance.**
   d. **After you have added all instances, click** `Save`.
4. **Add the Realm/DNS alias for each additional Access Manager instance:**
   a. **In the Access Manager Console, click Access Control and then the root (top-level) realm under Realm Name.**
   b. **Under Realm Attributes, add the Access Manager instance to Realm/DNS Aliases and then click Add. For example:** `amserver2.example.com`
   c. **After you have added all instances, click** `Save`.

◆ ◆ ◆ CHAPTER 4

Configuring Access Manager to Run as a Non-root User

In a typical deployment, Sun Java™ System Access Manager runs as superuser (`root`). In some situations, however, you might want Access Manager to run as a non-root user. This chapter describes how to install and configure Access Manager to run as a non-root user with either Sun Java System Web Server or Sun Java System Application Server Enterprise Edition (EE) as the web container:

Requirements

This chapter is intended for system administrators and software technicians who are deploying Access Manager and other Sun Java Enterprise System (Java ES) components. You should be familiar with the administrative commands for your deployment platform (Solaris™ system or Linux system) and the following tasks.

Task	Where to Find More Information
Understanding Access Manager technical concepts	Sun Java System Access Manager 7 2006Q4 Technical Overview

Task	Where to Find More Information

Running the Java ES installer to install Java ES components, including:
■ Sun Java System Access Manager
■ Sun Java System Directory Server
■ Sun Java System Message Queue
■ Access Manager web container:
  ■ Sun Java System Web Server
  ■ Sun Java System Application Server
	Sun Java Enterprise System 2006Q4 Installation Guide for UNIX

Applying any required patches for Access Manager and other Java ES components.	Check for required patches in the Java ES 2006Q4 Release Notes Collection:
	`http://docs.sun.com/coll/1315.1`
	For some components, you might need to check with your Sun Microsystems technical representative.
	You can download patches from SunSolve Online:
	`http://sunsolve.sun.com/`

Running the Access Manager `amconfig` script to deploy and configure Access Manager instances.	Chapter 2

Administering Java ES components, including starting and stopping Directory Server and the web container (Web Server or Application Server)	Java ES collection:
	`http://docs.sun.com/prod/entsys.05q4`
	Java ES component documentation:
	■ Directory Server:
	`http://docs.sun.com/coll/1316.1`
	■ Web Server: `http://docs.sun.com/coll/1308.1`
	■ Application Server:
	`http://docs.sun.com/coll/1310.1`
	■ Message Queue:
	`http://docs.sun.com/coll/1307.1`

Running Access Manager as a Non-root User With Web Server

The following procedure allows you to install and configure Access Manager with Web Server as the web container and then to run the components as a non-root user.

▼ To Install and Configure Access Manager with Web Server as the Web Container

1 As superuser (`root`), create a non-root user and group, if they do not already exist. Examples in this chapter use `amuser` and `amgroup` as the non-root user and group. For example, on Solaris 10 systems:

```
# groupadd amgroup
# mkdir /export/home
# useradd -d /export/home/amuser -m -g amgroup amuser
```

2 As superuser (`root`), install Directory Server and Administration Server by running the Java ES installer. Specific values that you must set are:

- On the Common Server Settings page, enter the non-root user (`amuser`) for System User and non-root group (`amgroup`) for System Group.
- Select port numbers for Directory Server and Administration Server that are greater than 1024. Do not use port number 389 or 390.

3 As the non-root user, start Administration Server and Directory Server. For example:

```
/javaes/ds/start-admin
...
/javaes/ds/slapd-host.example.com/start-slapd
```

All processes should be owned by the non-root user (`amuser` in `amgroup`). For example:

```
amuser 2474 1 0 01:32:08 ? 0:00 ./uxwdog -e -d /javaes/ds/admin-serv/config
amuser 2485 1 0 01:32:16 ? 0:01 ./ns-slapd -D /javaes/ds/slapd-host
  -i /javaes/ds/slapd-host/lo
amuser 2475 2474 0 01:32:08 ? 0:00 ns-httpd -d /javaes/ds/admin-serv/config
amuser 2477 2475 0 01:32:08 ? 0:01 ns-httpd -d /javaes/ds/admin-serv/config
```

4 As superuser (`root`), install Web Server 6.1 by running the Java ES installer. Specific values that you must set are:

- On the Common Server Settings page, enter the non-root user for System User and non-root group for System Group.
- On the Web Server: Administration (1 of 2) page, change the Administration Runtime User ID to the non-root user.
- On the Web Server: Default Web Server Instance (2 of 2) page, change the Runtime User ID to the non-root user and the Runtime Group to the non-root group. Specify a value for HTTP Port that is greater than 1024.

5 As the non-root user, start the Web Server administration instance and Web Server instance. All processes should be owned by the non-root user (`amuser` in `amgroup`). For example:

```
amuser 4200 1 0 02:00:44 ? 0:00 ./webservd-wdog -r
  /javaes/ws -d /javaes/ws/https-admserv/config -n https
```

```
amuser 2474 1 0 01:32:08 ? 0:00 ./uxwdog -e -d
  /javaes/ds/admin-serv/config
amuser 4202 4201 1 02:00:44 ? 0:02 webservd -r
  /javaes/ws -d /javaes/ws/https-admserv/config -n https-admser
amuser 4220 4219 1 02:00:54 ? 0:03 webservd -r
  /javaes/ws -d /javaes/ws/https-amhost.example.com/conf
amuser 4219 4218 0 02:00:54 ? 0:00 webservd -r
  /javaes/ws -d /javaes/ws/https-amhost.example.com/conf
amuser 4201 4200 0 02:00:44 ? 0:00 webservd -r
  /javaes/ws -d /javaes/ws/https-admserv/config -n https-admser
```

6 **As superuser (`root`), install Access Manager by running the Java ES installer. On the Configuration Type page, select the Configure Later option.**

7 **Depending on your platform, change the ownership of the following directories from `root` and `other` to the non-root user and non-root group:**

- Solaris systems: `/opt/SUNWma` and `/etc/opt/SUNWma`
- Linux systems: `/opt/sun/mobileaccess` and `/etc/opt/sun/mobileaccess`

For example, on Solaris systems:

```
# chown -R amuser:amgroup /opt/SUNWma /etc/opt/SUNWma
```

8 **As superuser (`root`), change to the Access Manager `/bin` directory, depending on your platform. For example:**

- Solaris systems: `cd /opt/SUNWam/bin`
- Linux systems: `cd /opt/sun/identity/bin`

9 **As superuser (`root`), make a copy of the `amsamplesilent` file. For example:**

```
# cp -p amsamplesilent am.non_root_install
```

10 **As superuser (`root`), edit the `am.non_root_install` file as follows:**

- Set `BASEDIR` to the same value that you selected for the Access Manager installation directory when you ran the Java ES installer.
- Set `NEW_OWNER` to the non-root user and `NEW_GROUP` to the non-root group.
- Update the following variables: `SERVER_HOST`, `SERVER_PORT`, `DS_HOST`, `DS_PORT`, `ROOT_SUFFIX`, `COOKIE_DOMAIN`, `WS61_ADMINPORT` and all related password fields, including `DS_DIRMGRPASSWD`, `ADMINPASSWD`, and `AMLDAPUSERPASSWD`.

11 **As superuser (`root`), run the `amconfig` script with the edited `am.non_root_install` file to deploy Access Manager. For example:**

```
# ./amconfig -s ./am.non_root_install
```

12 **As the non-root user, stop the Web Server Administration Server instance and Web Server instance.**

13 **As superuser (`root`), change the ownership of the Web Server installation directory to the non-root user and group. For example:**

```
# chown -R amuser:amgroup /opt/SUNWwbsvr
```

14 **As the non-root user, start the Web Server Administration Server instance and the Web Server instance.**

15 **Access the Web Server Administration Console in a browser and login as the Web Server administrator.**

16 **Select the instance on which you deployed Access Manager and click Manage.**

17 **Click Apply and then Apply Changes.**

Running Access Manager as a Non-root User With Application Server

The following procedure allows you to install and configure Access Manager with Application Server as the web container and then to run the components as a non-root user.

▼ To Install and Configure Access Manager with Application Server as the Web Container

1 **As superuser (`root`), create a non-root user and group, if they do not already exist. Examples in this chapter use `amuser` and `amgroup` as the non-root user and group. For example, on Solaris 10 systems:**

```
# groupadd amgroup
# mkdir /export/home
# useradd -d /export/home/amuser -m -g amgroup amuser
```

2 **As superuser (`root`), install Directory Server and Administration Server by running the Java ES installer. Specific values that you must set are:**

- On the Common Server Settings page, enter the non-root user (`amuser`) for System User and non-root group (`amgroup`) for System Group.
- Select port numbers for Directory Server and Administration Server that are greater than 1024. Do not use port number 389 or 390.

3 **As the non-root user, start Directory Server and Administration Server. For example:**

```
/javaes/ds/start-admin
...
/javaes/ds/slapd-host.example.com/start-slapd
```

All processes should be owned by the non-root user (`amuser` in `amgroup`). For example:

```
amuser 2474 1 0 01:32:08 ? 0:00 ./uxwdog -e -d
  /javaes/ds/admin-serv/config
amuser 2485 1 0 01:32:16 ? 0:01 ./ns-slapd -D /javaes/ds/slapd-host -i
  /javaes/ds/slapd-host/lo
amuser 2475 2474 0 01:32:08 ? 0:00 ns-httpd -d
  /javaes/ds/admin-serv/config
amuser 2477 2475 0 01:32:08 ? 0:01 ns-httpd -d
  /javaes/ds/admin-serv/config
```

4 As superuser (`root`), install Application Server 8.1 and Message Queue by running the Java ES installer. Specific values that you must set are:

- On the Installation Directories page, for the Application Server and Application Server Data and Configuration directories, enter values that are beneath the non-root user's home directory. For example, if the non-root user's home directory is `/export/home/amuser`, the Application Server installation directory could be `/export/home/amuser/as`.
- On the Common Server Settings page, enter the non-root user for System User and non-root group for System Group.
- On the Application Server Domain Administration Server (1 of 1) page, select port numbers that are greater than 1024 for the Application Server Administration Port, JMX Port, HTTP Port, and HTTPS Port.

5 As superuser (`root`), delete the Application Server domain created by the Java ES installer in the following location, depending on your platform:

- Solaris systems: `/export/home/amuser/as/appserver/bin`
- Linux systems: `/export/home/amuser/as/bin`

For example, to delete the Application Server domain:

```
#./asadmin delete-domain --domaindir /asdomains domain1
```

6 As superuser (`root`), change the ownership of the Application Server installation directory and the Application Server data and configuration directory to the non-root user and group. For example:

```
# chown -R amuser:amgroup /export/home/amuser/as /export/home/amuser/as_var/
```

7 As superuser (`root`), create an administration password file as follows:

```
# echo "AS_ADMIN_PASSWORD=application-server-admin-password" > /tmp/asAdminPassFile
```

8 Recreate the Application Server domain as the non-root user:

a. Change to the non-root user. For example:

```
# su - amuser
```

b. Change to the `/bin` directory. For example, on Solaris systems:

```
cd /export/home/amuser/as/appserver/bin
```

Or, on Linux systems:

```
cd /export/home/amuser/as/bin
```

c. Invoke the `asadmin create-domain` command to recreate the deleted domain. You will be prompted to enter and confirm the domain's administration password and the master password. For example:

```
./asadmin create-domain --domaindir /export/home/amuser/as_var/domains
--adminport 4849 --adminuser admin --passwordfile /tmp/asAdminPassFile
--instanceport 8080 --domainproperties domain.jmxPort=8686:http.ssl.port=8181
--savemasterpassword=true domain1
Please enter adminpassword> adminpassword
Please enter adminpassword again> adminpassword
Please enter the master password> masterpassword
Please enter the master password again> masterpassword
Using default port 7,676 for JMS.
Using default port 3,700 for IIOP.
Using default port 3,820 for IIOP_SSL.
Using default port 3,920 for IIOP_MUTUALAUTH.
Domain domain1 created.
```

9 As superuser (`root`), remove the Application Server administration password file. For example:

```
# rm -rf /tmp/asAdminPassFile
```

10 As the non-root user, use the `asadmin start-domain` command to start the Application Server domain that you just created. You will be prompted for the administration password. For example:

```
./asadmin start-domain --user admin domain1
```

The Application Server and Message Queue processes should be owned by the non-root user (`amuser` in `amgroup`). For example:

```
amuser 15009 15007 0 12:26:20 pts/4 0:00 /bin/sh
  /usr/bin/imqbrokerd -javahome /usr/jdk/entsys-j2se -varhome /export/home
amuser 15007 582 0 12:26:09 pts/4 2:20
  /export/home/amuser/as/appserver/lib/appservDAS domain1
amuser 15017 15009 0 12:26:20 pts/4 0:05 /usr/jdk/entsys-j2se/bin/java
  -server -cp /usr/bin/../../usr/share/lib/imq/imqb
```

11 Verify that the Application Server administration instance is accessible by entering the following URL in a browser:

`https://`*fqdn*`:`*as-admin-port*`/`

Where *fqdn* and *as-admin-port* are the fully qualified domain name and port.

12 Verify that the Application Server HTTP port is accessible by entering the following URL in a browser:

`http://`*fqdn*`:8080/`

Where *fqdn* is the fully qualified domain name.

13 Install Access Manager by running the Java ES installer. For the Configuration Type, select the Configure Later option.

14 As superuser (`root`), change the ownership of the following directories from `root` and `other` to the non-root user and non-root group, depending on your platform:

- Solaris systems: `/opt/SUNWma` and `/etc/opt/SUNWma`
- Linux systems: `/opt/sun/mobileaccess` and `/etc/opt/sun/mobileaccess`

For example:

```
# chown -R amuser:amgroup /opt/SUNWma /etc/opt/SUNWma
```

15 As superuser (`root`), change to the Access Manager `/bin` directory, depending on your platform:

- Solaris systems: `cd /opt/SUNWam/bin`
- Linux systems: `cd /opt/sun/identity/bin`

16 As superuser (`root`), make a copy of the `amsamplesilent` file. For example:

```
# cp -p amsamplesilent am.non_root_install
```

17 As superuser (`root`), edit the `am.non_root_install` file as follows:

- Set `BASEDIR` to the same value that you selected for the installation directory of Access Manager in the Java ES installer.
- Set `NEW_OWNER` to the non-root user and `NEW_GROUP` to the non-root group.
- Update the following variables: `SERVER_HOST`, `SERVER_PORT`, `DS_HOST`, `DS_PORT`, `ROOT_SUFFIX`, `COOKIE_DOMAIN`, `WEB_CONTAINER`, `AS81_HOME`, `AS81_ADMINPASSWD`, `AS81_INSTANCE_DIR`, `AS81_DOCS_DIR` and all related password fields, including `DS_DIRMGRPASSWD`, `ADMINPASSWD`, and `AMLDAPUSERPASSWD`.

Important: Set the `AS81_HOME` variable to the parent directory of the Application Server `/bin` directory.

See Example 4–1 for a sample edited `amsamplesilent` file.

18 As superuser (`root`), run the `amconfig` script with the edited `am.non_root_install` file to deploy Access Manager. For example:

```
# ./amconfig -s ./am.non_root_install
```

If you encounter the question "`Do you trust the above certificate [y|n]`" during the deployment of the Access Manager Web applications, specify "`y`" and press Enter.

19 As the non-root user, stop the Application Server domain and then restart it. First change to the `/bin` directory. For example, on Solaris systems:

```
cd /export/home/amuser/as/appserver/bin
```

Or, on Linux systems:

```
cd /export/home/amuser/as/bin
```

Then, stop and restart the Application Server domain. For example:

```
./asadmin stop-domain domain1
./asadmin start-domain --user admin domain1
```

The `asadmin start-domain` command will prompt you for the Application Server administration password.

20 Use a browser with the following URL to verify that the Access Manager Administrator Console is accessible.

`http://`*fqdn*`:8080/amserver/`

Where *fqdn* is the fully qualified domain name.

Example 4–1 Sample `amsamplesilent` File With Application Server as the Web Container

The following example shows a sample edited `amsamplesilent` file. For a description of these variables, see "Access Manager `amsamplesilent` File Configuration Variables" on page 27.

```
DEPLOY_LEVEL=1
BASEDIR=/export/home/amuser/am
SERVER_HOST=host.example.com
SERVER_PORT=8080
SERVER_PROTOCOL=http
CONSOLE_HOST=$SERVER_HOST
CONSOLE_PORT=$SERVER_PORT
CONSOLE_PROTOCOL=$SERVER_PROTOCOL
CONSOLE_REMOTE=false
DS_HOST=host.example.com
DS_PORT=8389
DS_DIRMGRDN="cn=Directory Manager"
DS_DIRMGRPASSWD=password
ROOT_SUFFIX="dc=host,dc=example,dc=com"
# ADMINPASSWD, the amadmin password, and AMLDAPUSERPASSWD,
# the amldapuser password, must be set to different values
ADMINPASSWD=password
AMLDAPUSERPASSWD=password
CONSOLE_DEPLOY_URI=/amconsole
SERVER_DEPLOY_URI=/amserver
PASSWORD_DEPLOY_URI=/ampassword
```

```
COMMON_DEPLOY_URI=/amcommon
COOKIE_DOMAIN=.iplanet.com
JAVA_HOME=/usr/jdk/entsys-j2se
AM_ENC_PWD=""
PLATFORM_LOCALE=en_US
# Non-root user and group
NEW_OWNER=amuser
NEW_GROUP=amgroup
####
XML_ENCODING=ISO-8859-1
NEW_INSTANCE=false
WEB_CONTAINER=AS8
AS81_HOME=/export/home/amuser/as/appserver
AS81_PROTOCOL=$SERVER_PROTOCOL
AS81_HOST=$SERVER_HOST
AS81_PORT=$SERVER_PORT
AS81_ADMINPORT=4849
AS81_ADMIN=admin
AS81_ADMINPASSWD="password"
AS81_INSTANCE=server
AS81_DOMAIN=domain1
AS81_INSTANCE_DIR=/export/home/amuser/as_var/domains/${AS81_DOMAIN:-domain1}
AS81_DOCS_DIR=/export/home/amuser/as_var/domains/${AS81_DOMAIN:-domain1}/docroot
# true if container is SSL enabled, installer will use SSL_PASSWORD
# to start server without user intervention
AS81_IS_SECURE=false
AS81_ADMIN_IS_SECURE=true
SSL_PASSWORD="sample"
DIRECTORY_MODE=1
USER_NAMING_ATTR=uid
ORG_NAMING_ATTR=o
ORG_OBJECT_CLASS=sunismanagedorganization
USER_OBJECT_CLASS=inetorgperson
DEFAULT_ORGANIZATION=
```

◆ ◆ ◆ CHAPTER 5

Configuring Access Manager With a Load Balancer

A load balancer distributes the client requests between the Access Manager instances in multiple server deployment. Before you use this information in the section, configure your Access Manager deployment as a site, as described in TBD, Configuring an Access Manager Deployment as a Site. A site includes multiple (two or more) instances of Access Manager installed on different host servers. All Access Managers instances must access the same Directory Server and use the same password encryption key. For information about installing Access Manager, see TBD, Installing Access Manager on Multiple Host Servers.

This section include the following information about using a load balancer:

Configuring SSL Termination for a Load Balancer

Before you configure a load balancer to handle SSL requests, first configure SSL for the Access Manger web container. For instructions, see Chapter 9, Configuring Access Manager in SSL Mode.

To configure SSL for a load balancer and Access Manager servers, consider the following cases:

- SSL configuration for only the load balancer: SSL termination.

  The load balancer terminates the SSL connection from the client and makes a separate SSL connection to the Access Manager servers.
- SSL configuration for only the Access Manager servers: SSL pass-through.

  The load balancer bypasses all the requests from the client to the Access Manager servers.

- SSL configuration for both the load balancer and Access Manager servers.

For all cases, except for the SSL pass-through configuration, you can use a normal server certificate to enable SSL termination for the load balancer. However, when you configure SSL pass-through for the load balancer and the Access Manager servers and the load balancer bypasses all the requests from the client to the Access Manager server, the following SSL problems exist for a normal server certificate:

- When a client accesses the Access Manager servers through the load balancer, the client gets the server certificate from the Access Manager server. The load balancer doesn't have an SSL server certificate and just bypasses the client requests to the back-end Access Manager servers. The client then receives a warning message saying that the host name and subject name in server certificate are different.
- To avoid the above problem, install a server certificate with the `SubjectDN` of the load balancer name; however, a problem occurs in the session validation between two Access Manager servers.

  For example, if a user gets a session from `amserver1` and a second request for the same user is directed to `amserver2`, then `amserver2` has to validate the users session to `amserver1`. When `amserver2` sends a session validation request to `amserver1`, it makes an SSL connection to `amserver1` and then gets the server certificate with the `SubjectDN` of the load balancer from `amserver1`. Because those two names (host name of `amserver1` and `subjectDN` in certificate) differ, `amserver2` stops the SSL handshaking, and the session validation fails.

To solve these problems, Access Manager provides these properties:

- `com.iplanet.am.jssproxy.trustAllServerCerts`

  If enabled (true), Access Manager ignores all certificate related issues (such as a name conflict) and continues the SSL handshaking.

Caution – To prevent a possible security risk, enable this property only for testing or when the enterprise network is tightly controlled. Avoid enabling this property if a security risk might occur (for example, if a server connects to a server in a different network).

- `com.iplanet.am.jssproxy.SSLTrustHostList`

  If enabled (true), Access Manager checks the platform server list in the `AMConfig.properties` file. If the server FQDNs of the two servers in the platform server list match, Access Manager continues the SSL handshaking.

- `com.iplanet.am.jssproxy.checkSubjectAltName`

  If enabled (by specifying a comma separated list of trusted FQDNs) and a server certificate includes the Subject Alternative Name (`SubjectAltName`) extension, Access Manager checks all name entries in the extension. If one of names in the `SubjectAltName` extension is the same as the server FQDN, Access Manager continues the SSL handshaking. Using this property is more secure than enabling the `com.iplanet.am.jssproxy.trustAllServerCerts` property. With a Public-Key Infrastructure (PKIX) definition, a certificate can have multiple subject names with `SubjectAltName` extension.

  To enable this property, set it to a comma separated list of trusted FQDNs. For example:

```
com.iplanet.am.jssproxy.checkSubjectAltName=
amserv1.example.com,amserv2.example.com
```

To get a certificate with `SubjectAltName` extension, see the next section.

Generating a CSR with the `SubjectAltName` Extension

To generate a certificate signing request (CSR) with the `SubjectAltName` extension, use the Certificate Database Tool (`certutil`). If `certutil` is not available in the `/usr/sfw/bin` directory, first install the `SUNWtlsu` package on Solaris systems or the `sun-nss-sun-nss-devel` RPM on Linux systems. If necessary, set the `LD_LIBRARY_PATH` environment variable to the appropriate `certutil` path.

For information about `certutil`, see: `http://www.mozilla.org/`

This section describes how to use the `certutil` if you are using Web Server or Application Server as the web container. If you are using BEA WebLogic Server or IBM WebSphere Application Server as the web container, refer to the respective BEA or IBM product documentation.

▼ To Generate a CSR with the `SubjectAltName` Extension

1 Log in as or become superuser (`root`).

2 Create a new certificate database (`cert8.db`) using the `certutil -N` option. If necessary, first create a directory for your database. For example:

```
# mkdir certdbdir
# cd certdbdir
# certutil -N -d .
```

When prompted by `certutil`, enter the password to encrypt your keys:

```
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.

Enter new password: your-password
Re-enter password:  your-password
```

3 Generate the CSR with the `SubjectAltName` extension. For example:

```
# certutil -R -s "cn=lb.example.com,o=example.com,c=us"
-o server.req -d . -a -8 amserv1.example.com,amserv2.example.com
```

When prompted by `certutil`, enter the password (or pin) and then type keys to generate the random seed to create your key:

```
Enter Password or Pin for "NSS Certificate DB": your-password

A random seed must be generated that will be used in the
creation of your key.  One of the easiest ways to create a
random seed is to use the timing of keystrokes on a keyboard.

To begin, type keys on the keyboard until this progress meter
is full.  DO NOT USE THE AUTOREPEAT FUNCTION ON YOUR KEYBOARD!

Continue typing until the progress meter is full:

|************************************************************|

Finished.  Press enter to continue:

Generating key.  This may take a few moments...
```

4 Send the CSR (`server.req` file in the example) to the Certificate Authority (CA). Get the server certificate and add it to the certificate database using the `certutil -A` option.

5 Copy the certificate database (`cert8.db`) to the web container directory.

- Web Server. Copy the `cert8.db` and `key3.db` databases to the `/opt/SUNWwbsrv/alias` directory and rename them using the Web Server instance name. For example:

  ```
  https-webserver.example.com-webserver-cert8.db
  https-webserver.example.com-webserver-key3.db
  ```

- Application Server. Copy the `cert8.db` and `key3.db` databases to the instance `/config` directory. For example:

  ```
  /var/opt/SUNWappserver/domains/domain1/config/cert8.db
  /var/opt/SUNWappserver/domains/domain1/config/key3.db
  ```

Configuring Access Manager For Load Balancer Cookies

To configure Access Manager for load balancer cookies, update the configuration for all Access Manager instances in the deployment so that the instances can recognize the load balancer. In this scenario, multiple (two or more) Access Manager instances are deployed on different host servers. A load balancer routes client requests to the various Access Manager instances. All Access Manager instances use the same Directory Server.

▼ To Configure Access Manager For Load Balancer Cookies

1 **In the Access Manager Console, configure the Access Manager deployment as a site, as described in TBD, Configuring an Access Manager Deployment as a Site. When you configure a deployment as a site, Access Manager automatically sets the `fqdnMap` property (in memory) to include the load balancer.**

2 **In the `AMConfig.properties` file for each Access Manager instance, add the following properties:**

```
com.iplanet.am.lbcookie.name=amlbcookie
com.iplanet.am.lbcookie.value=amserver
```

where *amlbcookie* is the load balancer cookie, and *amserver* is the name of the Access Manager host server for the instance.

3 **Restart all Access Manager instances by restarting the respective web container.**

Configuring a Load Balancer with SAML

In this scenario, an Access Manager site is using a load balancer to distribute client requests to various Access Manager instances, and the site has implemented the Security Assertions Markup Language (SAML) service. When a request is sent to an Access Manager instance through a load balancer, the instance must know which other Access Manager server in the deployment issued the original assertion or artifact in order to retrieve the SAML assertion.

The deployment must first be configured as a site. Multiple Access Manager instances are installed on host servers, and a load balancer routes client requests to the various instances. All Access Manager instances access the same Directory Server. Access Manager session failover is optional.

▼ To Configure a Load Balancer with SAML

1 **The Access Manager deployment must be configured as a site in order for SAML load balancing to**

work. If you haven't configured the Access Manager deployment as a site, follow the instructions in TBD, Configuring an Access Manager Deployment as a Site.

2. **Log in to the Access Manager Console as** `amadmin`**.**

3. **In the Access Manager Console, click** `Federation` **and then** `SAML`**.**

4. **Under the** `Properties` **section in** `SAML Profile`**, add or modify the following entries:**

   - Site Identifiers. Add each Access Manager instance in the deployment. All Access Manager instances must share the same Site ID and Site Issuer Name.
   - Trusted Partners. Add your partner's deployment site's Source ID (site ID), Issuer Name, and Host List. The unique Source ID (site ID) and Issuer Name for the Access Manager servers and the URL or IP address or host name of the load balancer will identify the deployment and will be given out to your partner's site for configuration.

     For information about these fields, see the Sun Java System Access Manager 7.1 2006Q4 Federation and SAML Administration Guide.

5. **Click** `Save` **to save your changes.**

Setting the `fqdnMap` Property

If you have configured an Access Manager deployment as a site, Access Manager automatically sets the `fqdnMap` property (in memory) to include the load balancer , and you do not need to set this property in the `AMConfig.properties` file. However, for the following Access Manager deployments, you must explicitly set the property:

- The deployment is not configured as a site.
- The deployment has virtual hosts that are mapped to a physical host.

If you need to set the `fqdnMap` property, set the property to the load balancer in the `AMConfig.properties` file for each Access Manager instance in the deployment. If necessary, first remove the comment character (`#`) from the property. For example:

```
com.sun.identity.server.fqdnMap[lb.example.com]=lb.example.com
```

Accessing an Access Manager Instance Through a Load Balancer

Accessing an Access Manager instance through a load balancer depends on the mode (realm or legacy) and the console you want to access. Use the following syntax to access an Access Manager instance through a load balancer:

`http://`*loadbalancer*`.`*domain*`:`*port*`/amserver/console|/amconsole`

In legacy mode, you can access both consoles:

- New Access Manager 7.1 2006Q4 Console. For example:
  `http://loadbalancer.example.com:80/amserver/console`
- Access Manager 6 2005Q1 Console. For example:
  `http://loadbalancer.example.com:80/amconsole`

In realm mode, you can access only the new Access Manager 7.1 2006Q4 Console. For example:

`http://loadbalancer.example.com:80/amserver/console`

◆ ◆ ◆ CHAPTER 6

Configuring an Access Manager Deployment as a Site

You can configure an Access Manager deployment as a site, which provides centralized configuration management for the deployment. When Access Manager is configured as a site, client requests always go through a load balancer, which simplifies the deployment as well as resolves issues such as a firewall between the client and the back-end Access Manager servers.

An Access Manager site includes the following components:

- Multiple (two or more) Access Manager instances are deployed on at least two different host servers. For example, you might deploy two instances on one server and a third instance on another server. Or you might deploy all instances on different servers. You can also configure the Access Manager instances in session failover mode, if required for your deployment.
- One or more load balancers route client requests to the various Access Manager instances. You configure each load balancer according to your deployment requirements (for example, to use round-robin or load average) to distribute the load between the Access Manager instances.
- All Access Manager instances access the same Directory Server.

Configuring an Access Manager Site

If you have an Access Manager multiple server deployment, use either of these methods to configure your deployment as a site:

- If you plan to implement Access Manager session failover, the `amsfoconfig` script configures a deployment as a site. See Implementing Access Manager Session Failover.
- If you don't plan to implement session failover, follow the steps in this section.

When you configure a deployment as a site, you perform these functions in the Access Manager Console:

- Add the load balancer URL to the Site Name (site ID).
- Map the load balancer Site Name (site ID) to each Access Manager instance in the Platform Server List.

- Add the load balancer to the Realm/DNS Aliases.

In addition, Access Manager automatically sets the `fqdnMap` property (in memory) to include the load balancer, so you do not need to explicitly set this property in the `AMConfig.properties` file.

▼ To Configure Access Manager as a Site

The following procedure refers to the Access Manager 7.1 2006Q4 Console in Realm Mode.

1. **Log in to the Access Manager Console as `amAdmin`.**

2. **Add the load balancer URL to the `Site Name`:**

   a. **In the Access Manager Console, click `Configuration`, `System Properties`, and then `Platform`.**

   b. **Under `Site Name`, click `New` and enter the following values for the load balancer:**

      - `Server`: Load balancer protocol, host name, and port. For example: `http://lb.example.com:80`
      - `Site Name`: Unique two-digit site identifier (site ID). For example: 10

        When you are finished, click `OK`.

   c. **After adding the load balancer to the `Site Name`, click `Save`. The entry for the load balancer now includes the site ID. For example:** `http://lb.example.com:80|10`

      The site ID must be unique with respect to server IDs and other site IDs. For example, you cannot use 01 for both a site ID and a server ID.

3. **On the same Console panel, map the load balancer to each Access Manager instance:**

   a. **In the `Server` list under `Instance Name`, click each instance name to display the `Edit Server Instance` panel for the instance.**

   b. **Map the `Site Name` (site ID) for the load balancer to the Access Manager instance. For example, using a load balancer with a `Site Name` of 10, for the first server, the `Instance Name` would `01(|10)`.**

   c. **Click `OK` and repeat the steps for the other Access Manager instances.**

      When you are finished, all Access Manager instances should be mapped to the load balancer. For example:

      ```
      http://amserver1.example.com:8080|01|10
      http://amserver2.example.com:8080|02|10
      http://amserver3.example.com:8080|03|10
      ```

   d. **Click `Save` to save the configuration.**

4 **Add the Realm/DNS alias for the load balancer:**

   a. **In the Access Manager Console, click** `Access Control` **and then the root or top-level realm under** `Realm Name`**.**

   b. **Under** `Realm Attributes`**, add the load balancer to** `Realm/DNS Aliases` **and then click** `Add`**. For example:** `lb.example.com`**.**

   c. **Click** `Save` **to save your changes.**

5 **For clients such as a policy agent, the load balancer (as opposed to the individual Access Manager instances) should be the sole entry point. For example, if you are using a policy agent, modify the appropriate entries in the** `AMAgent.properties` **file to point to the load balancer.**

◆ ◆ ◆ CHAPTER 7

Configuring Access Manager Sessions

Access Manager session configuration includes:

Setting Session Quota Constraints

The session quota constraints feature allows Access Manager to limit users to a specific number of active, concurrent sessions based on configurable attributes. An Access Manager administrator can set session quota constraints at the following levels:

- Globally. Constraints apply to all users.
- To an entity (organization or realm, role, or user). Constraints apply only to the specific users that belong to the entity.

Deployment Scenarios for Session Quota Constraints

The following Access Manager deployments support session quota constraints:

- Access Manager Single Server Deployment

  In this scenario, Access Manager is deployed on a single host server. Access Manager maintains the active session counts in memory for all logged in users. When a user attempts to log in to the server, Access Manager checks whether the number of the valid sessions for the user exceeds the session quota and then takes action based on the configured session quota constraints options.

- Access Manager Session Failover Deployment

  In this scenario, multiple instances of Access Manager are deployed on different host servers in a session failover configuration. The Access Manager instances are configured for session failover using Sun Java System Message Queue (Message Queue) as the communications broker and the Berkeley DB by Sleepycat Software, Inc. as the session store database. For more information about Access Manager session failover, see Chapter 8.

In a session failover deployment, when a user attempts to log in, the Access Manager server receiving the session creation request first retrieves the session quota for the user from the Access Manager identity repository. Then, the Access Manager server fetches the session count for the user directly from the centralized session repository (accumulating all the sessions from all the Access Manager servers within the same site) and checks whether the session quota has been exhausted. If the session quota has been exhausted for the user, the Access Manager server takes action based on the configured session quota constraints options.

If session constraints are enabled in a session failover deployment and the session repository is not available, users (except superuser) are not allowed to log in.

In a session failover deployment, if an Access Manager instance is down, all the *valid* sessions previously hosted by that instance are still considered to be valid and are counted when the server determines the actual active session count for a given user. An Access Manager multiple server deployment that is not configured for session failover does not support session quota constraints.

Multiple Settings For Session Quotas

If a user has multiple settings for session quotas at different levels, Access Manager follows this precedence to determine the actual quota for the user:

- user (highest)
- role/organization/realm (based on the conflict resolution levels)
- global (lowest)

For example, Ken is a member of both the marketing and management roles. Session quotas are defined as follows (all have the same conflict resolution level):

- organization - 1
- marketing role - 2
- management role - 4
- user Ken - 3

Ken's quota is 3.

For more information about the session quota constraints attributes, see the Access Manager Console online help.

Configuring Session Quota Constraints

To configure session quota constraints, the top-level Access Manager administrator (such as `amAdmin`) must set specific attributes in the Access Manager Console for one of the Access Manager instances in your deployment.

▼ To Configure Session Quota Constraints

1. **Log in to Access Manager Console as a top-level Access Manager administrator (such as `amAdmin`) .**

2. **Set the following attributes in the Access Manager Console for one of the Access Manager instances.**

   Enable Quota Constraints is a global attribute that enables or disables the session quota constraints feature. If this attribute is enabled, Access Manager enforces session quota constraints whenever a user attempts to logs in via a new client (and thus create a new session).

   The default is disabled (OFF).

   Read Timeout for Quota Constraint defines the time in milliseconds that an inquiry to the session repository for the active user session counts continues before timing out. If the maximum wait time is reached due to the unavailability of the session repository, the session creation request is rejected.

   The default is 6000 milliseconds.

   Resulting Behavior If Session Quota Exhausted determines the behavior if a user exhausts the session constraint quota. This attribute takes effect only if the "Enable Quota Constraints" attribute is enabled. Values can be:

   - `DENY_ACCESS`. Access Manager rejects the login request for a new session.
   - `DESTROY_OLD_SESSION`. Access Manager destroys the next expiring existing session for the same user and allows the new login request to succeed.

   The default is `DESTROY_OLD_SESSION`.

   Exempt Top-Level Admins From Constraint Checking specifies whether session constraint quotas apply to the administrators who have the Top-level Admin Role. This attribute takes effect only if the "Enable Quota Constraints" attribute is enabled.

   The default is NO.

   The super user defined for Access Manager in the `AMConfig.properties` file (`com.sun.identity.authentication.super.user`) is always exempt from session quota constraint checking.

   Active User Sessions defines the maximum number of concurrent sessions for a user. Access Manager includes both a dynamic attribute and a user attribute, with same attribute name.

   The default is 5.

   Note – If you reset any of these attributes, you must restart the server for the new value to take effect.

3. **When you have finished click Save.**

Configuring Session Property Change Notifications

The session property change notification feature causes Access Manager to send a notification to all registered listeners when a change occurs on a specific session property. This feature takes effect when the "Enable Property Change Notifications" attribute is enabled (ON) in the Access Manager Console.

For example, in a single sign-on (SSO) environment, one Access Manager session can be shared by multiple applications. When a change occurs on a specific session property defined in the "Notification Properties" list, Access Manager sends a notification to all registered listeners.

All client applications participating in the SSO automatically get the session notification if they are configured in the notification mode. The client cached sessions are automatically updated based on the new session state (including the change of any session property, if there is any). An application that wants to take a specific action based on a session notification can write an implementation of the `SSOTokenListener` interface and then register the implementation through the `SSOToken.addSSOTokenListener` method. For more information, see the Sun Java System Access Manager 7.1 2006Q4 Developer's Guide.

▼ To Configure Session Property Change Notifications

1. **Log in to Access Manager Console as `amAdmin`.**
2. **Click the Configuration tab.**
3. **Under Global Properties, click Session.**
4. **Set "Enable Property Change Notifications" to ON.**
5. **In the "Notification Properties" list, add each property for which you want a notification sent when the property is changed.**
6. **When you have finished adding properties to the list, click Save.**

◆ ◆ ◆ CHAPTER 8

Implementing Session Failover

Access Manager provides a web container independent session failover implementation using Sun Java System Message Queue (Message Queue) as the communications broker and the Berkeley DB by Sleepycat Software, Inc. as the session store database. The information about session failover includes these topics:

Access Manager Session Failover Scenario

Figure 8–1 shows an Access Manager session failover deployment scenario that includes these components:

- Three Access Manager instances, running on different host servers on supported web containers. All Access Manager instances access the same Directory Server (not shown in the figure).
- Message Queue brokers, running in cluster mode on different servers.
- Berkeley DB client (`amsessiondb`), running on the same servers as the Message Queue brokers.
- Load balancer to improve performance and security.
- Client requests can originate from a Web browser, C or Java application using the Access Manager SDK, or a J2EE/Web agent.

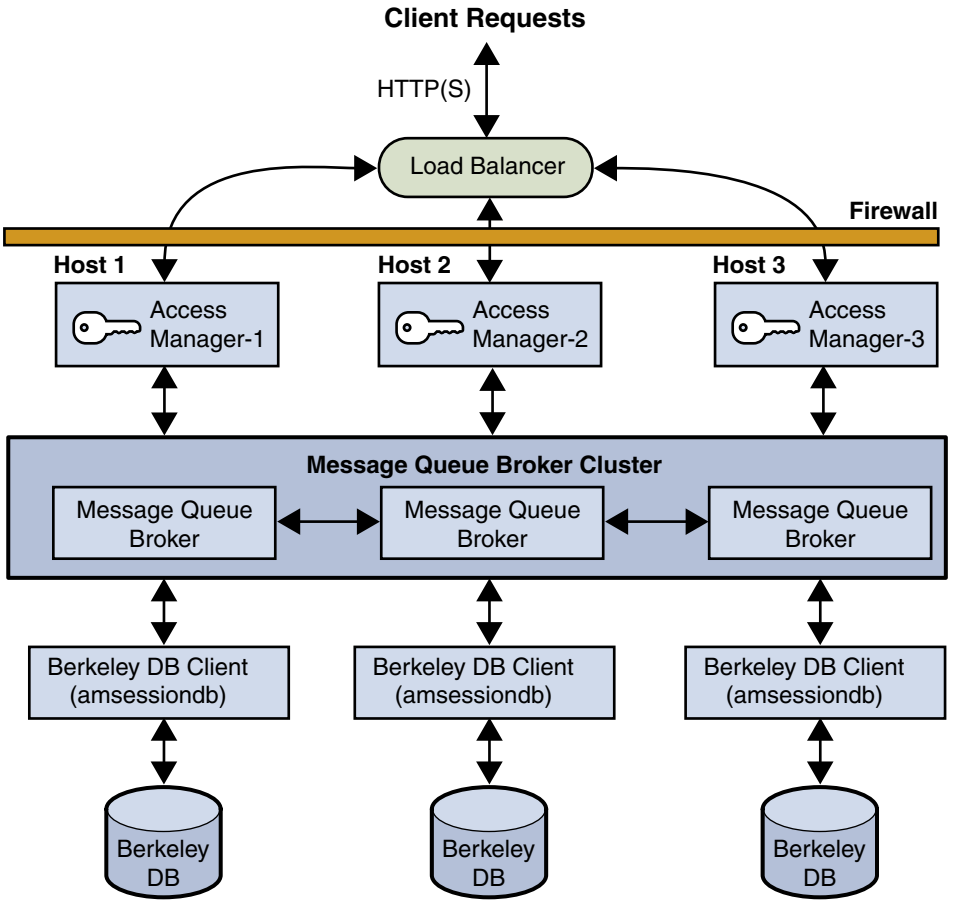

FIGURE 8–1 Access Manager Session Failover Scenario

Installing the Session Failover Components

The following table describes how to install the components required for Access Manager session failover.

TABLE 8–1 Installation of Access Manager Session Failover Components

Component	How to install ...
Access Manager	Install the first instance of Access Manager on each host server using the Java ES installer. The installer adds the required session failover Solaris packages or Linux RPMs. Reference: Sun Java Enterprise System 2006Q4 Installation Guide for UNIX. When you install Access Manager using the Java ES installer, you can select either Realm Mode (version 7.x) or Legacy Mode (version 6.x). Access Manager session failover is supported in both modes. After you run the Java ES installer, run the `amconfig` script to: ■ Configure the first Access Manager instance, if you specified the Configure Later option during installation. ■ Redeploy or reconfigure an installed Access Manager instance. For information, see TBD, Installing Access Manager on Multiple Host Servers.
Message Queue	Install Message Queue using the Java ES installer. Reference: Sun Java Enterprise System 2006Q4 Installation Guide for UNIX
Berkeley DB Session Failover Client (Access Manager subcomponent)	Install the Session Failover Client using the Java ES installer. On the installer Component Selection page, check Session Failover Client. The Java ES installer adds the Access Manager packages or RPMs required for the Berkeley DB and `amsessiondb` client: ■ Solaris OS: `SUNWamsfodb`, `SUNWbdb`, and `SUNWbdbj` packages. ■ Linux OS: `sun-identity-sfodb`, `sun-berkeleydatabase-core`, and `sun-berkeleydatabase-java` RPMs. You can install the Session Failover Client on a server that is running Access Manager; however, for improved performance, consider installing the subcomponent on a server that is not running Access Manager.

Caution – In a multiple server deployment where all Access Manager instances share the same Directory Server, all Access Manager instances must use the same password encryption key value. When you install the first Access Manager instance, save the password encryption key value from the `am.encryption.pwd property` in the `AMConfig.properties` file. Then, when you run the Java ES installer or `amconfig` script to install or configure Access Manager instances on other host servers, use this same value for the password encryption key.

Configuring Access Manager for Session Failover

Configuring Access Manager for session failover involves these steps:

Each step is described in detail in the following sections.

Tip – To determine if session failover is enabled for a deployment, change the `com.iplanet.services.debug.level` property from `error` to `message` in the `AMConfig.properties` file. Then, check the `amSession` logs in the `/var/opt/SUNWam/debug` directory on Solaris systems or the `/var/opt/sun/identity/debug` directory on Linux systems.

1–Disabling Cookie Encoding

On each host server that is running an Access Manager instance, disable cookie encoding as follows, depending on the web container:

- If Web Server is the web container, make sure the following property in the `AMConfig.properties` file is set to false (which is the default value set by the Java ES installer):

  ```
  com.iplanet.am.cookie.encode=false
  ```

  In the `sun-web.xml` file, set the `encodeCookies` property to false. For example:

  ```
  <sun-web-app>
  <property name="encodeCookies" value="false"/>
  ...
  </sun-web-app>
  ```

- If Application Server, BEA WebLogic, or IBM WebSphere Application Server is the web container, set the following property in the `AMConfig.properties` file to false:

  ```
  com.iplanet.am.cookie.encode=false
  ```

The Access Manager client should not do any cookie encoding or decoding. A remote SDK client must be in sync with the Access Manager server side settings, either in the `AMConfig.properties` file or the web container's `sun-web.xml` file.

2–Editing the Web Container `server.xml` File

On each host server that is running an Access Manager instance, add the installed locations of `imq.jar` and `jms.jar` in the `server.xml` (or equivalent) configuration file for the Access Manager web container. For example, on Solaris systems:

```
<JAVA javahome="/usr/jdk/entsys-j2se" serverclasspath=
"/usr/share/lib/imq.jar:/usr/share/lib/jms.jar:
/opt/SUNWwbsvr/bin/https/jar/webserv-rt.jar:
${java.home}/lib/tools.jar:
/opt/SUNWwbsvr/bin/https/jar/webserv-ext.jar:
/opt/SUNWwbsvr/bin/https/jar/webserv-jstl.jar:
/usr/share/lib/ktsearch.jar"
```

3–Adding a New User in the Message Queue Server

If you don't want to use the `guest` user as the Message Queue user name and password, add a new user and password to connect to the Message Queue broker on servers where Message Queue is installed. For example, on Solaris systems, to add a new user named `amsvrusr`:

```
# /usr/bin/imqusermgr add -u amsvrusr -p password
```

Then, make the `guest` user inactive by issuing the following command:

```
# /usr/bin/imqusermgr update -u guest -a false
```

4–Editing the `amsessiondb` Script (if Needed)

The `amsessiondb` script is called by the `amsfo` script to start the Berkeley DB client (`amsessiondb`), create the database, and set specific database values. The script contains variables that specify various default paths and directories:

```
JAVA_HOME=/usr/jdk/entsys-j2se/
IMQ_JAR_PATH=/usr/share/lib
JMS_JAR_PATH=/usr/share/lib
BDB_JAR_PATH=/usr/share/db.jar
BDB_SO_PATH=/usr/lib
AM_HOME=/opt/SUNWam
```

If any of these components are not installed in their default directories, edit the `amsessiondb` script and set the variables, as needed, to the correct locations.

5–Running the `amsfoconfig` Script

Access Manager provides the `amsfoconfig` script to configure an Access Manager deployment for session failover.

Requirements to Run the `amsfoconfig` Script

To run the `amsfoconfig` script, an Access Manager deployment must meet the following requirements:

- Two or more Access Manager instances must be installed and configured in the deployment, but the deployment cannot be configured as a site. If the `amsfoconfig` script determines that the deployment is configured as a site or that any of the server entries in the platform server list are site enabled, the script displays a message and exits. To configure session failover manually, see Configuring Session Failover Manually

- The Java Message Queue (MQ) broker must be installed and configured on at least two servers in the deployment.
- The Berkeley DB client and database must be installed and configured in the deployment.
- Directory Server must be running, accessible to the script, and configured with Access Manager data.

Functions of the `amsfoconfig` Script

The `amsfoconfig` script reads the `amsfo.conf` configuration file and then configures an Access Manager deployment for session failover by performing these functions:

- Configures a new site. The script uses the Access Manager instances in the platform server list and the load balancer information from the `amsfo.conf` file to create a new site for the Access Manager session failover deployment. The script modifies the existing platform server list, so that after the site is configured, all server entries under the platform server list then belong to the site.

  For example, `http://server1.example.com:80|01` changes to `http://server1.example.com:80|01|10`, if the default value of 10 is used as the `SiteID`.
- Modifies the existing Realm/DNS alias list. The script appends the host name of the load balancer to the list. This host name is obtained from the `lbServerHost` variable of the `amsfo.conf` file.
- Loads session failover configuration XML into Directory Server. The script dynamically generates the session configuration XML file based on the configuration information and loads the generated XML into Directory Server. This information corresponds to the Secondary Configuration Instance under Session in the Access Manager Console.

The following table lists the Access Manager session failover scripts and configuration files.

TABLE 8–2 Access Manager Session Failover Scripts and Configuration Files

Name	Description and Location
`amsofconfig`	Script to configure Access Manager for session failover.
	Solaris systems: *AccessManager-base*`/SUNWam/bin`
	Linux systems: *AccessManager-base*`/identity/bin`
`amsfo`	Script to start and stop the Message Queue broker and `amsessiondb` client.
	Solaris systems: *AccessManager-base*`/SUNWam/bin`
	Linux systems: *AccessManager-base*`/identity/bin`
`amsfopasswd`	Script to generate the encrypted Message Queue broker user password.
	Solaris systems: *AccessManager-base*`/SUNWam/bin`
	Linux systems: *AccessManager-base*`/identity/bin`

TABLE 8–2 Access Manager Session Failover Scripts and Configuration Files *(Continued)*

Name	Description and Location
`amsfo.conf`	Session failover configuration file.
	Solaris systems: *AccessManager-base*`/SUNWam/lib`
	Linux systems: *AccessManager-base*`/sun/identity/lib`
`amProfile.conf`	Session failover environment file.
	Solaris systems: `etc/opt/SUNWam/config`
	Linux systems: `etc/opt/sun/identity/config`
AccessManager-base represents the base installation directory for Access Manager. The default values are:	
Solaris systems: `/opt`	
Linux systems: `/opt/sun`	

Running the `amsfoconfig` Script

The `amsfoconfig` script configures Access Manager for session failover.

▼ To Run the `amsfoconfig` Script

1. **Log in as or become superuser (`root`).**
2. **Set the variables in the `amsfo.conf` file, as described in Table 6–3.**
3. **Run the script. For example, on a Solaris system with Access Manager installed in the default directory:**

   ```
   # cd /opt/SUNWam/bin
   # ./amsfoconfig
   ```

   The script displays status information as it runs.
4. **When the `amsfoconfig` script prompts you, enter the following passwords:**
   - Access Manager administrator (`amAdmin`) password
   - Message Queue broker user password
5. **To check the results, see the `/var/tmp/amsfoconfig.log` file.**

Variables in the `amsfo.conf` File

The following table describes the variables in the `amsfo.conf` file that are used by the `amsfoconfig` script. Set these variables as needed for your deployment before you run the `amsfoconfig` script.

TABLE 8–3 Variables in the `amsfo.conf` File Used by the `amsfoconfig` Script

Variable	Description
`CLUSTER_LIST`	Message Queue broker list participating in the cluster. The format is: *host1*`:`*port*`,`*host2*`:`*port*`,`*host3*`:`*port* For example: `jmq1.example.com:7777,jmq2.example.com:7777, jmq3.example.com:7777` There is no default.
`lbServerPort`	Port for the load balancer. The default is 80.
`lbServerProtocol`	Protocol (`http` or `https`) used to access the load balancer. The default is `http`.
`lbServerHost`	Name of the load balancer. For example: `lbhost.example.com`
`SiteID`	Identifier for the new site (and the load balancer) that the `amsfoconfig` script will create. `SiteID` can be any value greater than the Server IDs that already exist in the platform server list. The default is 10.

`amsfoconfig` Script Sample Run

The following example shows a sample run of the `amsfoconfig` script.

```
Welcome to Sun Java System Access Manager 7.1 2006Q4

Session Failover Configuration Setup script.
=============================================================
=============================================================
Checking if the required files are present...
=============================================================

Running with the following Settings.
-----------------------------------------------
Environment file: /etc/opt/SUNWam/config/amProfile.conf
Resource file: /opt/SUNWam/lib/amsfo.conf
         -----------------------------------------------
Using /opt/SUNWam/bin/amadmin

Validating configuration information.
Done...

Please enter the LDAP Admin password:
```

```
(nothing will be echoed): password1
Verify: password1
Please enter the JMQ Broker User password:
(nothing will be echoed): password2
Verify: password2

Retrieving Platform Server list...
Validating server entries.
Done...

Retrieving Site list...
Validating site entries.
Done...

Validating host: http://amhost1.example.com:7001|02
Validating host: http://amhost2.example.com:7001|01
Done...

Creating Platform Server XML File...
Platform Server XML File created successfully.

Creating Session Configuration XML File...
Session Configuration XML File created successfully.

Creating Organization Alias XML File...
Organization Alias XML File created successfully.

Loading Session Configuration schema File...
Session Configuration schema loaded successfully.

Loading Platform Server List File...
Platform Server List server entries loaded successfully.

Loading Organization Alias List File...
Organization Alias List loaded successfully.

Please refer to the log file /var/tmp/amsfoconfig.log for additional
information.
################################################################
Session Failover Setup Script. Execution end time 10/05/05 13:34:44
################################################################
```

Starting and Stopping the Session Failover Components

Access Manager provides the `amsfo` script to perform these functions:

- Start and stop the Java Message Queue (MQ) broker specified for the session failover deployment.
- Start and stop the `amsessiondb` client specified for the session failover deployment.
- Read the `amsfo.conf` configuration file and take specific actions based on variables in the file. For example, you can have the script first delete and then recreate the Berkeley DB database.
- Write the `amsessiondb.log`, `jmq.pid`, and `amdb.pid` files in the `/tmp/amsession/logs/` directory. The default log directory is determined by the `LOG_DIR` variable in the `amsfo.conf` file.

To start the Access Manager session failover components, follow this sequence:

1. Set the variables in the in the `amsfo.conf` configuration file, as required by your deployment. For a description of these variables, see Table 6–4
2. Run the `amsfo` script to start the Java Message Queue (MQ) broker and the `amsessiondb` client. For detailed information, see Running the amsfo Script.
3. Start each Access Manager instance by starting the respective web container. For information, see the Sun Java System Access Manager 7.1 2006Q4 Administration Guide.

Running the `amsfo` Script

The `amsfo` script includes the start and stop options:

Usage: `amsfo { start | stop }`

▼ To Run the `amsfo` Script

1 Log in as or become superuser (`root`).

2 Set the variables in the `amsfo.conf` file, as required for your deployment. For a description of these variables, see Table 6–4.

3 Run the script. For example, to start the session failover components on a Solaris system with Access Manager installed in the default directory:

```
# cd  /opt/SUNWam/bin
# ./amsfo start
```

4 To check the results of the script, see the `/tmp/amsession/logs/amsessiondb.log` file.

Variables in the `amsfo.conf` Configuration File

Set the following variables as needed for your deployment before you run the `amsfo` script.

TABLE 8–4 `amsfo.conf` Configuration File

Variable	Description
`AM_HOME_DIR`	Access Manager default installation directory. The default directory depends on the platform: Solaris systems: *AccessManager-base*`/SUNWam` Linux systems: *AccessManager-base*`/identity` *AccessManager-base* represents the base installation directory for Access Manager. The default values are `/opt` on Solaris systems and `/opt/sun` on Linux systems.
`AM_SFO_RESTART`	Specifies (true or false) whether the script should automatically restart the `amsessiondb` client. The default is true (restart the `amsessiondb` client).
`CLUSTER_LIST`	Message Queue broker list participating in the cluster. The format is: *host1*`:`*port*`,`*host2*`:`*port*`,`*host3*`:`*port* For example: `jmq1.example.com:7777,jmq2.example.com:7777, jmq3.example.com:7777` There is no default.
`DATABASE_DIR`	Directory where the session database files will be created. The default is `"/tmp/amsession/sessiondb"`.
`DELETE_DATABASE`	Specifies (true or false) whether the script should delete and then create a new database when the `amsessiondb` process is restarted. The default is true.
`LOG_DIR`	Location of the log directory. The default is `"/tmp/amsession/logs"`.
`START_BROKER`	Specifies (true or false) whether the Message Queue broker should be started with the `amsessiondb` process. Set this variable as follows: true - The Message Queue broker will run on the same machine as the `amsessiondb` process. false - The Message Queue broker and the `amsessiondb` process will run on different machines. The default is true.
`BROKER_INSTANCE_NAME`	Name of the Message Queue broker instance to start. The default is `aminstance`.

TABLE 8–4 `amsfo.conf` Configuration File *(Continued)*

Variable	Description
`BROKER_PORT`	Port for the local Message Queue broker instance. The default is 7777.
`BROKER_VM_ARGS`	Java VM arguments. The default is `"-Xms256m -Xmx512m"`, which sets the maximum value based on the system resources.
`USER_NAME`	User name used to connect to the Message Queue broker. The default is guest. If you specified a different user name under step 3–Add a New User in the Message Queue Server, set `USER_NAME` to that name.
`PASSWORDFILE`	Location of the password file that contains the encrypted password used to connect to the Message Queue broker. To generate the encrypted password, use the `amsfopasswd` script, as described in `amsfopasswd` Script The default is *$AM_HOME_DIR*`/.password`, where *$AM_HOME_DIR* specifies the Access Manager default installation directory.

Running the `amsfopasswd` Script

The `amsfopasswd` script accepts the Message Queue broker password in clear text and returns the encrypted password in a file. You can then use this file as input to the `amsfo` script (`PASSWORDFILE` variable).

The `amsfopasswd` script is located in the following directory:

- Solaris systems: *AccessManager-base*`/SUNWam/bin`
- Linux systems: *AccessManager-base*`/identity/bin`

The default *AccessManager-base* installation directory is `/opt` on Solaris systems and `/opt/sun` on Linux systems.

Use the following syntax to run the `amsfopasswd` script.

```
amsfopasswd -f filename | --passwordfile filename
            -e password | --encrypt password
amsfopasswd -h | --help
```

The following table describes the `amsfopasswd` script arguments.

TABLE 8–5 `amsfopasswd` Script Arguments

Argument	Description
`-f` *filename* `|` `--passwordfile` *filename*	Path to the destination file where `amsfopasswd` stores the encrypted password.

TABLE 8–5 `amsfopasswd` Script Arguments *(Continued)*

Argument	Description
`-e` *password* | `--encrypt` *password*	Clear text password that `amsfopasswd` encrypts.
`-h | --help`	Display the `amsfopasswd` command usage and then exit.

▼ To Run the `amsfopasswd` Script

1. **Log in as or become superuser (`root`).**
2. **Run the `amsfopasswd` script. For example, on a Solaris system with Access Manager installed in the default directory:**

```
# cd /opt/SUNWam/bin
# ./amsfopasswd -f /opt/SUNWam/.password -e mypassword
```

3. **Use the encrypted password in the `/opt/SUNWam/.password` file as input to the `amsfo` script (`PASSWORDFILE` variable)**

Configuring Session Failover Manually

In some situations, you might need to manually configure Access Manager for session failover. For example, you do not plan to run the `amsfoconfig` script. Or, the `amsfoconfig` script exited with one of the following messages before finishing the configuration: “Site is already configured” or “Server entry is already site configured”.

These steps describe how to manually configure Access Manager for session failover:

- “1–Install the Required Components in the Deployment” on page 90
- “2–Configure the Access Manager Deployment as a Site” on page 91
- “3–Create a New Secondary Configuration Instance for the Load Balancer” on page 91
- “4–Perform Session Failover Miscellaneous Configuration Tasks” on page 91
- “5–Start the Session Failover Components” on page 92
- “`amsessiondb` Script” on page 92

These steps are equivalent to the previous steps that described how to install the required components, configure session failover using the `amsfoconfig` script and then start the various components.

1–Install the Required Components in the Deployment

Install all components in the deployment, including Access Manager instances, load balancer, Message Queue, and the Berkeley DB client. For more information, see Installing the Session Failover Components.

2–Configure the Access Manager Deployment as a Site

If you do not plan to run the `amsfoconfig` script, which configures multiple Access Manager instances and a load balancer as a site, you must configure the deployment, as described in TBD, Configuring an Access Manager Deployment as a Site.

3–Create a New Secondary Configuration Instance for the Load Balancer

To create a new secondary configuration instance for your load balancer, follow these steps:

1. Log in to the Access Manager 7.1 2006Q4 Console as `amAdmin`.
2. Click `Configuration`, `Global Properties`, `Session`, and then `Secondary Configuration Instance`.
3. c. Click `New`, and add the following values:
   - `Name`. Load balancer URL. For example: `http://lb.example.com:80`
   - `Session Store User`. Name you are using to connect to the Message Queue Server (if other than guest).
   - `Session Store Password`. Password for the Session Store User.
   - `Maximum Wait Time`. 5000 (Use the default unless you require another value).
   - `Database Url`: Message Queue broker address list. For example:

     ```
     mqsvr1.example.com:7777,mqsvr2.example.com:7777,
     mqsvr3.example.com:7777
     ```

     The default Message Queue port is 7676. If you are using Application Server as the web container, however, consider using another port, because port 7676 might already be in use by Application Server. For the range of the valid port numbers, refer to the Message Queue documentation.
4. Click `Add` to save your changes.

4–Perform Session Failover Miscellaneous Configuration Tasks

Perform the following tasks (which are the same as if you are running the `amsfoconfig` script):

- Disable Cookie Encoding.
- Edit the Web Container `server.xml` File.
- Add a New User in the Message Queue Server.
- Edit the `amsessiondb` Script (if needed).

5–Start the Session Failover Components

Run the `amsfo` script to start the Message Queue broker and Berkeley DB client (`amsessiondb`). Then, start each Access Manager instance by starting the respective web container. See Starting the Session Failover Components.

`amsessiondb` Script

The `amsessiondb` script is called by the `amsfo` script to start the Berkeley DB client (`amsessiondb`), create the database, and set specific database values.

Note – The recommended method to start and stop the Access Manager session failover components is to run the `amsfo` script and let it call the `amsessiondb` script. The following information is included only in case you might need to run the `amsessiondb` script independently.

Before you run the `amsessiondb` script, make sure you have the paths set correctly, as described under 4–Edit the amsessiondb Script (if Needed).

When you run the `amsessiondb` script, you can enter the Message Queue broker password on the command line as clear text (`-w` or `--password` option). However, if you prefer to use an encrypted password in a file (`-f` or `--passwordfile` option), first run the `amsfopasswd` script to encrypt the Message Queue broker clear text password to a file. Then run the `amsessiondb` script, using this file for the `-f` or `--passwordfile` option.

Use the following syntax to run the `amsessiondb` script.

```
amsessiondb [ -u username | --username username ]
[ -w password | --password password |
-f filename | --passwordfile filename ]
[ -c cachesize | --cachesize cachesize ]
[ -b dbdirectory | --dbdirectory dbdirectory ]
-a MQServerAddressList | --clusteraddress MQServerAddressList
[ -s numcleanexpiredsessions | --numcleansessions numcleanexpiredsessions ]
[ -v | --verbose ]
[ -i statsinterval | --statsInterval statsinterval ]
amsessiondb -h | --help
amsessiondb -n | --version
```

The following table describes the `amsessiondb` script arguments.

TABLE 8–6 `amsessiondb` Script Arguments

Argument	Description
-u *username* | --username *username*	User name to connect to the Message Queue broker. Specify the user you specified under 3–Add a New User in the Message Queue Server. Default is "guest".
-w *password* | --password *password*	Clear text password for the user name used to connect to the Message Queue broker. Specify the password you specified under 3–Add a New User in the Message Queue Server. Default is "guest".
-f *filename* | --passwordfile *filename*	File that contains the encrypted password for accessing the Message Queue broker. Note If you specify this option, do not specify the -w or --password option.
-c *cachesize* | --cachesize *cachesize*	Cache size in MB. Default is 8 MB.
-b *dbdirectory* | --dbdirectory *dbdirectory*	Base directory where the Berkeley DB database (`amsessions.db`) is created. Default is "sessiondb", created in the directory where you are running the `amsessiondb` script. **Note** To ensure that you have sufficient disk space where you are creating the database, allow 1 GB for each 100,000 sessions.
-a *MQServerAddressList* | --clusteraddress *MQServerAddressList*	Message Queue broker address list, in the format: *host*`1:`*port*`[,`*host*`2:`*port*`,`*host*`3:`*port*`,...]` For example: `mqsvr1:7777,mqsvr2:7777`
-s numcleanexpiredsessions | --numcleansessions numcleanexpiredsessions	Number of expired sessions to be deleted for each cleanup interval. Default is 1000.
`-v | --verbose`	Run in verbose mode. Results are sent to the standard output. Default is non-verbose mode.
-i statsinterval | --statsInterval statsinterval	Interval in seconds to print the statistics for total requests, reads, writes, and deletes to the standard output. Default is 60 seconds.
`-h | --help`	Display `amsessiondb` command usage and then exit.
`-n | --version`	Return the version of Access Manager currently installed and then exit.

The following example shows the amsessiondb script.

```
amsessiondb -u amsvrusr -f pwfile -c 128 -b sessiondb
-a host1:7777,host2:7777
```

Performance Tests With the amsessiondb Client

Performance tests with the amsessiondb client include this criteria:

- The approximate number of operations on the Berkeley DB was two times the number of authentications per second.
- Tests were conducted with the following configuration:
  - Write data – 3 Kbytes
  - Duration – 1 minute
  - Berkeley DB cache size – 28 Mbytes (default cache size in Access Manager 7.1 2006Q4 is 32 Mbytes)

The following table shows the results of the tests.

TABLE 8–7 Performance Tests With the amsessiondb Client

Disk	Notes
Normal IDE disk: 666 writes per second	Each site can support up to 300 authentications per second. Therefore, IDE disks are not recommended.
Normal 10K RPM SCSI disk on Sun Blade server: 1520 writes per second	Each site can support up to 750 authentications per second.
Seagate Cheetah 15K RPM SCSI disk: 1860 writes per second	Each site can support up to 900 authentications per second.
Sun T-300 disk array: 2700 writes per second	Each site can support up to 1300 authentications per second.
Disk using swap space in /tmp: 3300 writes per second	Each site can support up to 1600 authentications per second.

◆ ◆ ◆ CHAPTER 9

Installing and Configuring Third-Party Web Containers

Sun Java™ System Access Manager 7.1 2006Q4 supports the following third-party Web containers:

- BEA WebLogic Server 8.1 SP4

  `http://www.bea.com/products/weblogic/server/`

- IBM WebSphere Application Server 5.1.1.6

  `http://www-306.ibm.com/software/webservers/appserv/was/support/`

This chapter includes these topics:

Requirements For Using a Third-Party Web Container

The requirements to use either BEA WebLogic Server or IBM WebSphere Application Server as the Web container include:

- You must have previously obtained the Web container software from BEA or IBM.
- WebLogic Server and WebSphere Application Server are not part of the Sun Java Enterprise System (Java ES). Therefore, you must install and configure them independently of the Java ES installer.
- The Web container must be installed, configured, and running when you configure Access Manager by running the `amconfig` script. Therefore, you should be familiar with administration tasks for the Web container, such as configuring, starting, and stopping an instance.
- Access Manager requires Sun Java System Directory Server. Either install a new Directory Server using the Java ES installer, or specify an existing Directory Server when you run the installer.

General Steps For Using a Third-Party Web Container

To use a third-party Web container, follow these general steps:

1. Install and configure the Web container by following the BEA or IBM documentation.
2. Install Sun Java System Directory Server (if necessary), Access Manager with the Configure Later option, and any other Java ES components, by running the Java ES installer.
3. Start the Web container.
4. Configure Access Manager for the Web container by running the `amconfig` script with configuration parameters specified in the silent configuration input file (copy of the `amsamplesilent` file).
5. Restart the Web container.

Installing and Configuring BEA WebLogic Server 8.1 SP4

To install and configure BEA WebLogic Server 8.1 SP4, and to start and stop instances, follow the BEA documentation:

`http://e-docs.bea.com/wls/docs81/`

During installation and configuration, save the information to set the configuration variables shown in "WebLogic Application Server 8.1 SP4 Configuration Variables" on page 96 when you run the Access Manager `amconfig` script.

▼ To Install and Configure BEA WebLogic Application Server 8.1 SP4

1 Install WebLogic Application Server 8.1 SP4 and any required patches.

2 Configure WebLogic Application Server using either the Administration Console or command-line interface.

3 Start WebLogic Application Server using either the Administration Console or command-line interface.

WebLogic Application Server 8.1 SP4 Configuration Variables

The following table describes the configuration variables that you set in the `amsamplesilent` file (or copy of the file) when you run the `amconfig` script to configure Access Manager with BEA WebLogic Server 8.1 SP4 as the Web container.

TABLE 9–1 BEA WebLogic Server 8.1 SP4 Configuration Variables

Configuration Variable	Description
`WEB_CONTAINER`	Web container variable. Set to `WL8`.
`WL8_HOME`	WebLogic Server home directory. Default: `/usr/local/bea`
`WL8_PROJECT_DIR`	WebLogic Server project directory. Default: `user_projects`
`WL8_DOMAIN`	WebLogic Server domain name. Default: `mydomain`
`WL8_CONFIG_LOCATION`	Parent directory of the location of the WebLogic Server start script.
`WL8_SERVER`	WebLogic Server server name. Default: `myserver`
`WL8_INSTANCE`	WebLogic Server instance name. Default: `/usr/local/bea/weblogic81 ($WL8_HOME/weblogic81)`
`WL8_PROTOCOL`	WebLogic Server protocol. Default: `http`
`WL8_HOST`	WebLogic Server host name. Default: Host name of the server
`WL8_PORT`	WebLogic Server port. Default: 7001
`WL8_SSLPORT`	WebLogic Server SSL port. Default: 7002
`WL8_ADMIN`	WebLogic Server administrator. Default: `"weblogic"`
`WL8_PASSWORD`	WebLogic Server administrator password.
`WL8_JDK_HOME`	WebLogic Server JDK home directory. Default: `/usr/local/bea/jdk142_04 ($WL8_HOME/jdk142_04)`

Installing and Configuring IBM WebSphere Application Server 5.1.1.6

To install and configure IBM WebSphere Application Server 5.1.1.6, and to start and stop instances, follow the IBM documentation:

`http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1/index.jsp`

During installation and configuration, save the information to set the configuration variables shown in "IBM WebSphere Application Server Configuration Variables" on page 98 when you run the Access Manager `amconfig` script.

▼ To Install and Configure IBM WebSphere Application Server

1 **Install WebSphere Application Server and any required patches.**

2 **Verify that the WebSphere Application Server installation was successful.**

a. **Make sure the** `server.xml` **file exists in the following directory:**

```
/opt/WebSphere/AppServer/config/cells/cell-name/noes/node-name/servers/server1
```

b. **Start the server with the** `startServer.sh` **utility. For example:**

```
# /opt/WebSphere/AppServer/bin/startServer.sh server1
```

c. **In a Web browser, use the following URL to view the sample Web application:**

`http://`*fqdn*`:`*port*`/snoop`

Where *fqdn* and *port* specify the server name and port number.

3 **After you have verified a successful installation, stop the server using the** `stopServer.sh` **utility. For example:**

```
# /opt/WebSphere/AppServer/bin/stopServer.sh server1
```

4 **Install any required patches using the** `updateWizard.sh` **utility.**

5 **Restart WebSphere Application Server using the** `startServer.sh` **utility.**

IBM WebSphere Application Server Configuration Variables

The following table describes the configuration variables that you set in the `amsamplesilent` file (or copy of the file) when you run the `amconfig` script to configure Access Manager with WebSphere Application Server as the Web container.

TABLE 9–2 IBM WebSphere Application Server 5.1 Configuration Variables

Variable	Description
`WEB_CONTAINER`	Web container variable. Set to `WAS5`.
`WAS51_HOME`	WebSphere home directory. Default: `/opt/WebSphere/AppServer`
`WAS51_JDK_HOME`	WebSphere JDK home directory. Default: `/opt/WebSphere/AppServer/java`

TABLE 9–2 IBM WebSphere Application Server 5.1 Configuration Variables *(Continued)*

Variable	Description
WAS51_CELL	WebSphere cell. Default: host-name value
WAS51_NODE	WebSphere node name. Default: host name of the server where WebSphere is installed. Default: hostname value
WAS51_INSTANCE	WebSphere instance name. Default: server1
WAS51_PROTOCOL	WebSphere protocol. Default: http
WAS51_HOST	WebSphere host name. Default: Hostname of the server
WAS51_PORT	WebSphere port. Default: 9080
WAS51_SSLPORT	WebSphere SSL port. Default: 9081
WAS51_ADMIN	WebSphere administrator. Default: "admin"
WAS51_ADMINPORT	WebSphere administrator port. Default: 9090

Installing Access Manager and Other Java ES Components

Run the Java ES installer to install these components:

- Sun Java System Directory Server. Either install a new Directory Server, or use an existing Directory Server, if you prefer.
- Access Manager with the Configure Later option.
- Other Java ES components as needed. For example, if you are configuring Access Manager for session failover, install Sun Java System Message Queue.

For information about running the installer, see *Sun Java Enterprise System 2006Q4 Installation Guide for UNIX*.

Configuring Access Manager Using the amconfig Script

To configure or reconfigure an Access Manager for a third-party Web container, set variables in a copy of the amsamplesilent file and run the amconfig script.

▼ To Configure Access Manager Using the amconfig Script

1 **Login as (or become) superuser (root).**

2. **Copy the `amsamplesilent` file and rename the file to describe the new instance you want to configure. For example, if you plan to create a new Access Manager instance for WebLogic Application Server, you might rename the file to `am_weblogic_server`.**

3. **Set the variables in the `am_weblogic_server` file to configure (or reconfigure) the Access Manager instance. For example:**

```
AM_REALM=enabled
DEPLOY_LEVEL=1
NEW_INSTANCE=false
WEB_CONTAINER=WAS5 # WebLogic Application Server is the web container
DIRECTORY_MODE=4  # Directory Server is provisioned with user data
AM_ENC_PW=password-encryption-key-value
...
```

Caution – In a multiple server deployment that shares the same Directory Server, all Access Manager instances must use the same value for the password encryption key. Set the `AM_ENC_PWD` variable in the copy of the `amsamplesilent` file with the same encryption key value used for other instances before you run the `amconfig` script.

4. **Run the `amconfig` script.**

   For example, on Solaris systems with Access Manger installed in the default directory, run `amconfig` using the new `am_weblogic_server` file as the configuration input file:

```
# cd /opt/SUNWam/bin/
# ./amconfig -s ./am_weblogic_server
```

   The `amconfig`script reads the variables in the `am_weblogic_server` file and then runs in silent mode (`-s` option) to configure Access manager for WebLogic Application Server Web container.

   For more information about the `amsamplesilent` file and running the `amconfig` script, see Chapter 2Chapter 2.

5. **In case you might need to reconfigure or uninstall this instance later, save the new `am_weblogic_server` file.**

6. **Restart the Web container.**

◆ ◆ ◆ CHAPTER 10

Configuring Access Manager in SSL Mode

Using Secure Socket Layer (SSL) with simple authentication guarantees confidentiality and data integrity. To enable Access Manager in SSL, mode you would typically:

- Configure Access Manager with a secure web container
- Configure Access Manager to a secure Directory Server

Configuring Access Manager With a Secure Sun Java Enterprise System Web Server

To configure Access Manager in SSL mode with Web Server, see the following steps:

▼ To Configure a Secure Web Server

1 **In the Access Manager console, go to the Service Configuration module and select the Platform service. In the Server List attribute, remove the `http://` protocol, and add the `https://` protocol. Click Save.**

Note – Be sure to click Save. If you don't, you will still be able to proceed with the following steps, but all configuration changes you have made will be lost and you will not be able to log in as administrator to fix it.

Steps 2 through 24 describe the Web Server.

2 **Log on to the Web Server console. The default port is 8888.**

3 **Select the Web Server instance on which Access Manager is running, and click Manage.**

This displays a pop-up window explaining that the configuration has changed. Click OK.

4 **Click on the Apply button located top right corner of the screen.**

5. **Click Apply Changes.**
   The Web Server should automatically restart. Click OK to continue.

6. **Stop the selected Web Server instance.**

7. **Click the Security Tab.**

8. **Click on Create Database.**

9. **Enter the new database password and click OK.**
   Ensure that you write down the database password for later use.

10. **Once the Certificate Database has been created, click on Request a Certificate.**

11. **Enter the data in the fields provided in the screen.**
    The Key Pair Field Password field is the same as you entered in Step 9. In the location field, you will need to spell out the location completely. Abbreviations, such as CA, will not work. All of the fields must be defined. In the Common Name field, provide the hostname of your Web Server.

12. **Once the form is submitted, you will see a message such as:**

```
--BEGIN CERTIFICATE REQUEST---

afajsdllwqeroisdaoi234rlkqwelkasjlasnvdknbslajowijalsdkjfalsdflasdf

alsfjawoeirjoi2ejowdnlkswnvnwofijwoeijfwiepwerfoiqeroijeprwpfrwl

--END CERTIFICATE REQUEST--
```

13. **Copy this text and submit it for the certificate request.**
    Ensure that you get the Root CA certificate.

14. **You will receive a certificate response containing the certificate, such as:**

```
--BEGIN CERTIFICATE---

afajsdllwqeroisdaoi234rlkqwelkasjlasnvdknbslajowijalsdkjfalsdflasdf

alsfjawoeirjoi2ejowdnlkswnvnwofijwoeijfwiepwerfoiqeroijeprwpfrwl

--END CERTIFICATE---
```

15. **Copy this text into your clipboard, or save the text into a file.**

16. **Go to the Web Server console and click on Install Certificate.**

17. **Click on Certificate for this Server.**

18. **Enter the Certificate Database password in the Key Pair File Password field.**

19. **Paste the certificate into the provided text field, or check the radio button and enter the filename in the text box. Click Submit.**

    The browser will display the certificate, and provide a button to add the certificate.

20. **Click Install Certificate.**

21. **Click Certificate for Trusted Certificate Authority.**

22. **Install the Root CA Certificate in the same manner described in steps 16 through 21.**

23. **Once you have completed installing both certificates, click on the Preferences tab in the Web Server console.**

24. **Select Add Listen Socket if you wish to have SSL enabled on a different port. Then, select Edit Listen Socket.**

25. **Change the security status from Disabled to Enabled, and click OK to submit the changes, click Apply and Apply Changes.**

    Steps 26–29 apply to Access Manager.

26. **Open the** `AMConfig.properties` **file. By default, the location of this file is** `etc/opt/SUNWam/config`.

27. **Replace all of the protocol occurrences of** `http://` **to** `https://`, **except for the Web Server Instance Directory. This is also specified in** `AMConfig.properties`**, but must remain the same.**

28. **Save the** `AMConfig.properties` **file.**

29. **In the Web Server console, click the ON/OFF button for the Access Manager hosting web server instance.**

    The Web Server displays a text box in the Start/Stop page.

30. **Enter the Certificate Database password in the text field and select Start.**

Configuring Access Manager with a Secure Sun Java System Application Server

Setting up Access Manager to run on an SSL-enabled Application server is a two-step process. First, secure the Application Server instance to the installed Access Manager, then configure Access Manager itself.

Setting Up Application Server 6.2 With SSL

This section describes the steps to set up Application Server 6.2 in SSL mode.

▼ To Secure the Application Server Instance

1. **Log into the Sun Java System Application Server console as an administrator by entering the following address in your browser:**

   ```
   http://fullservername:port
   ```

   The default port is 4848.

2. **Enter the username and password you entered during installation.**

3. **Select the Application Server instance on which you installed (or will install) Access Manager. The right frame displays that the configuration has changed.**

4. **Click Apply Changes.**

5. **Click Restart. The Application Server should automatically restart.**

6. **In the left frame, click Security.**

7. **Click the Manage Database tab.**

8. **Click Create Database, if it is not selected.**

9. **Enter the new database password and confirm, then click the OK button. Make sure that you write down the database password for later use.**

10. **Once the Certificate Database has been created, click the Certificate Management tab.**

11. **Click the Request link, if it is not selected.**

12 Enter the following Request data for the certificate

a. Select it if this is a new certificate or a certificate renewal. Many certificates expire after a specific period of time and some certificate authorities (CA) will automatically send you renewal notification.

b. Specify the way in which you want to submit the request for the certificate.

If the CA expects to receive the request in an E-mail message, check CA E-mail and enter the E-mail address of the CA. For a list of CAs, click List of Available Certificate Authorities.

If you are requesting the certificate from an internal CA that is using the Certificate Server, click CA URL and enter the URL for the Certificate Server. This URL should point to the certificate server's program that handles certificate requests.

c. Enter the password for your key-pair file (this is the password you specified in step 9).

d. Enter the following identification information:

Common Name. The full name of the server including the port number.

Requestor Name. The name of the requestor.

Telephone Number. The telephone number of the requestor

Common Name . The fully qualified name of the Sun Java System Application Server on which the digital certificate will be installed.

E-mail Address. The E-mail address of the administrator.

Organization Name. The name of your organization. The certificate authority may require any host names entered in this attribute belong to a domain registered to this organization.

Organizational Unit Name. The name of your division, department, or other operational unit of your organization.

Locality Name (city). The name of your city or town.

State Name. The name of the state or province in which your organization operates if your organization is in the United States or Canada, respectively. Do not abbreviate.

Country Code. The two-letter ISO code for your country. For example, the code for the United States is `US`.

13 Click the OK button. A message will be displayed, for example:

```
--BEGIN NEW CERTIFICATE REQUEST---
afajsdllwqeroisdaoi234rlkqwelkasjlasnvdknbslajowijalsdkjfalsdfla
alsfjawoeirjoi2ejowdnlkswnvnwofijwoeijfwiepwerfoiqeroijeprwpfrwl
--END NEW CERTIFICATE REQUEST--
```

14 Copy all of this text to a file and click OK. Make sure that you get the Root CA certificate.

15 Select a CA and follow the instructions on that authority's web site to get a digital certificate. You can get the certificate from CMS, Verisign or Entrust.net

16 After you receive your digital certificate from the certificate authority, you can copy the text into your clipboard, or save the text into a file.

17 Go to the Application Server console and click on the Install link.

18 Select Certificate For This Server.

19 Enter the Certificate Database password in the Key Pair File Password field.

20 Paste the certificate into the provided text field, Message text (with headers), or enter the filename in the Message that is in this file text box. Select the appropriate radio button.

21 Click OK button. The browser displays the certificate, and provides a button to add the certificate.

22 Click Add Server Certificate.

23 Install the Root CA Certificate in the same manner described above. However, select Certificate for Trusted Certificate Authority.

24 Once you have completed installing both certificates, expand the HTTP Server node in the left frame

25 Select HTTP Listeners under HTTP Server.

26 Select `http-listener-1`. The browser displays the socket information.

27 Change the value of the port used by `http-listener-1` from the value entered while installing application server, to a more appropriate value such as 443.

28 Select SSL/TLS Enabled.

29 Select Certificate Nickname.

30 Specify the Return server. This should match the common name specified in Step 12.

31 Click Save.

32 Select the Application Server instance on which you will install the Access Manager software. The right frame shows that the configuration has changed.

33 Click Apply Changes.

34 Click Restart. The application server should automatically restart.

Configuring Application Server 8.1 With SSL

The basic steps to configure Application Server 8.1 with SSL are as follows. See the Application Server 8.1 documentation for detailed instructions.

1. Create a secure port on the Application server through the Application Server Administration console. For more information, see "Configuring Security" in the Sun Java System Application Server Enterprise Edition 8.1 Administration Guide at the following location:

   `http://docs.sun.com/app/docs/coll/1310.1`

2. Verify that the certificate authority (CA) that trusts the server's certificate is present in the web container's trust database. Then, obtain and install a server certificate for the web container. For more information, see "Working with Certificates and SSL" in the Sun Java System Application Server Enterprise Edition 8.1 Administration Guide at the following location:

   `http://docs.sun.com/app/docs/coll/1310.1`

3. Restart the web container.

Configuring Access Manager in SSL Mode

This section describes the steps to configure Access Manager in SSL mode. Before you set up SSL for Access Manager, make sure that you configured the web container for your deployment.

▼ To Configure Access Manager in SSL Mode

1 In the Access Manager console, go to the Service Configuration module and select the Platform service. In the Server List attribute, add the same URL with the HTTPS protocol and an SSL-enabled port number. Click Save.

Note – If a single instance of Access Manager is listening on two ports (one in HTTP and one in HTTPS) and you try to access Access Manager with a stalled cookie, Access Manager will become unresponsive. This is not a supported configuration.

---

2 Open the `AMConfig.properties` file from the following default location:

`/etc/opt/SUNWam/config.`

3 Replace all of the protocol occurrences of `http://` to `https://` and change the port number to an SSL-enabled port number.

4 Save the `AMConfig.properties` file.

5 Restart the Application Server.

Configuring AMSDK with a Secure BEA WebLogic Server

The BEA WebLogic Server must first be installed and configured as a web container before you configure it with the AMSDK in SSL. For installation instructions, see the BEA WebLogic server documentation. To configure WebLogic as a web container for Access Manager, see Chapter 2, Access Manager 7.1 2006Q4 Configuration Scripts.

▼ To Configure a Secure WebLogic Instance

1. **Create a domain using the quick start menu**
2. **Go to the WebLogic installation directory and generate the certificate request.**
3. **Apply for the server certificate using the CSR text file to a CA.**
4. **Save the approved certificate in to a text file. For example,** `approvedcert.txt`.
5. **Load the Root CA in** `cacerts` **by using the following commands:**

   ```
   cd jdk141_03/jre/lib/security/
   ```

   ```
   jdk141_03/jre/bin/keytool -keystore cacerts -keyalg RSA -import -trustcacerts -alias
   "<alias name>" -storepass changeit -file /opt/bea81/cacert.txt
   ```

6. **Load the Server certificate by using the following command:**

   ```
   jdk141_03/jre/bin/keytool -import -keystore <keystorename> -keyalg RSA -import
   -trustcacerts -file approvedcert.txt -alias "mykey"
   ```

7. **Login to WebLogic console with your username and password.**
8. **Browse to the following location:**

   yourdomain> Servers> myserver> Configure Keystores
9. **Select Custom Identity and then Java Standard Trust**
10. **Enter the keystore location. For example,** `/opt/bea81/keystore`.
11. **Enter Keystore Password and Keystore Pass Phrase. For example:**

    Keystore Password: `JKS/Java Standard Trust` (for WL 8.1 it is only JKS)

    Key Store Pass Phrase: `changeit`
12. **Review the SSL Private Key Settings Private Key alias and password.**

Note – You must use the full strength SSL licence or SSL startup will fail

13 **In Access Manager, the following parameters in** `AmConfig.properties` **are automatically configured during installation. If they are not, you can edit them appropriately:**

```
com.sun.identity.jss.donotInstallAtHighestPriority=true [ this is not
 required for AM 6.3 and above]
com.iplanet.security.SecureRandomFactoryImpl=com.iplanet.am.util.SecureRandomFactoryImpl
com.iplanet.security.SSLSocketFactoryImpl=netscape.ldap.factory.JSSESocketFactory
com.iplanet.security.encryptor=com.iplanet.services.util.JCEEncryption
```

If your JDK path is the following:

```
com.iplanet.am.jdk.path=/usr/jdk/entsys-j2se
```

then use the keytool utility to import the root CA in the certificate database. For example:

```
/usr/jdk/entsys-j2se/jre/lib/security
/usr/jdk/entsys-j2se/jre/bin/keytool -keystore cacerts
-keyalg RSA -import -trustcacerts -alias "machinename" -storepass changeit -file
/opt/bea81/cacert.txt
```

The keytool utility is located in the following directory:

```
/usr/jdk/entsys-j2se/jre/bin/keytool
```

14 **Remove** `-D"java.protocol.handler.pkgs=com.iplanet.services.comm"` **from the Access Manager** `amadmin` **command line utility.**

15 **Configure Access Manager in SSL Mode. For more information, see** **"Configuring Access Manager in SSL Mode" on page 107****.**

Configuring AMSDK with a Secure IBM WebSphere Application Server

The IBM WebSphere Server must first be installed and configured as a web container before you configure it with the AMSDK in SSL. For installation instructions, see the WebSphere server documentation. To configure WebLogic as a web container for Access Manager, see Chapter 2, Access Manager 7.1 2006Q4 Configuration Scripts.

▼ To Configure a Secure WebSphere Instance

1. **Start `ikeyman.sh`, located in the Websphere `/bin` directory.**
2. **From the Signer menu, import the certification authority's (CA) certificate.**
3. **From the Personal Certs menu, generate the CSR.**
4. **Retrieve the certificate created in the previous step.**
5. **Select Personal Certificates and import the server certificate.**
6. **From the WebSphere console, change the default SSL settings and select the ciphers.**
7. **Set the default IBM JSSE SSL provider.**
8. **Enter the following command to import the Root CA certificate from the file you just created into application server JVM Keystore:**

   ```
   $ appserver_root-dir/java/bin/ keytool -import -trustcacerts -alias cmscacert
   -keystore ../jre/lib/security/cacerts -file
   /full_path_cacert_filename.txt
   ```

   app-server-root-dir is the root directory for the application server and *full_path_cacert_filename.txt* is the full path to the file containing the certificate.

9. **In Access Manager, update the following parameters in `AmConfig.properties` to use JSSE:**

   ```
   com.sun.identity.jss.donotInstallAtHighestPriority=true
   com.iplanet.security.SecureRandomFactoryImpl=com.iplanet.
   am.util.SecureRandomFactoryImpl
   com.iplanet.security.SSLSocketFactorImpl=netscape.ldap.factory.
   JSSESocketFactory
   com.iplanet.security.encyptor=com.iplanet.services.unil.JCEEncryption
   ```

10. **Configure Access Manager in SSL Mode. For more information, see "Configuring Access Manager in SSL Mode" on page 107.**

Configuring Access Manager to Directory Server in SSL Mode

To provide secure communications over the network, Access Manager includes the LDAPS communications protocol. LDAPS is the standard LDAP protocol, but it runs on top of the Secure Sockets Layer (SSL). In order to enable SSL communication, you must first configure the Directory Server in SSL mode and then connect Access Manager to Directory Server. The basic steps are as follows:

1. Obtain and install a certificate for your Directory Server, and configure the Directory Server to trust the certification authority's (CA) certificate
2. Turn on SSL in your directory.
3. Configure the authentication, policy and platform services to connect to an SSL-enabled Directory Server.
4. Configure Access Manager to securely connect to the Directory Server backend.

Configuring Directory Server in SSL Mode

In order to configure the Directory Server in SSL mode, you must obtain and install a server certificate, configure the Directory Server to trust the CA's certificate and enable SSL. Detailed instructions on how to complete these tasks are included in Chapter 11, "Managing Authentication and Encryption" in the *Directory Server Administration Guide*. This document can be found in the following location:

http://docs.sun.com/coll/DirectoryServer_04q2
(`http://docs.sun.com/coll/DirectoryServer_04q2`)

If your Directory Server is already SSL-enabled, go to the next section for details on connecting Access Manager to Directory Server.

Connecting Access Manager to the SSL-enabled Directory Server

Once the Directory Server has been configured for SSL mode, you need to securely connect Access Manager to the Directory Server backend.

▼ To Connect Access Manager to Directory Server

1 In the Access Manager Console, go to the LDAP Authentication service in the Service Configuration module.

a. Change the Directory Server port to the SSL port.

b. Select the Enable SSL Access to LDAP Server attribute.

2 Go to the Membership Authentication service in the Service Configuration module.

a. Change the Directory Server port to the SSL port.

b. Select the Enable SSL Access to LDAP Server attribute.

3. **Go to the Policy Configuration service located in Service Configuration.**

   a. **Change the Directory Server port to the SSL port.**

   b. **Select the Enable LDAP SSL attribute.**

4. **Open the** `serverconfig.xml` **in a text editor. The file is in the following location:**

   `/etc/opt/SUNWam/config`

   a. **In the** `<Server>` **element, change the following values:**

      `port` - enter the port number of the secure port to which Access Manager listens (636 is the default).

      `type`- change SIMPLE to SSL.

   b. **Save and close** `serverconfig.xml`**.**

5. **Open the** `AMConfig.properties` **file from the following default location:**

   /etc/opt/`SUNWam/config.`

   Change the following properties:

   a. `com.iplanet.am.directory.port = 636` **(if using the default)**

   b. `ssl.enabed = true`

   c. **Save** `AMConfig.properties`**.**

6. **Restart the server**

◆ ◆ ◆ CHAPTER 11

Deploying the Client SDK

The Access Manager Client SDK subcomponent allows you to implement stand-alone applications that can access an Access Manager server to use services such as authentication, SSO, authorization, auditing, logging, user management, and SAML.

Requirements for an Access Manager Client SDK Deployment

Requirements for an Access Manager Client SDK deployment include:

- The Access Manager Client SDK must be installed in one of these Web containers:
  - Sun Java System Application Server
  - Sun Java System Web Server
  - BEA WebLogic Server
  - IBM WebSphere Application Server

  For the specific versions supported of each Web container, see the *Sun Java System Access Manager 7.1 2006Q4 Release Notes*.
- The Access Manager Client SDK must use the same password encryption key as the Access Manager server instances in the deployment.

Installing and Configuring the Access Manager Client SDK

Installing and configuring (or reconfiguring) the Access Manager Client SDK involves running the Java ES installer and the `amconfig` script. One or more Access Manager full server instances must be installed and running in the deployment.

▼ To Install and Configure the Access Manager Client SDK

1 **Log in as or become superuser (`root`) on the server where you want to deploy the Access Manager Client SDK.**

2 **Get the Java ES installer. For information, see "Getting the Java ES Installer" on page 21.**

3 **Install the Access Manager Web container that you plan to use for the Access Manager Client SDK:**

- Web Server or Application Server: Install using the Java ES installer.
- BEA WebLogic Server or IBM WebSphere Application Server: See Chapter 9.

4 **Install the Access Manager Client SDK by running the Java ES installer with either the Configure Now or Configure Later option. On the installer Component Selection page, check Client SDK.**

If you are using the Configure Now option, see "Access Manager Client SDK Configuration Variables" on page 115 for the values that you must specify during installation.

5 **If you specified the Configure Later option during the previous step, or if you need to reconfigure the Client SDK, run the `amconfig` script as follows:**

a. **Copy the `amsamplesilent` file and set the configuration variables in the new file. For example, you might name the new file as `ClientSDK_config`. See "Access Manager Client SDK Configuration Variables" on page 115 for the variables that you need to set.**

b. **Run the `amconfig` script using the new configuration file. For example, on a Solaris system with Access Manager installed in the default directory:**

```
# cd /opt/SUNWam/bin
# ./amconfig -s ./ClientSDK_config
```

6 **Restart the Web container for the Access Manager Client SDK.**

Example 11–1 Access Manager Client SDK Sample Configuration File

```
DEPLOY_LEVEL=9
APPLICATION_USER=username
APPLICATION_PASSWD=application-user-password
AM_ENC_SECRET=am-secret-password
AM_ENC_LOCAL=am-password-encryption-key-used-by-the-Access-Manager-server
DEBUG_LEVEL=error
DEBUG_DIR=/var/opt/SUNWam/logs
```

Access Manager Client SDK Configuration Variables

TABLE 11–1 Access Manager Client SDK Configuration Variables

Variable	Description
DEPLOY_LEVEL	DEPLOY_LEVEL=9 - Configure (or reconfigure) the Access Manager Client SDK. DEPLOY_LEVEL=19 - Uninstall the Access Manager Client SDK.
SERVER_NAME, SERVER_HOST, SERVER_PORT, ADMIN_PORT,	Corresponding values that used for the full Access Manager server installation.
SERVER_DEPLOY_URI, CONSOLE_DEPLOY_URI	**Important** You must set the password encryption key (AM_ENC_PWD) to the same value used by the Access Manager server instance.
ADMINPASSWD, AMLDAPUSERPASSWD, COOKIE_DOMAIN, AM_ENC_PWD	
DS_HOST, DS_DIRMGRPASSWD, and ROOT_SUFFIX	Corresponding Directory Server values that were used for the full Access Manager server installation.
NEW_OWNER and NEW_GROUP	Runtime user and group that will own the Web container processes on which the Access Manager Client SDK will be deployed.
PAM_SERVICE_NAME	If the Access Manager Client SDK host is running the Linux OS, set to "password".
WEB_CONTAINER Web container configuration variables	Web container on which the Access Manager Client SDK is or will be deployed. For example, if the Web container is Sun Java System Web Server 7 2006Q4, set WEB_CONTAINER=WS. Set the configuration variables for the Web container specified by WEB_CONTAINER. For more information, see "Web Container Configuration Variables" on page 33.
DISTAUTH_PROTOCOL	Protocol (http or https) used by the Web container instance on which the Access Manager Client SDK is or will be deployed. Default: http
DISTAUTH_HOST	Fully qualified host name where the Access Manager Client SDK is located. Default: distAuth_sample.com
DISTAUTH_PORT	Port on DISTAUTH_HOST on which the Access Manager Client SDK has been or will be deployed. Default: 80
APPLICATION_USER	User name for the application. Default: username
APPLICATION_PASSWD	Password of the user for the application. Default: none
AM_ENC_LOCAL	Password encryption key. Default: none

TABLE 11–1 Access Manager Client SDK Configuration Variables *(Continued)*

Variable	Description
DEBUG_LEVEL	Level for the debug service. Values can be: error, warning, or message. Default: error
DEBUG_DIR	Directory where the debug files will be created. Default: Solaris systems: /var/opt/SUNWam/logs Linux systems: /var/opt/sun/identity/logs
BASEDIR	Base directory where the Access Manager Client SDK is installed.
CONSOLE_HOST, CONSOLE_PORT, and CONSOLE_PROTOCOL	Corresponding values for the host on which the Access Manager console has been deployed.
CONSOLE_REMOTE	TBD. The default value is false.
CLIENT_DEPLOY_URI	Deployment URI that will be used on the local host by the Access Manager Client SDK. The default value is /amclient.

Accessing the Client SDK

To access the Client SDK, use the following URL in your browser:

client_sdk_protocol://*client_sdk_server*: *client_sdk_port*/*client_sdk_deploy_URI*/UI/Login

Where:

client_sdk_protocol	Protocol (http or https) used by the Web container instance on which the Client SDK is deployed.
client_sdk_server_host	Fully qualified host name of the Client SDK server.
client_sdk_server_port	Port for the host name of the Client SDK.
client_sdk_deploy_URI	Deployment URI prefix for the Client SDK. The default value is /amclient.

For example:

```
https://clientserver.example.com:80/amclient
```

◆ ◆ ◆ CHAPTER 12

Deploying a Distributed Authentication UI Server

The Distributed Authentication UI subcomponent provides for secure, distributed authentication across two firewalls in an Access Manager deployment. You install the Distributed Authentication UI subcomponent on one or more servers within the non-secure (DMZ) layer of an Access Manager deployment. This subcomponent acts as an authentication interface between end users and the Access Manager instances behind the second firewall, thus eliminating the exposure of the Access Manager service URLs to the end users. A Distributed Authentication UI server does not run Access Manager; it exists only to provide the authentication interface between end users and an Access Manager instance. This chapter describes these topics:

Distributed Authentication UI Server Overview

Requirements for a Distributed Authentication UI Server Deployment

Requirements for a Distributed Authentication UI server deployment include:

- The Distributed Authentication UI server must be installed in one of these Web containers:
  - Sun Java System Application Server
  - Sun Java System Web Server
  - BEA WebLogic Server
  - IBM WebSphere Application Server

For the specific versions supported of each Web container, see the *Sun Java System Access Manager 7.1 2006Q4 Release Notes*.

- A Distributed Authentication UI server must use the same password encryption key as the Access Manager server instances in the deployment.

Several other considerations for a Distributed Authentication UI server include:

- If you are deploying multiple Distributed Authentication UI servers behind a load balancer, stickiness is not required for the load balancer to talk to only one Distributed Authentication UI server for authentication process completion.
- The HTTP Basic and MSISDN authentication modules are not supported through the Distributed Authentication UI.

Distributed Authentication UI Server Deployment Scenario

The following figure shows a Distributed Authentication UI server deployment scenario.

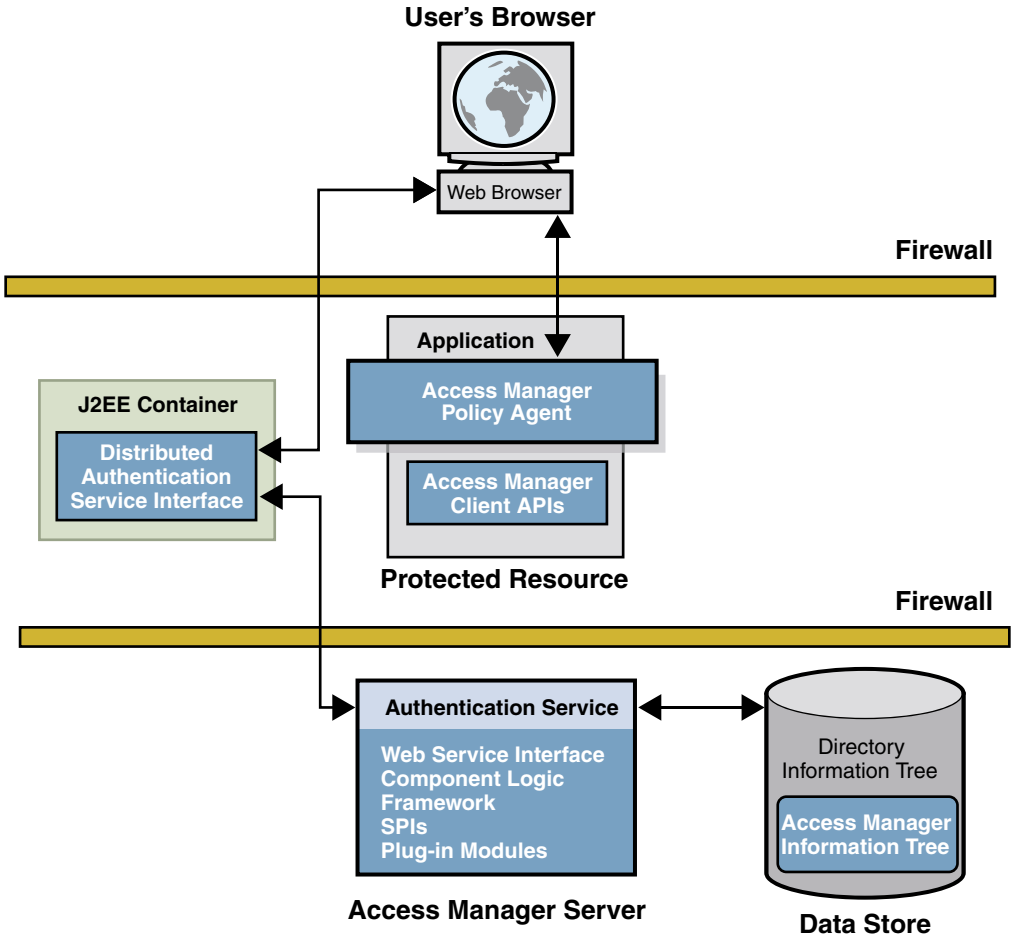

FIGURE 12–1 Distributed Authentication UI Server Deployment Scenario

Flow for a Distributed Authentication End-User Request

In a typical deployment scenario using one or more Distributed Authentication UI servers, an end-user request follows this flow:

1. An end user sends an HTTP or HTTPS request from a Web browser to access a protected resource.
2. If the request does not have a cookie containing an SSO token, the Access Manager policy agent issues a redirect to its authentication URL, which is the URL of the Distributed Authentication UI server in the DMZ (usually through a load balancer).

3. The end user follows the redirect and sends a request to the Distributed Authentication UI server.
4. The Distributed Authentication UI server communicates the request to an Access Manager instance behind the second firewall to determine the appropriate authentication method.
5. The Access Manager instance determines the appropriate authentication method and then returns the presentation framework to the Distributed Authentication UI server.
6. Using the information from the Access Manager instance, the Distributed Authentication UI server returns a login page to the user's Web browser.
7. The end user replies with the login credentials (such as user name and password) to the Distributed Authentication UI server.
8. The Distributed Authentication UI server uses the Access Manager Client SDK to send the end user's credentials to the Access Manager instance behind the second firewall.
9. Access Manager tries to authenticate the end user using the appropriate authentication method:
   - If the authentication is successful, Access Manager returns the SSO token, and the Distributed Authentication UI server redirects the end user to the protected resource.
   - If the authentication is not successful, Access Manager returns the appropriate error information.

Installing and Configuring a Distributed Authentication UI Server

Installing and configuring (or reconfiguring) a Distributed Authentication UI server involves running the Java ES installer and the `amconfig` script on the Distributed Authentication UI server. One or more Access Manager full server instances must be installed and running in the deployment.

▼ To Install and Configure a Distributed Authentication UI Server

1 Log in as or become superuser (`root`) on the Distributed Authentication UI server.

2 Get the Java ES installer. For information, see "Getting the Java ES Installer" on page 21.

3 Install the Access Manager Web container that you plan to use for the Distributed Authentication UI server:

- Web Server or Application Server: Install using the Java ES installer.
- BEA WebLogic Server or IBM WebSphere Application Server: See the respective BEA or IBM product documentation for installation instructions.

4. **Install the Distributed Authentication UI subcomponent by running the Java ES installer with either the Configure Now or Configure Later option. On the installer Component Selection page, check Distributed Authentication.**

   If you are using the Configure Now option, see "Distributed Authentication UI Server Configuration Variables" on page 122 for the values that you must specify during installation.

5. **If you specified the Configure Later option during the previous step, or if you need to reconfigure the Distributed Authentication UI server, run the `amconfig` script as follows:**

   a. **Copy the `amsamplesilent` file and set the configuration variables in the new file. For example, you might name the new file as `DistAuth_config`. See "Distributed Authentication UI Server Configuration Variables" on page 122 for the variables that you need to set.**

   b. **Run the `amconfig` script using the new configuration file. For example, on a Solaris system with Access Manager installed in the default directory:**

   ```
   # cd /opt/SUNWam/bin
   # ./amconfig -s ./DistAuth_config
   ```

6. **Restart the Web container on the Distributed Authentication UI server.**

Example 12–1 Distributed Authentication UI Server Sample Configuration File

```
DEPLOY_LEVEL=8
DISTAUTH_PROTOCOL=http
DISTAUTH_HOST=distauth.example.com
DISTAUTH_PORT=80
APPLICATION_USER=username
APPLICATION_PASSWD=application-user-password
AM_ENC_SECRET=am-secret-password
AM_ENC_LOCAL=am-password-encryption-key-used-by-the-Access-Manager-server
DEBUG_LEVEL=error
DEBUG_DIR=/var/opt/SUNWam/logs
```

Distributed Authentication UI Server Configuration Variables

TABLE 12–1 Distributed Authentication UI Server Configuration Variables

Variable	Description
DEPLOY_LEVEL	DEPLOY_LEVEL=8 - Configure (or reconfigure) a Distributed Authentication UI server. DEPLOY_LEVEL=18 - Uninstall a Distributed Authentication UI server.
SERVER_HOST, SERVER_PORT SERVER_DEPLOY_URI, CONSOLE_DEPLOY_URI ADMINPASSWD, AMLDAPUSERPASSWD, COOKIE_DOMAIN, AM_ENC_PWD	Corresponding values that used for the full Access Manager server installation. **Important** You must set the password encryption key (AM_ENC_PWD) to the same value used by the Access Manager server instance.
DS_HOST, DS_DIRMGRPASSWD, and ROOT_SUFFIX	Corresponding Directory Server values that were used for the full Access Manager server installation.
NEW_OWNER and NEW_GROUP	Runtime user and group that will own the Web container processes on which the Distributed Authentication UI server will be deployed.
PAM_SERVICE_NAME	If the Distributed Authentication UI server host is running the Linux OS, set to password.
WEB_CONTAINER Web container configuration variables	Web container on which the Distributed Authentication UI server is or will be deployed. For example, if the Web container is Sun Java System Web Server 7 2006Q4, set WEB_CONTAINER=WS. Set the configuration variables for the Web container specified by WEB_CONTAINER. For more information, see "Web Container Configuration Variables" on page 33.
DISTAUTH_PROTOCOL	Protocol (http or https) used by the Web container instance on which the Distributed Authentication UI server is or will be deployed. Default: http
DISTAUTH_HOST	Fully qualified host name where the Distributed Authentication UI server is located. Default: distAuth_sample.com
DISTAUTH_PORT	Port on DISTAUTH_HOST on which the Distributed Authentication UI server has been or will be deployed. Default: 80
APPLICATION_USER	User name for the application. Default: username
APPLICATION_PASSWD	Password of the user for the application. Default: none

TABLE 12–1 Distributed Authentication UI Server Configuration Variables *(Continued)*

Variable	Description
`AM_ENC_SECRET`	Password encryption secret key from the server. Default: none
`AM_ENC_LOCAL`	Password encryption key. Default: none
`DEBUG_LEVEL`	Level for the debug service. Values can be: `error`, `warning`, or `message`. Default: `error`
`DEBUG_DIR`	Directory where the debug files will be created. Default: Solaris systems: `/var/opt/SUNWam/logs` Linux systems: `/var/opt/sun/identity/logs`
`BASEDIR`	Base directory where the Distributed Authentication UI server was installed.
`CONSOLE_HOST`, `CONSOLE_PORT`, and `CONSOLE_PROTOCOL`	Corresponding values for the host on which the Access Manager console has been deployed.
`CONSOLE_REMOTE`	TBD
`DISTAUTH_DEPLOY_URI`	Deployment URI that will be used on the local host by the Distributed Authentication UI server. The default value is `/amdistauth`.

Accessing the Distributed Authentication User Interface

To access the Distributed Authentication UI, use the following URL in your browser:

DA_server_protocol`://`*DA_server_host*`:` *DA_server_port*/*DA_deploy_URI*`/UI/Login`

Where:

DA_server_protocol	Protocol (`http` or `https`) used by the Web container instance on which the Distributed Authentication UI server is deployed.
DA_server_host	Fully qualified host name of the Distributed Authentication UI server.
DA_server_port	Port for the host name of the Distributed Authentication UI server.
DA_deploy_URI	Deployment URI prefix for the Distributed Authentication UI server. The default value is `/amdistauth`.

For example:

```
https://daserver.example.com:80/amdistauth/UI/Login
```

Index