Sun Java System Access Manager 7.1 Postinstallation Guide

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

Part No: 819–5899–18
October 2, 2009

Copyright 2009 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun^{TM} Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2009 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou des applications de brevet en attente aux Etats-Unis et dans d'autres pays.

Cette distribution peut comprendre des composants développés par des tierces personnes.

Certaines composants de ce produit peuvent être dérivées du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays; elle est licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la legislation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la legislation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement designés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

091002@22749

Contents

Figures

Tables

Examples

Preface

The *Sun Java System Access Manager 7.1 Postinstallation Guide* provides information about configuring Sun Java™ System 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 or configure Access Manager for session failover.

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 Technical Overview*.
- Deployment platform: Solaris™, Linux, HP-UX, or Windows 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)

How This Book Is Organized

This book is organized by common configuration tasks, as outlined in the Contents.

Related Books

Related documentation is available as follows:

- "Access Manager 7.1 Documentation Set" on page 18
- "Sun Java Enterprise System 5 Documentation" on page 19

Access Manager 7.1 Documentation Set

The following table describes the Access Manager documentation set, which is available on the following Web site:

`http://docs.sun.com/coll/1292.2`

TABLE P–1 Access Manager 7.1 Documentation Set

Title	Description
Sun Java System Access Manager 7.1 Documentation Center	Contains links to commonly referenced information in the Access Manager documentation collection.
Sun Java System Access Manager 7.1 Release Notes	Describes new features, problems fixed, installation notes, and known issues and limitations. The Release Notes are updated periodically after the initial release to describe any new features or problems.
Sun Java System Access Manager 7.1 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.
Sun Java System Access Manager 7.1 Deployment Planning Guide	Provides planning and deployment solutions for Access Manager based on the solution life cycle.
Sun Java System Access Manager 7.1 Postinstallation Guide (this guide)	Provides 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 or configure Access Manager for session failover.
Sun Java System Access Manager 7.1 Administration Guide	Describes how to use the Access Manager console as well as manage user and service data via the command line interface.

TABLE P–1 Access Manager 7.1 Documentation Set *(Continued)*

Title	Description
Sun Java System Access Manager 7.1 Administration Reference	Provides reference information for the Access Manager command-line interface (CLI), configuration attributes, `AMConfig.properties` attributes, `serverconfig.xml` file attributes, log files, and error codes.
Sun Java System Access Manager 7.1 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.
Sun Java System Access Manager 7.1 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.
Sun Java System Access Manager 7.1 C API Reference	Provides summaries of data types, structures, and functions that make up the public Access Manager C APIs.
Sun Java System Access Manager 7.1 Java API Reference	Provides information about the implementation of Java packages in Access Manager.
Sun Java System Access Manager 7.1 Performance Tuning and Troubleshooting Guide	Provides information about how to tune Access Manager and its related components for optimal performance.
Sun Java System Access Manager Policy Agent 2.2 User's Guide	Provides an overview of Policy Agent software, including the web agents and J2EE agents that are currently available. To view the Access Manager Policy Agent 2.2 documentation collection, see: `http://docs.sun.com/coll/1322.1`

Sun Java Enterprise System 5 Documentation

The following table provides links to documentation collections for related Java ES products.

TABLE P–2 Related Sun Java Enterprise System 5 Documentation

Product	Link
Sun Java Enterprise System 5	`http://docs.sun.com/prod/entsys.06q4`
Sun Java System Directory Server Enterprise Edition 6	`http://docs.sun.com/coll/1224.1`
Sun Java System Web Server 7	`http://docs.sun.com/coll/1308.3`

TABLE P–2 Related Sun Java Enterprise System 5 Documentation *(Continued)*

Product	Link
Sun Java System Application Server Enterprise Edition 8.2	http://docs.sun.com/coll/1310.3
Sun Java System Message Queue 3.7 UR1	http://docs.sun.com/coll/1307.2
Sun Java System Web Proxy Server 4.0.4	http://docs.sun.com/coll/1311.4
Sun Java System Identity Manager 7	http://docs.sun.com/coll/1514.2

Searching Sun Product Documentation

Besides searching Sun product documentation from the docs.sun.com℠ web site, you can use a search engine by typing the following syntax in the search field:

search-term `site:docs.sun.com`

For example, to search for "broker," type the following:

```
broker site:docs.sun.com
```

To include other Sun web sites in your search (for example, java.sun.com, www.sun.com, and developers.sun.com), use `sun.com` in place of `docs.sun.com` in the search field.

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–3 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–4 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	`#`

Revision History

TABLE P–5 Revision History

Date and Part Number	Description of Change
February 14, 2007 (819–5899–10)	Initial release.
May 18, 2007 (819–5899–11)	In Chapter 12, "Deploying Access Manager as a Single WAR File," revised the information about the deploying an Access Manager 7.1 WAR file (`amserver.war`) and a Distributed Authentication UI server WAR file (`amauthdistui.war`).
June 7, 2007 (819–5899–12)	In Chapter 12, "Deploying Access Manager as a Single WAR File," added the requirement that to run the Configurator, the code set in the `LANG` environment variable must be set to `ISO8859-1`.
January 31, 2008 (819–5899–13)	Added Chapter 14, "Removing Access to the Access Manager Console." In Appendix A, "Directory Server Considerations": ■ Clarified the "Configuring Different Root Suffixes for the Access Manager Information Tree and User Directory Nodes" on page 185 section. ■ Added the "Disabling Persistent Searches in Directory Server" on page 183 section. ■ Added the "Specifying a User Naming Attribute Other Than the User ID (`uid`)" on page 191 section.
February 19, 2008 (819–5899–14)	In Chapter 6, "Implementing Session Failover," added the "Removing the Session Failover Configuration" on page 95 section. In Appendix A, "Directory Server Considerations," added the "Changing the Naming Attribute After Installation" on page 192 section.
March 4, 2008 (819–5899–15)	Added Appendix C, "Using Active Directory as the User Data Store."

TABLE P–5 Revision History *(Continued)*

Date and Part Number	Description of Change
May 5, 2008 (819–5899–16)	Clarified the "Configuring Different Root Suffixes for the Access Manager Information Tree and User Directory Nodes" on page 185 section, because Active Directory cannot be used as the configuration data store.
May 30, 2008 (819–5899–17)	Clarified the required LDIF file in Appendix C, "Using Active Directory as the User Data Store."
October 2, 2009 (819–5899–18)	Added a note that the Client Detection service is disabled for a WAR file deployment in "Considerations for an Access Manager WAR File Deployment" on page 167. Updated the permissions for a for a WAR file deployment in "Adding Access Manager Permissions to the Server Policy File" on page 160.

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 the *Sun Java System Access Manager 7.1 Postinstallation Guide*, and the part number is 819–5899.

◆ ◆ ◆ CHAPTER 1

Getting Started

The *Sun Java™ System Access Manager 7.1 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 or configure Access Manager for session failover.

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

This chapter describes these topics:

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:

If you are deploying an Access Manager WAR file, see Chapter 12, "Deploying Access Manager as a Single WAR File."

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 "Getting the Java ES Software" in *Sun Java Enterprise System 5 Installation Guide for UNIX* or the *Sun Java Enterprise System 5 Installation Guide for Microsoft Windows*.

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` template 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 information about the Java ES installer, see the *Sun Java Enterprise System 5 Installation Guide for UNIX* or the *Sun Java Enterprise System 5 Installation Guide for Microsoft Windows*.

Access Manager Single WAR File Deployment

You can also download an Access Manager 7.1 WAR file from the following web site:

http://www.sun.com/download/index.jsp

If you are using the Java EE 5 SDK release, you can also download the Access Manager 7.1 WAR file (and other components) from the following web site:

http://java.sun.com/javaee/downloads/index.jsp

To deploy an Access Manager WAR file, one of the following web containers must be running on the host server:

- Sun Java System Web Server 7
- Sun Java System Application Server Enterprise Edition 8.2
- Sun Java System Application Server Platform Edition 9 (as part of the Java EE 5 SDK release)
- BEA WebLogic Server
- IBM WebSphere Application Server

After you download the WAR file, follow these steps to deploy and configure Access Manager 7.1:

1. Deploy the Access Manager 7.1 WAR file using the web container's administrator console or CLI command.
2. Launch Access Manager 7.1, and you will be directed to the Configurator page, where you can provide information such as the host server URL, `amadmin` password, and the configuration directory.
3. Launch Access Manager 7.1 again, and you will be directed to the Access Manager Console login page.

For more information, see Chapter 12, "Deploying Access Manager as a Single WAR File."

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. The default base installation directory depends on your platform:

- Solaris systems: `/opt`

- Linux systems: /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.

Note – On Windows systems, the corresponding files are amconfig.bat and AMConfigurator.properties. These files are installed in the *javaes-install-dir*\identity\setup directory, where *javaes-install-dir* is the Java ES 5 installation directory. The default value is C:\Program Files\Sun\JavaES5.

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 "Access Manager amsamplesilent File Configuration Variables" on page 33.

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, "Running the Access Manager amconfig Script."

Access Manager Tuning Scripts

After you install Access Manager, you can tune your deployment for optimum performance using the Access Manager tuning scripts. These scripts allow you to tune Access Manager, the Solaris™ Operating System (OS), the web container, and Directory Server.

The Java Enterprise System installer installs the Access Manager tuning scripts and related files in the following directory, depending on your platform:

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

AccessManager-base represents the Access Manager base installation directory. The default base installation directory depends on your platform:

- Solaris systems: `/opt`
- Linux systems: `/opt/sun`

The `amtune` script is a top-level script that calls other tuning scripts as needed. This script is not interactive; before you run `amtune`, you edit parameters in the `amtune-env` configuration file to specify the tuning you want to perform for your specific environment. The `amtune-env` configuration file includes two major sections:

- Performance related parameters that you set to control the tuning
- An internal section that is maintained by Access Manager engineering and should not be modified

You can run the `amtune` script in two modes:

- Review mode: `amtune` reports tuning recommendations but does not make any actual changes to your environment.
- Change mode: `amtune` makes actual changes, except for Directory Server, depending on parameters in the `amtune-env` configuration file.

The `amtune` script does not automatically tune Directory Server. Most deployments have applications other than Access Manager that also access Directory Server, so you don't want to make tuning changes without considering how they would affect the other applications.

Before you tune Directory Server, first back up your Directory Server data.

When you run `amtune`, the script creates a tar file that contains the Directory Server tuning script, `amtune-directory`. Untar this file in a temporary directory and then run the script in review mode. When you are certain that your changes are acceptable for all applications at your deployment, run `amtune-directory` in change mode.

For detailed information about running the tuning scripts and setting tuning parameters in the `amtune-env` configuration file, see the *Sun Java System Access Manager 7.1 Performance Tuning and Troubleshooting Guide*.

◆ ◆ ◆ 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:

Note – On Windows systems, the corresponding files are `amconfig.bat` and `AMConfigurator.properties`. These files are installed in the *javaes-install-dir*`\identity\setup` directory, where *javaes-install-dir* is the Java ES 5 installation directory. The default value is `C:\Program Files\Sun\JavaES5`.

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`) are 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. The default base installation directory depends on your platform:

- Solaris systems: `/opt`
- Linux systems: `/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
- 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.

To run the amconfig script, use this syntax:

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

where:

-s runs amconfig in silent mode.

To run amconfig.bat, either double click on the file or execute the file from the command prompt. The amconfig.bat does not accept any command-line parameters like the amconfig script.

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 33.

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 – On Windows systems, to configure Access Manager, run `amconfig.bat` with `AMConfigurator.properties`. These files are installed in the *javaes-install-dir*`\identity\setup` directory, where *javaes-install-dir* is the Java ES 5 installation directory. The default value is `C:\Program Files\Sun\JavaES5`.

To run `amconfig.bat`, double click on the file or execute the file from the Windows command prompt.

Unsupported Scripts

In the Access Manager 7.1 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" on page 34
- "Access Manager Configuration Variables" on page 35
- "Web Container Configuration Variables" on page 40
- "Directory Server Configuration Variables" on page 46

Other configuration variables are documented in the following chapters:

- Access Manager client SDK: Chapter 10, "Deploying the Client SDK"
- Distributed Authentication UI server: Chapter 11, "Deploying a Distributed Authentication UI Server"

Note – On Windows systems, the silent configuration input file is AMConfigurator.properties. This file is installed in the *javaes-install-dir*\identity\setup directory, where *javaes-install-dir* is the Java ES 5 installation directory. The default value is C:\Program Files\Sun\JavaES5.

Although Windows paths use backslashes (\), the AMConfigurator.properties file must use only slashes (/) and should not contain any backslashes. For example: C:/Sun/JavaES5

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.

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

Operation	DEPLOY_LEVEL Variable Value and Description
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
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 features and console. ■ disabled: Access Manager operates in Legacy Mode, with Access Manager 6 2005Q1 features and console. In Legacy Mode, Access Manager has Access Manager 6 2005Q1 features, in addition to Access Manager 7.1 and console. You will be directed to Access Manager mode, depending on the deployment descriptor you use: ■ Realm Mode: http://*host*:*port*/amserver ■ Legacy Mode: http://*host*:*port*/amconsole Default: enabled **Caution –** Access Manager Realm Mode is enabled by default. If you are deploying Access Manager with 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 On Solaris systems, PLATFORM_DEFAULT is /opt On Linux systems, PLATFORM_DEFAULT is /opt/sun On HP—UX systems, PLATFORM_DEFAULT is /opt/sun On Windows systems, the base installation directory is the Java ES installation directory. The default value is C:\Program Files\Sun\JavaES5.
SERVER_NAME	Name of local host where 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: 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 35.
ROOT_SUFFIX	Initial or root suffix of the directory user management node. 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 35.
SM_CONFIG_BASEDN	Initial or root suffix of the Access Manager information tree (service management node). By default, the value of SM_CONFIG_BASEDN is the same as the ROOT_SUFFIX variable. On Windows system, set to blank if the value is same as the ROOT_SUFFIX variable.
ADMINPASSWD ADMIN_PASSWORD (Windows systems only)	Password for the Access Manager 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 35.
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

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

Variable	Description
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.
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 of 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

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

Variable	Description
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 **Application Server Consideration**: If you are deploying Access Manager with Application Server as the web container, use the Domain Administration Server (DAS) as the web container for testing purposes only. In a production environment, create a new Application Server instance to use as the Access Manager web container and set NEW_INSTANCE=true.
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 Release Notes*.

TABLE 2–3 Access Manager WEB_CONTAINER Variable

WEB_CONTAINER Value	Web Container
WS	"Sun Java System Web Server 7" on page 40
WS6	"Sun Java System Web Server 6.1 SP5" on page 41
AS8 (default)	"Sun Java System Application Server 8.1" on page 42
WL8	"BEA WebLogic Server 8.1" on page 44
WAS5	"IBM WebSphere Application Server 5.1" on page 45

Sun Java System Web Server 7

This section describes the configuration variables for Web Server 7.

TABLE 2–4 Web Server 7 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 WS_HOME value. Default: Solaris systems: /var/opt/SUNWwbsvr7/https-$SERVER_HOST Linux systems: /var/opt/sun/webserver7/https-$SERVER_HOST HP-UX systems: https-$SERVER_HOST Windows systems: https-*hostname*
WS_HOME	Web Server instance directory. Defaults: Solaris systems: /var/opt/SUNWwbsvr7 Linux systems: /var/opt/sun/webserver7/$WS_INSTANCE HP-UX systems: /var/opt/sun/webserver7 Windows systems: *javaes-install-dir*/webserver7 *javaes-install-dir* represents the Java ES 5 installation directory. The default value is C:\Program Files\Sun\JavaES5.
WS_PROTOCOL	Protocol (http or https) used by the Web Server instance. Default: SERVER_PROTOCOL variable
WS_HOST	Fully qualified domain name on which the Web Server instance is listening for connections. Default: SERVER_HOST variable 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)
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 35 variable)
WS61_HOME	Web Server base installation directory. Default: Solaris systems: /opt/SUNWwbsvr HP-UX systems: /opt/sun/webserver Windows systems: *javaes-install-dir*/webserver *javaes-install-dir* represents the Java ES 5 installation directory. The default value is C:\Program Files\Sun\JavaES5.
WS61_PROTOCOL	Protocol used by the Web Server instance set by the "Sun Java System Web Server 6.1 SP5" on page 41 variable where Access Manager will be deployed: http or https. Default: Access Manager protocol ("Access Manager Configuration Variables" on page 35 variable)
WS61_HOST	Fully qualified host name for the Web Server instance ("Sun Java System Web Server 6.1 SP5" on page 41 variable). Default: Access Manager host instance ("Access Manager Configuration Variables" on page 35 variable)
WS61_PORT	Port on which Web Server listens for connections. Default: Access Manager port number ("Access Manager Configuration Variables" on page 35 variable)
WS61_ADMINPORT	Port on which the Web Server Administration Server listens for connections. Default: 8888
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: Solaris systems: /opt/SUNWappserver/appserver HP-UX systems: /opt/sun/appserver Windows systems: *javaes-install-dir*/appserver *javaes-install-dir* represents the Java ES 5 installation directory. The default value is C:\Program Files\Sun\JavaES5.
AS81_PROTOCOL	Protocol used by the Application Server instance: http or https. Default: Access Manager protocol ("Access Manager Configuration Variables" on page 35 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 35 variable)
AS81_PORT	Port on which Application Server instance listens for connections. Default: Access Manager port number ("Access Manager Configuration Variables" on page 35 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 35.
AS81_INSTANCE	Name of the Application Server instance that will run Access Manager. Default: server
AS81_DOMAIN	Path to the Application Server directory for the domain to which you want to deploy this Access Manager instance. Default: domain1

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

Variable	Description
AS81_INSTANCE_DIR	Path to the directory where Application Server stores files for the instance. Default: Solaris systems: /var/opt/SUNWappserver/domains/domain1 HP-UX systems: /var/opt/sun/appserver/domains/domain1 Windows systems: *javaes-install-dir*/appserver/domains/domain1 *javaes-install-dir* represents the Java ES 5 installation directory. The default value is C:\Program Files\Sun\JavaES5.
AS81_DOCS_DIR	Directory where Application Server stores content documents. Default: Solaris systems: /var/opt/SUNWappserver/domains/domain1/docroot HP-UX systems: /var/opt/sun/appserver/domains/domain1/docroot Windows systems: *javaes-install-dir*/appserver/domains/domain1/docroot *javaes-install-dir* represents the Java ES 5 installation directory. The default value is C:\Program Files\Sun\JavaES5.
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: Solaris systems: /usr/local/bea Windows systems: *weblogic-install-dir* For example: C:/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 **Note**: For a WebLogic managed server deployment, set WL8_SERVER to the name of the managed instance within the domain, and set SERVER_PORT=7001, to point to the WebLogic Admin Server port.
WL8_INSTANCE	WebLogic instance name. Default: Solaris systems: /usr/local/bea/weblogic81 ($WL8_HOME/weblogic81) Windows systems: *weblogic-install-dir*/weblogic81
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 35.
WL8_JDK_HOME	WebLogic JDK home directory. Default: "BEA WebLogic Server 8.1" on page 44 /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: Solaris systems: /opt/WebSphere/AppServer Windows systems: *websphere-install-dir*/WebSphere/AppServer For example: C:/WebSphere/AppServer
WAS51_JDK_HOME	WebSphere JDK home directory. Default: Solaris systems: /opt/WebSphere/AppServer/java Windows systems: *websphere-install-dir*/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, see the *Sun Java System Access Manager 7.1 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 for multiple Access Manager instances on either the same host server or on multiple host servers. The naming attributes and object classes are the same, so the configuration scripts load the installExisting.ldif and umsExisting.xml 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.xml 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 multiple-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 add 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 To specify another value such as the user's email attribute (mail) or common name (cn), see "Specifying a User Naming Attribute Other Than the User ID (uid)" on page 191.
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

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. See also Chapter 3, "Deploying Multiple Access Manager Instances."

This section also describes the following scenarios:

Configuring and Reconfiguring an Instance of Access Manager on UNIX and Linux Systems

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.

The following steps apply to Solaris, HP-UX, and Linux systems.

▼ To Configure or Reconfigure an Instance of Access Manager on UNIX and Linux Systems

1. **Log in as an administrator, depending on the web container for the instance.**

   For example, if Web Server 7 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 7, the following steps use an input file named `amnewinstanceforWS7` in the `/reconfig` directory.

3. **In the `amnewinstanceforWS7` file, set the `DEPLOY_LEVEL` variable to one of the values described for a "Deployment Mode Variable" on page 34 operation.**

   For example, set `DEPLOY_LEVEL=21` to reconfigure a full installation.

4. **In the `amnewinstanceforWS7` file, set the `NEW_INSTANCE` variable to false:**

   ```
   NEW_INSTANCE=false
   ```

5. **Set other variables in the `amnewinstanceforWS7` file to configure or 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 35
     - "Web Container Configuration Variables" on page 40
     - "Directory Server Configuration Variables" on page 46

6. **Run the `amconfig` script, specifying your edited input file.**

   For example, on Solaris systems with Access Manager installed in the default directory:

   ```
   # cd opt/SUNWam/bin/
    # ./amconfig -s ./reconfig/amnewinstanceforWS7
   ```

   The `-s` option runs the script in silent mode. The `amconfig` script calls other configuration scripts as needed, using variables in the `amnewinstanceforWS7` file to reconfigure the instance.

Configuring and Reconfiguring an Instance of Access Manager on Windows Systems

The following steps apply only to Windows systems.

▼ To Configure or Reconfigure an Instance of Access Manager on Windows Systems

1. **Log in as an administrator, depending on the web container for the Access Manager instance.**

2. **Make a copy of the `AMConfigurator.properties` silent configuration input file.**

   For example: `AMConfigurator-redeploy.properties`

3. **In the new `AMConfigurator-redeploy.properties` file, set the `DEPLOY_LEVEL` variable to one of the values described for a "Deployment Mode Variable" on page 34 operation.**

   For example, set `DEPLOY_LEVEL=21` to reconfigure a full installation.

4. **In the `AMConfigurator-redeploy.properties` file, set the `NEW_INSTANCE` variable to `false`.**

5. **Set other variables in the `AMConfigurator-redeploy.properties` file to configure or reconfigure the instance.**

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

   - "Access Manager Configuration Variables" on page 35
     - "Web Container Configuration Variables" on page 40

- "Directory Server Configuration Variables" on page 46

6. **Edit the `amconfig.bat` file and change `AMConfigurator.properties` to `AMConfigurator-redeploy.properties`.**

7. **Run `amconfig.bat` by double clicking on the file or executing the file from the Windows command prompt.**

Uninstalling Access Manager on UNIX and Linux Systems

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.

The following steps apply to Solaris, HP-UX, and Linux systems.

▼ To Uninstall an Instance of Access Manager on UNIX and Linux Systems

1. **Log in as an administrator, depending on the web container for the instance.**

   For example, if Web Server 7 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 7, the following steps use an input file named `amnewinstanceforWS7` in the `/unconfigure` directory.

3. **In the `amnewinstanceforWS7` file, set the `DEPLOY_LEVEL` variable to one of the values described for an "Deployment Mode Variable" on page 34 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 with Access Manager installed in the default directory:

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

   The `-s` option runs the script in silent mode. The `amconfig` script reads the `amnewinstanceforWS7` 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 Access Manager on Windows Systems

The following steps apply only to Windows systems.

To Uninstall an Instance of Access Manager on UNIX and Linux Systems

1. **Log in as an administrator, depending on the web container for the Access Manager instance.**
2. **Make a copy of the `AMConfigurator.properties` silent configuration input file.**

   For example: `AMConfigurator-uninstall.properties`
3. **In the new `AMConfigurator-redeploy.properties` file, set `DEPLOY_LEVEL=11`.**
4. **Edit the `amconfig.bat` file as follows:**
   - Change `–configure` to `–unconfigure`.
   - Change `AMConfigurator.properties` to `AMConfigurator-uninstall.properties`.
5. **Run `amconfig.bat` by double clicking on the file or executing the file from the Windows command prompt.**

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 34 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 51.**

   For example, on Solaris systems with Access Manager installed in the default directory:

   ```
   # cd opt/SUNWam/bin/
   # ./amconfig -s ./newinstances/amnewws7instance
   ```

   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 5 Installation Guide for UNIX* or the *Sun Java Enterprise System 5 Installation Guide for Microsoft Windows*.

◆ ◆ ◆ 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`.

  On Windows systems, the `amconfig.bat` and `AMConfigurator.properties` files are available in the default installation directory: `C:\Program files\Sun\JavaES5`.
- 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 5 Installation Guide for UNIX* or the *Sun Java Enterprise System 5 Installation Guide for Microsoft Windows*.

Running the Java ES Installer on UNIX and Linux Systems

Considerations for running the Java ES installer on Solaris, HP-UX, and Linux systems to install an Access Manager instance 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 5 Installation Guide for UNIX* or the *Sun Java Enterprise System 5 Installation Guide for Microsoft Windows*.

Running the Java ES Installer on Windows Systems

Considerations for running the Java ES installer on Windows systems to install an Access Manager instance 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.bat` and `AMConfigurator.properties` files are available in the following default installation directory: `C:\Program files\sun\JavaES`.
- When you run the installer, specify either the "Configure Automatically during install" or "Configure Manually after install" option.
  - Configure Automatically during install: 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 Manually after install: When you run the Java ES installer, you specify only minimal configuration values. Then, you later configure the specific components by running a batch file or using an administration console. Access Manager provides the `amconfig.bat` and `AMConfigurator.properties` files 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 Manually after install" 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 Manually after install" option.
  3. Configure Access Manager for the web container by setting variables in the `AMConfigurator.properties` configuration file (or a copy of the file) and then running `amconfig.bat`.

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

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 7, you might rename the file to `amwebsvr7`.

c. Set the variables in the amwebsvr7 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=WS # Web Server 7 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 Chapter 13, “Changing the Password Encryption Key.”

3 Run the amconfig script.

For example, on Solaris systems with Access Manager installed in the default directory, run amconfig using the new amwebsvr7 file as the configuration input file:

```
# cd /opt/SUNWam/bin/
# ./amconfig -s ./newinstances/amwebsvr7
```

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

The amconfigscript reads the variables in the amwebsvr7 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, "Running the Access Manager amconfig Script."

4 **In case you might need to reconfigure or uninstall this instance later, save the new** amwebsvr7 **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 list (or the DNS Alias list in Legacy Mode). You must explicitly add these values for additional Access Manager instances.

If you are using Access Manager in Legacy Mode, see "Adding Additional Instances to the Platform Server List and DNS Alias List in Legacy Mode" on page 59.

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

1 **Log in to the Access Manager 7.1 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`**.**

Adding Additional Instances to the Platform Server List and DNS Alias List in Legacy Mode

The following procedure refers to the Access Manager 7.1 in Legacy Mode.

▼ To Add Additional Instances to the Platform Server List and DNS Alias List in Legacy Mode

1. **Log in to the Access Manager Legacy Console as** `amadmin` **on the first Access Manager host server.**

2. **Add each additional instance to the Platform Server List:**

   a. **Click** `Service Configuration`**.**

   b. **In the left pane, click the** `Platform` **link.**

   c. **Under the** `Server List`**, add each additional host server. For example:**

   ```
   http://amserver2.example.com:58080|02
   http://amserver3.example.com:58080|03
   ```

   d. **After you have added all instances, click** `Save`**.**

3. **Add each additional instance to the DNS Alias List:**

   a. **Click** `Identity Management`**.**

   b. **Make sure that** `View: Organizations` **is selected in the left pane.**

c. In the DNS Alias Name **field in the right pane, add each additional host server name. For example:**

```
amserver2.example.com
amserver3.example.com
```

d. After you have added all instances, click Save**.**

◆ ◆ ◆ CHAPTER 4

Configuring Access Manager With a Load Balancer

A load balancer distributes the client requests between the Access Manager instances in multiple server deployment. To use a load balancer, your deployment must be configured as a site. A site includes multiple (two or more) instances of Access Manager deployed on different host servers. All Access Managers instances must access the same Directory Server and use the same password encryption key.

This chapter includes the following configuration topics:

Configuring an Access Manager Deployment as a Site

An Access Manager deployment configured as a site allows centralized configuration management for the entire deployment.

Requirements for an Access Manager Site

An Access Manager site includes the following components:

- **Multiple server deployment**: 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.

- **Load balancer**: 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. A load balancer simplifies the deployment, as well as resolves issues such as a firewall between the client and the back-end Access Manager servers.

  You can use a hardware or software load balancer with your Access Manager deployment. For example, for information about the Application Server Load Balancing Plug-in, see the *Sun Java System Application Server Enterprise Edition 8.2 Quick Start Guide*.
- **Directory Server**: All Access Manager instances access the same Directory Server.

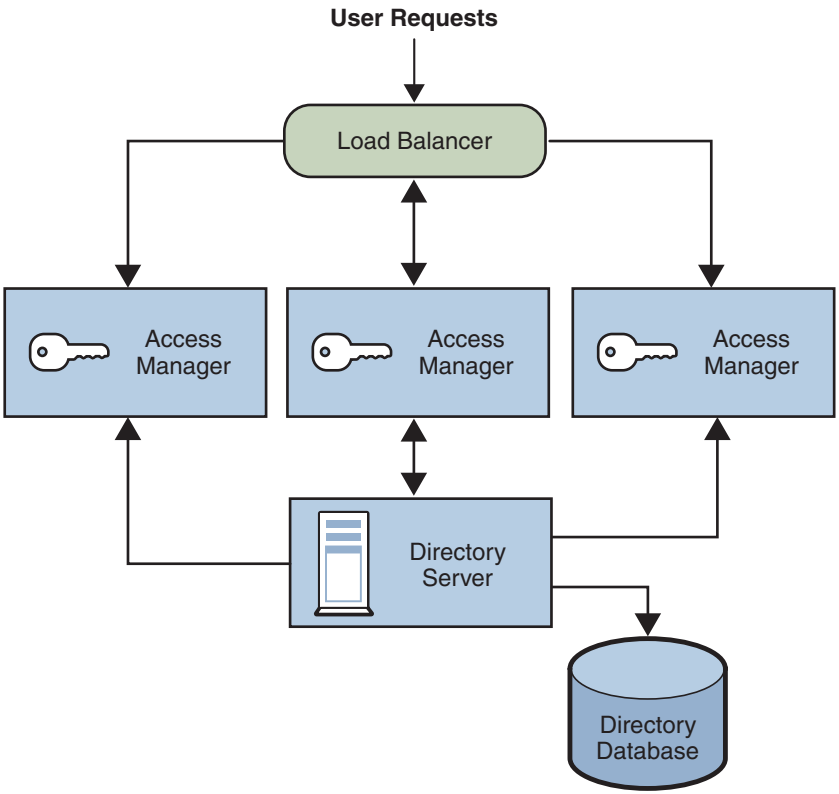

FIGURE 4–1 Access Manager Site

Access Manager Site Configuration

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 Chapter 6, "Implementing Session Failover."
- If you don't plan to implement session failover, perform these steps the Access Manager Console, as described in this section:
- 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 in Realm Mode

The following procedure refers to the Access Manager 7.1 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.**

▼ To Configure Access Manager as a Site in Legacy Mode

The following procedure refers to the Access Manager 7.1 Console in Legacy Mode.

1 **Log in to the Access Manager Legacy Console as `amadmin` on the first Access Manager host server.**

2 **Add each additional instance to the Platform Server List:**

a. **Click Service Configuration.**

b. **In the left pane, click the Platform link.**

c. **Under the Server List, add each additional host server.**

For example:

```
http://amserver2.example.com:58080|02
http://amserver3.example.com:58080|03
```

d. After you have added all instances, click Save.

3 Add each additional instance to the DNS Alias List:

a. Click Identity Management.

b. Make sure that View: Organizations is selected in the left pane.

c. In the DNS Alias Name field in the right pane, add each additional host server name.

For example:

```
amserver2.example.com
amserver3.example.com
```

d. After you have added all instances, click Save.

Configuring Cookie-Based Sticky Request Routing

When Access Manager servers are deployed behind a load balancer, cookie-based sticky request routing prevents a client request from being misrouted to an incorrect Access Manager server (that is, to a server that is not hosting the session).

In the previous behavior, without cookie-based sticky request routing, requests from non-browser based clients (such as policy agents and clients using the remote Access Manager client SDK) were often misrouted to an Access Manager server that was not hosting the session. Then, in order to send the request to the correct server, the Access Manager server had to validate the session using back-channel communication, which usually caused some performance degradation.

Cookie-based sticky request routing prevents the need for this back-channel communication and thus improves Access Manager performance.

▼ To Configure Cookie-Based Sticky Request Routing

Before You Begin The Access Manager deployment must be configured as a site. For information, see "Configuring an Access Manager Deployment as a Site" on page 61. When you configure a deployment as a site, Access Manager automatically sets the `fqdnMap` property (in memory) to include the load balancer.

1 To specify a cookie name, set the `com.iplanet.am.lbcookie.name` property in the `AMConfig.properties` file.

Access Manager then generates the load balancer cookie value using the two-byte server ID (such as 01, 02, and 03). If you do not specific a cookie name, Access Manager generates the load balancer cookie value using the default name `amlbcookie` plus the two-byte server ID.

If you set the cookie name on the Access Manager server, you must use the same name in the `AMAgent.properties` file for a Policy Agent. Also, if you are using the Access Manager client SDK, you must also use the same cookie name used by the Access Manager server.

Note: Do not set the `com.iplanet.am.lbcookie.value` property, because Access Manager sets the cookie value using the two-byte server ID.

2 **Configure the load balancer with the cookie name from Step 1.**

You can use a hardware or software load balancer with your Access Manager deployment. For information about configuring the BIG-IP® load balancer manufactured by F5 Networks, see *Deployment Example 1: Access Manager 7.1 Load Balancing, Distributed Authentication UI, and Session Failover*.

3 **If session failover is implemented, enable the `com.sun.identity.session.resetLBCookie` property for both Policy Agents and the Access Manager server. For example:**

```
com.sun.identity.session.resetLBCookie='true'
```

- For a Policy Agent, add and enable the property in the `AMAgent.properties` file.
- For the Access Manager server, add and enable the property in the `AMConfig.properties` file.

If a failover situation occurs, the session is routed to a secondary Access Manager server, and the load balancer cookie value is set using the server ID for the secondary Access Manager server. Any subsequent requests for the session are routed to the secondary Access Manager server.

Configuring SSL Termination for a Load Balancer

Before you configure a load balancer to handle SSL requests, first configure SSL for the Access Manager web container. For information, see Chapter 8, "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 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 "Configuring an Access Manager Deployment as a Site" on page 61.

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 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 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 Console. For example:

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

◆ ◆ ◆ CHAPTER 5

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 as the session store database. For more information about Access Manager session failover, see Chapter 6, "Implementing Session Failover."

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 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 6

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 as the session store database. This chapter describes these topics:

Access Manager Session Failover Scenario

Figure 6–1 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.

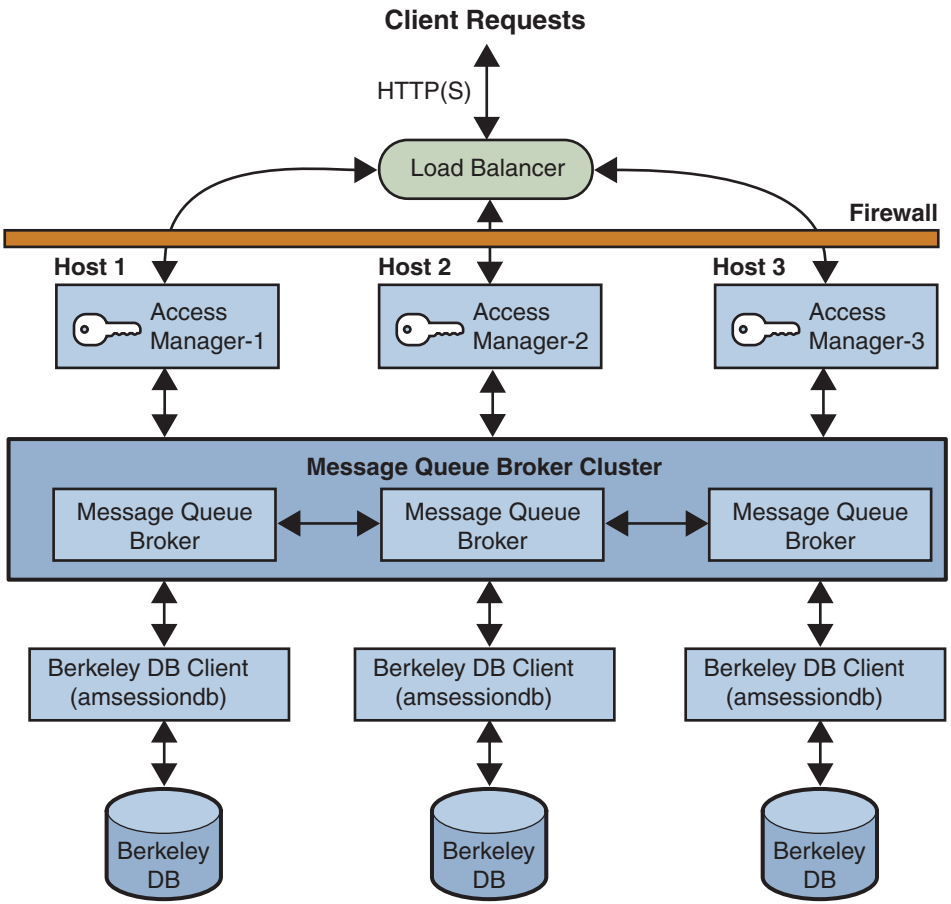

FIGURE 6–1 Access Manager Session Failover Scenario

Installing the Session Failover Components

The following table describes how to install the components required for Access Manager session failover using the Java ES installer. If you are deploying an Access Manager WAR file, see Chapter 12, "Deploying Access Manager as a Single WAR File."

TABLE 6–1 Installation of Access Manager Session Failover Components Using the Java ES Installer

Component	Installation
Sun Java System 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. Installation reference: *Sun Java Enterprise System 5 Installation Guide for UNIX* or the *Sun Java Enterprise System 5 Installation Guide for Microsoft Windows* When you install Access Manager using the Java ES installer, select either Realm Mode or Legacy Mode. 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. Reference: Chapter 3, "Deploying Multiple Access Manager Instances"
Sun Java System Message Queue	Install Message Queue using the Java ES installer. Installation reference: *Sun Java Enterprise System 5 Installation Guide for UNIX* or the *Sun Java Enterprise System 5 Installation Guide for Microsoft Windows* Message Queue documentation: `http://docs.sun.com/coll/1307.2`
Session Failover Client Berkeley DB	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 and HP-UX 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, depending on your platform:

- Solaris systems: `/var/opt/SUNWam/debug` directory
- Linux and HP-UX systems: `/var/opt/sun/identity/debug` directory
- Windows systems: *javaes-install-dir*`\identity\debug`

  javaes-install-dir represents the Java ES 5 installation directory. The default value is `C:\Program Files\Sun\JavaES5`.

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 in directories that begin with `https-`, set the `encodeCookies` property to false. For example:

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

- If Sun Java System 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–Modifying the Web Container Server classpath

On each host server that is running an Access Manager instance, use the web container Admin console or CLI command to add the installed locations of the `imq.jar` and `jms.jar` files to the server classpath.

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.

Note – On Windows systems, Access Manager provides the amsfo.pl script and amsfo.conf file to configure an Access Manager deployment for session failover. To run this script, Active Perl version 5.8 or later is required.

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 6–2 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`

TABLE 6–2 Access Manager Session Failover Scripts and Configuration Files *(Continued)*

Name	Description and Location
`amsfopassword`	Script to generate the encrypted Message Queue broker user password. Solaris systems: *AccessManager-base*`/SUNWam/bin` Linux and HP-UX systems: *AccessManager-base*`/identity/bin` Windows systems: *javaes-install-dir*`\identity\bin` *javaes-install-dir* represents the Java ES 5 installation directory. The default value is `C:\Program Files\Sun\JavaES5`.
`amsfo.conf`	Session failover configuration file. Solaris systems: *AccessManager-base*`/SUNWam/lib` Linux and HP-UX systems: *AccessManager-base*`/sun/identity/lib` Windows systems: *javaes-install-dir*`\identity\lib` *javaes-install-dir* represents the Java ES 5 installation directory. The default value is `C:\Program Files\Sun\JavaES5`.
`amProfile.conf`	Session failover environment file. Solaris systems: `etc/opt/SUNWam/config` Linux and HP-UX systems: `etc/opt/sun/identity/config` Windows systems: *javaes-install-dir*`\identity\config` *javaes-install-dir* represents the Java ES 5 installation directory. The default value is `C:\Program Files\Sun\JavaES5`.

AccessManager-base represents the base installation directory for Access Manager. The default values are:

Solaris systems: `/opt`

Linux and HP-UX 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 `amsfoconfig` script (or `amsfo.pl` script on Windows systems) . 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 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 6–3 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 2005Q4

        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: password2(nothing will be echoed):
Verify: password2

         Retrieving Platform Server list...

         Validating server entries.
         Done...

         Retrieving Site list...

         Validating site entries.
         Done...

         Validating host: http://amhost1.example.com:80|01

         Validating host: http://amhost2.example.com:80|02
         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 Organization Alias List File...

            Organization Alias List loaded successfully.

            Loading Platform Server List File...

            Platform Server List server entries loaded successfully.


Please refer to the log file /var/tmp/amsfoconfig.log for additional information.
#################################################################
Session Failover Setup Script. Execution end time 12/12/06 15:03:30
#################################################################
```

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 6–3.
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" on page 88.
3. Start each Access Manager instance by starting the respective web container.

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 6–4 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).

TABLE 6–4 `amsfo.conf` Configuration File *(Continued)*

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.
`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`.
`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.

TABLE 6–4 `amsfo.conf` Configuration File *(Continued)*

Variable	Description
`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 `amsfopassword` script, as described in `amsfopassword` Script The default is *$AM_HOME_DIR*`/.password`, where *$AM_HOME_DIR* specifies the Access Manager default installation directory.

Running the `amsfopassword` Script

The `amsfopassorwd` 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 `amsfopassword` 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 `amsfopassword` script.

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

The following table describes the `amsfopassword` script arguments.

TABLE 6–5 `amsfopassword` Script Arguments

Argument	Description
`-f` *filename* `|` `--passwordfile` *filename*	Path to the destination file where `amsfopassword` stores the encrypted password.
`-e` *password* `|` `--encrypt` *password*	Clear text password that `amsfopassword` encrypts.
`-h | --help`	Display the `amsfopassword` command usage and then exit.

▼ To Run the `amsfopassword` Script

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

2. **Run the `amsfopassword` script. For example, on a Solaris system with Access Manager installed in the default directory:**

```
# cd /opt/SUNWam/bin
# ./amsfopassword -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 91
- "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 92
- "4–Perform Session Failover Miscellaneous Configuration Tasks" on page 92
- "5–Start the Session Failover Components" on page 92
- "`amsessiondb` Script" on page 93

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" on page 78.

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 "Configuring an Access Manager Deployment as a Site" on page 61.

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 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 and Stopping the Session Failover Components" on page 87.

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–Editing the amsessiondb Script (if Needed)" on page 82.

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 amsfopassword 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 6–6 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".

TABLE 6–6 `amsessiondb` Script Arguments *(Continued)*

Argument	Description
-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
```

Removing the Session Failover Configuration

In this scenario, you want to remove the session failover configuration for a deployment.

▼ To Remove a Session Failover Configuration

1. **In the Access Manager Administration Console, remove the session failover configuration (that is, the secondary configuration under Session Service in the Console).**
2. **Restart all the Access Manager servers participating in the cluster.**
3. **Shutdown the Message Queue broker instances and `amsessiondb` instances using the `amsfo` script on the target systems.**

   For more information, see "Running the `amsfo` Script" on page 88.
4. **In the web container `server.xml` file, remove the installed locations of the `imq.jar` and `jms.jar` files. For example:**

```
<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"
```

5. **Optionally, uninstall the Message Queue and Berkeley DB components from the target systems.**

Next Steps Several other considerations are:

- After you remove the session failover configuration, determine whether you also want to remove the site configuration for the deployment. If you keep the site configuration without session failover, session constraints (if configured) are not supported.
- If the cookie encoding setting on the Access Manager server side is restored, the corresponding setting of the cookie encoding for a remote client might also need to be restored.

◆ ◆ ◆ CHAPTER 7

Installing and Configuring Third-Party Web Containers

Sun Java™ System Access Manager 7.1 supports the following third-party web containers:

- BEA WebLogic Server

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

- IBM WebSphere Application Server

  `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:

- WebLogic Server and WebSphere Application Server are not part of the Sun Java Enterprise System (Java ES). Therefore, you must obtain the web container software from BEA or IBM and then install and configure them independently of the Java ES installer.
- You should be familiar with administration tasks for the web container, including configuring, starting, and stopping an instance.

  When you configure Access Manager by running the `amconfig` script (or `amconfig.bat` on Windows systems), the web container must be installed, configured, and running.
- 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.

General Steps For Using a Third-Party Web Container

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

1. If necessary, install Sun Java System Directory Server.
2. Install and configure the web container by following the BEA or IBM documentation.
3. Install Access Manager by running the Java ES installer with the Configure Later option.
4. Start the web container.
5. Configure Access Manager for the web container by running the `amconfig` script with configuration parameters specified in the `amsamplesilent` file (or a copy of the file). On Windows systems, run `amconfig.bat` with configuration parameters specified in the `AMConfigurator.properties` file (or a copy of the file).
6. 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://download-llnw.oracle.com/docs/cd/E13222_01/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 99 when you run the Access Manager `amconfig` script (or `amconfig.bat` on Windows systems).

▼ 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.

On Windows systems, On Windows systems, run `amconfig.bat` with configuration parameters specified in the `AMConfigurator.properties` file (or a copy of the file).

TABLE 7–1 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 101 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.

On Windows systems, On Windows systems, run `amconfig.bat` with configuration parameters specified in the `AMConfigurator.properties` file (or a copy of the file).

TABLE 7–2 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`
`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 7.1 with the Configure Later option.
- Other Java ES components as needed. For example, if you are planning to configure Access Manager for session failover, install Sun Java System Message Queue.

For information about running the installer, see *Sun Java Enterprise System 5 Installation Guide for UNIX* or the *Sun Java Enterprise System 5 Installation Guide for Microsoft Windows*.

Configuring Access Manager Using the amconfig Script

On Windows systems, run `amconfig.bat` with configuration parameters specified in the `AMConfigurator.properties` file (or a copy of the file).

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 configure an Access Manager instance for WebLogic Application Server, you might name the file as `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. Before you run the `amconfig` script, set the `AM_ENC_PWD` variable in the copy of the `amsamplesilent` file with the same encryption key value used for other instances.

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

For example, on Solaris systems with Access Manager 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 amconfigscript reads the variables in the am_weblogic_server file and then runs in silent mode (-s option) to configure Access Manager for the WebLogic Application Server web container.

For more information about the amsamplesilent file and running the amconfig script, see Chapter 2, "Running the Access Manager amconfig Script"Chapter 2.

5 Restart the web container.

Next Steps In case you might need to reconfigure or uninstall this instance later, save the new am_weblogic_server file.

◆ ◆ ◆ CHAPTER 8

Configuring Access Manager in SSL Mode

Using the Secure Sockets Layer (SSL) protocol with simple authentication guarantees confidentiality and data integrity. To enable Access Manager to use SSL, mode you would typically:

Configuring Access Manager With a Secure Sun Java System Web Server

This section describes how to configure Access Manager in SSL mode with Sun Java System Web Server.

▼ To Configure a Secure Web Server

1. **Login to the Access Manager Console as `amadmin`.**
2. **Click Configuration, System Properties, and then Platform.**
3. **Under Server Instance, click the server name.**
4. **Change the `http://` protocol to the `https://` protocol.**
5. **Click OK and then Save.**

Note – Be sure to click Save. If you don't, you will still be able to continue 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.

6 **Login to the Web Server console. The default port is 8888.**

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

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

8 **Click Apply and then Apply Changes.**

9 **Click Apply Changes.**

Web Server should automatically restart. Click OK to continue.

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

11 **Click the Security Tab.**

12 **Click on Create Database.**

13 **Enter the new database password and click OK.**

Ensure that you write down the database password for later use.

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

15 **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.

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

```
--BEGIN CERTIFICATE REQUEST---

afajsdllwqeroisdaoi234rlkqwelkasjlasnvdknbslajowijalsdkjfalsdflasdf

alsfjawoeirjoi2ejowdnlkswnvnwofijwoeijfwiepwerfoiqeroijeprwpfrwl

--END CERTIFICATE REQUEST--
```

17 **Copy this text and submit it for the certificate request.**

Ensure that you get the Root CA certificate.

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

```
--BEGIN CERTIFICATE---

afajsdllwqeroisdaoi234rlkqwelkasjlasnvdknbslajowijalsdkjfalsdflasdf

alsfjawoeirjoi2ejowdnlkswnvnwofijwoeijfwiepwerfoiqeroijeprwpfrwl

--END CERTIFICATE---
```

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

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

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

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

23 **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.

24 **Click Install Certificate.**

25 **Click Certificate for Trusted Certificate Authority.**

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

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

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

29 **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.

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

31 **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.**

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

33 **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.

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

Next Steps If you are configuring Access Manager certificate authentication with an SSL-enabled Web Server 6.1 instance and want Web Server to accept both certificate-based and non-certificate-based authentication requests, set the following value in the Web Server `obj.conf` file:

```
PathCheck fn="get-client-cert" dorequest="1" require="0
```

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 8.2 With SSL

This section describes the steps to set up Application Server 8.2 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*.
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" also in the *Sun Java System Application Server Enterprise Edition 8.1 Administration Guide*.

   The *Sun Java System Application Server Enterprise Edition 8.1 Administration Guide* is available in the following collection:

   `http://docs.sun.com/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 "Configuring Access Manager Using the `amconfig` Script" on page 102.

▼ 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
[not required for Access Manager 6.3 and later]
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 111****.**

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 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 111.**

Configuring Access Manager With Directory Server in SSL Mode

Access Manager uses the LDAPS communications protocol to provide secure communications over the network with Directory Server. LDAPS is the standard LDAP protocol that runs on top of the Secure Sockets Layer (SSL) to encrypt data. The basic steps are as follows:

Configuring Directory Server in SSL Mode

To configure Directory Server in SSL mode, you must obtain and install a server certificate, configure Directory Server to trust the CA's certificate, and then enable SSL. For the detailed steps to complete these tasks, see "Using SSL With Directory Server" in *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide*.

After you finish, or if your Directory Server is already SSL-enabled, continue with the next section to configure Access Manager to connect to the SSL-enabled Directory Server.

Configuring Access Manager to Connect to an SSL-Enabled Directory Server

After Directory Server is configured for SSL mode, you must configure Access Manager to securely connect to Directory Server. You perform some of the following steps in the Access Manager Console, and then you edit the `serverconfig.xml` and `AMConfig.properties` files.

▼ To Configure Access Manager to Connect to an SSL-Enabled Directory Server

1. **Login to the Access Manager Console as `amadmin`.**
2. **Click the Configuration tab.**
3. **Under Authentication Service Name, click LDAP.**
   On the LDAP pane:
   a. **Under Primary LDAP Server, change the Directory Server port to the SSL port.**
   b. **For SSL Access to LDAP Server, click Enabled.**
   c. **Click Save.**
4. **Click Back to Configuration and then under Authentication Service Name, click Membership.**
   On the Membership pane:
   a. **Under Primary LDAP Server, change the Directory Server port to the SSL port.**
   b. **For SSL Access to LDAP Server, click Enabled.**
   c. **Click Save.**
5. **Click Back to Configuration and then under Global Properties, click Policy Configuration.**
   On the Policy Configuration pane:
   a. **Under Primary LDAP Server, change the Directory Server port to the SSL port.**
   b. **For LDAP SSL, click Enabled.**
   c. **Click Save and log out of the console.**
6. **In the `serverconfig.xml` file, change the following values in the `<Server>` element:**
   - For `port`, specify the SSL port to which Access Manager listens (default is 636).
   - For `type`, change `SIMPLE` to `SSL`.
7. **In the `AMConfig.properties` file, set the following properties:**
   - `com.iplanet.am.directory.port=636` (if you are using the default port)
   - `com.iplanet.am.directory.ssl.enabed=true`

8 **Restart the Access Manager web container.**

More Information

Configuration File Locations

The `serverconfig.xml` and `AMConfig.properties` files are in the following directory, depending on you platform:

- Solaris systems: `/etc/opt/SUNWam/config`
- Linux systems: `/etc/opt/sun/identity/config`

◆ ◆ ◆ CHAPTER 9

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 deployments, however, you might want Access Manager to run as a non-root user. This chapter describes how to install and configure Access Manager 7.1 to run as a non-root user, including these tasks:

Creating Non-root Users

As superuser (`root`), create the non-root users and groups, if they do not already exist, that you want to run Directory Server and the Access Manager web container. The examples in this chapter use the following non-root users and groups:

- Directory Server: `dirservd` in the `dirservd` group
- Web Server: `webservd` in the `webservd` group
- Application Server: `appservd` in the `appservd` group

Using Port Numbers Lower Than 1024 on Solaris 10 Systems

On Solaris 10 systems, you can allow a non-root user to use port numbers lower than 1024, by adding the `net_privaddr` privilege to the user. The `net_privaddr` privilege allows a process to bind to a privileged port number (1-1023). Thus, on Solaris 10 systems, the `dirservd` user can start Directory Server on port 389, or the `webservd` user can start Web Server on port 80.

For example, the following commands add this privilege to the non-root users:

```
# useradd -c "Directory Server reserved UID" -d / dirservd
# groupadd dirservd
# usermod -G dirservd dirservd
# usermod -K defaultpriv=basic,net_privaddr dirservd

# useradd -c "Web Server reserved UID" -d / webservd
# groupadd webservd
# usermod -G webservd webservd
# usermod -K defaultpriv=basic,net_privaddr webservd

# useradd -c "Applicaion Server reserved UID" -d / appservd
# groupadd appservd
# usermod -G appservd appservd
# usermod -K defaultpriv=basic,net_privaddr appservd
```

Note: The `net_privaddr` privilege applies only to Solaris 10 systems. It does not apply to earlier versions of the Solaris OS or to Linux systems.

Installing Sun Java System Directory Server 6.0

Follow the next procedure to install Sun Java System Directory Server Enterprise Edition 6.0 to run as a non-root user. This procedure uses `dirservd` as the non-root user.

If you prefer, you can also use an existing Directory Server, running either as `root` or a non-root user.

For more information about Directory Server 6.0, see the following documentation collection:

`http://docs.sun.com/coll/1224.1`

▼ To Install Directory Server Enterprise Edition 6.0

1 **On the server where you want to install Directory Server, log in as or become superuser (`root`).**

2 **As superuser (`root`), install Directory Server Enterprise Edition 6.0 by running the Java ES installer with the Configure Now option.**

Set the installation values as required for your Directory Server deployment. The specific values that you must set for a non-root user include:

- On the Specify Common Server Settings page, enter the non-root user (`dirservd`) for System User and non-root group (`dirservd`) for System Group.
- On the Directory Server: Specify Instance Creation Information page, specify port numbers for the Directory Instance Port and the Directory Instance SSL Port.

Note: If you are running the Solaris 10 OS, you can use port numbers lower than 1024 by assigning the `net_privaddr` privilege to the non-root user, as described in "Using Port Numbers Lower Than 1024 on Solaris 10 Systems" on page 119.

3 **After the Java ES installer has finished, login as or become the non-root user and start the Directory Server instance. For example:**

```
> cd /opt/SUNWdsee/ds6/bin
> ./dsadm start /var/opt/SUNWdsee/DS-instance
```

All Directory Server processes should be owned by the non-root user (`dirservd`).

Installing Access Manager to Run as a Non-root User With Web Server 7.0

Follow the next procedure to install and configure Access Manager 7.1 with Sun Java System Web Server Enterprise Edition 7.0 as the web container. This procedure uses `webservd` as the non-root user in examples.

This procedure runs the Java ES installer twice:

1. You first run the installer with the Configure Now option to install and configure Web Server 7.0.
2. You run the installer with the Configure Later option to install Access Manager 7.1. Then you run the `amconfig` script to configure the Access Manager 7.1 instance.

For more information about Web Server 7.0, see the following documentation collection:

`http://docs.sun.com/coll/1308.3`

▼ To Install and Configure Access Manager with Web Server 7.0 as the Web Container

Before You Begin Consider these preliminary tasks:

- The non-roon user and group must already exist. See "Creating Non-root Users" on page 119.
- Directory Server must be installed and running. See "Installing Sun Java System Directory Server 6.0" on page 120.

1 **On the server where you want to install Web Server 7.0 and Access Manager 7.1, log in as or become superuser (`root`).**

2 **As superuser (`root`), install Web Server 7.0 by running the Java ES installer with the Configure Now option.**

Set the installation values as required for your Web Server 7.0 deployment. The specific values that you must set for a non-root user include:

- On the Specify Common Server Settings page, specify the non-root user (`webservd`) for System User and non-root group (`webservd`) for System Group.
- On the Web Server: Specify Administration Server Settings page, change the Runtime User ID to the non-root user (`webservd`).
- On the Web Server: Specify Instance Settings page, change the Runtime UNIX User ID to the non-root user (`webservd`)

3 **After the Java ES installer has finished installing Web Server 7.0, login as or become the non-root user (`webservd`).**

4 **Start the Web Server 7.0 administration server and the Web Server instance using the `startserv` script.**

Note: In the current release, if you try to start the Web Server instance using the `wadm start-instance` command, the command returns an error.

All processes should be owned by the non-root user (`webservd`).

5 **Login as or become superuser (`root`) and restart the Java ES installer to install Access Manager 7.1.**

On the Choose a Configuration Type page, select the Configure Later option.

6 **After the Java ES installer has finished, depending on your platform, change the ownership of the following directories from `root` and `other` to the non-root user (`webservd`) and non-root group (`webservd`):**

- 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 webservd:webservd /opt/SUNWma /etc/opt/SUNWma
```

7 **As superuser (`root`), change to the Access Manager 7.1 `/bin` directory, depending on your platform:**

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

8. **As superuser (`root`), make a copy of the `amsamplesilent` file to use to configure Access Manager 7.1. For example:**

   ```
   # cp -p amsamplesilent ws7nonroot_config
   ```

9. **As superuser (`root`), edit the `ws7nonroot_config` file to configure Access Manager 7.1 with Web Server 7.0 as the web container:**

   - Set the `NEW_OWNER` variable to the non-root user (`webservd`) and the `NEW_GROUP` variable to the non-root group (`webservd`).
   - Set `WEB_CONTAINER=WS` to specify Web Server 7.0 as the web container. For a description of other Web Server 7.0 variables, see "Web Container Configuration Variables" on page 40.
   - Set other Access Manager 7.1 variables, as required by your deployment. For a description of these variables, see "Access Manager Configuration Variables" on page 35.

10. **As superuser (`root`), run the `amconfig` script with the edited `ws7nonroot_config` file to configure Access Manager 7.1.**

    For example, on Solaris systems:

    ```
    # cd /opt/SUNWam/bin
    # ./amconfig -s ./ws7nonroot_config
    ```

11. **Access the Web Server 7.0 Administration Console in a browser and login as the Web Server administrator.**

12. **Select the instance on which you deployed Access Manager 7.1 and click Manage.**

Installing Access Manager to Run as a Non-root User With Application Server

Follow the next procedure to install and configure with Access Manager 7.1 with Sun Java System Application Server Enterprise Edition 8.2 as the web container. This procedure uses `appservd` as the non-root user in examples.

This procedure runs the Java ES installer twice:

1. You first run the installer with the Configure Now option to install and configure Application Server 8.2.
2. You run the installer with the Configure Later option to install Access Manager 7.1. Then you run the `amconfig` script to configure the Access Manager 7.1 instance.

For more information about Application Server 8.2, see the following documentation collection:

`http://docs.sun.com/coll/1310.3`

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

Before You Begin Consider these preliminary tasks:

- The non-roon user and group must already exist. See "Creating Non-root Users" on page 119.
- Directory Server must be installed and running. See "Installing Sun Java System Directory Server 6.0" on page 120

1 **On the server where you want to install Application Server 8.2 and Access Manager 7.1, log in as or become superuser (`root`).**

2 **As superuser (`root`), install Application Server 8.2 by running the Java ES installer with the Configure Now option.**

When you select Application Server 8.2, the installer automatically selects Message Queue 3.7 UR1.

Set the installation values as required for your Application Server 8.2 deployment. The specific values that you must set for a non-root user include:

- On the Specify 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/appservd`, the Application Server installation directory would be `/export/home/appservd/as`.
- On the Specify Common Server Settings page, enter the non-root user (`appservd`) for System User and non-root group (`appservd`) for System Group.
- On the Application Server Domain Administration Server (1 of 1) page, select port numbers for the Application Server Admin Port, JMX Port, HTTP Port, and HTTPS Port.

Note: If you are running the Solaris 10 OS, you can use port numbers lower than 1024 by assigning the `net_privaddr` privilege to the non-root user, as described in "Using Port Numbers Lower Than 1024 on Solaris 10 Systems" on page 119.

3 **After the Java ES installer has finished installing Application Server 8.2, 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/appservd/as/appserver/bin`
- Linux systems: `/export/home/appservd/as/bin`

For example, to delete the Application Server 8.2 domain:

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

4. **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 appservd:appservd /export/home/appservd/as /export/home/appservd/as_var/
   ```

5. **If you plan to use an administration password file in `asadmin` commands, as superuser (`root`), create the file.**

   The following examples use `/tmp/asAdminPWFile` as the administration password file name. Specify the passwords in this file as follows:

   - `AS_ADMIN_PASSWORD=`*application-server-admin-password*
   - `AS_MASTERPASSWORD=`*master-password*

   Caution: The administration password file contains passwords in clear text. Secure this file as appropriate for your deployment.

6. **Recreate the Application Server domain as the non-root user:**

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

      ```
      # su - appservd
      ```

   b. **Change to the `/bin` directory, depending on your platform:**

      Solaris systems: `/export/home/appservd/as/appserver/bin`

      Linux systems: `/export/home/appservd/as/bin`

   c. **Recreate the deleted domain using the `asadmin create-domain` command.**

      For example:

      ```
      ./asadmin create-domain --domaindir /export/home/appservd/as_var/domains
      --adminport 4949 --adminuser admin --passwordfile /tmp/asAdminPWFile
      --instanceport 80 --domainproperties domain.jmxPort=86:http.ssl.port=81
      --savemasterpassword=true domain1
      ...
      Domain domain1 created.
      ```

7. **As the non-root user, start the Application Server 8.2 domain that you just created using the `asadmin start-domain` command. For example:**

   ```
   ./asadmin start-domain --user admin --passwordfile /tmp/asAdminPWFile domain1
   ```

   The Application Server and Message Queue processes should be owned by the non-root user (`appservd`).

8. **To verify that the Application Server 8.2 administration instance is accessible, use the following URL:**

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

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

9 **To verify that the Application Server HTTP port is accessible, use the following URL:**

`http://fqdn:8080/`

Where *fqdn* is the fully qualified domain name.

10 **Login as or become superuser (`root`) and restart the Java ES installer to install Access Manager 7.1.**

On the Choose a Configuration Type page, select the Configure Later option.

11 **After the installation finished, as superuser (`root`), change the ownership of the following directories from `root` and `other` to the non-root user (`appservd`) and non-root group (`appservd`), depending on your platform:**

- 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 appservd:appservd /opt/SUNWma /etc/opt/SUNWma
```

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

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

13 **As superuser (`root`), make a copy of the `amsamplesilent` file to use to configure Access Manager 7.1. For example:**

```
# cp -p amsamplesilent as8nonroot_config
```

14 **As superuser (`root`), edit the `as8nonroot_config` file as follows:**

- Set `NEW_OWNER` to the non-root user (`appservd`) and `NEW_GROUP` to the non-root group (`appservd`).
- Set the `AS81_HOME` variable to the parent directory of the Application Server 8.2 `/bin` directory.
- Set `WEB_CONTAINER=AS8` to specify Application Server 8.2 as the web container. For a description of other Application Server 8.2 variables, see "Web Container Configuration Variables" on page 40.
- Set other Access Manager 7.1 variables, as required by your deployment. For a description of these variables, see "Access Manager Configuration Variables" on page 35.

15 **As superuser (`root`), run the `amconfig` script with the edited `as8nonroot_config` file to deploy Access Manager 7.1. For example:**

```
# ./amconfig -s ./as8nonroot_config
```

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.

16 **As the non-root user, change to the `/bin` directory. For example:**

Solaris systems: `/export/home/appservd/as/appserver/bin`

Linux systems: `/export/home/appservd/as/bin`

17 **As the non-root user, stop the Application Server 8.2 domain and then restart it. For example:**

```
./asadmin stop-domain domain1
./asadmin start-domain --user admin --passwordfile /tmp/asAdminPWFile domain1
```

18 **Tto verify that the Access Manager 7.1 Admin Console is accessible, use the following URL:**

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

Where *fqdn* is the fully qualified domain name.

◆ ◆ ◆ CHAPTER 10

Deploying the Client SDK

The Access Manager Client SDK allows you to implement standalone applications that can access an Access Manager server to use services such as authentication, SSO, authorization, auditing, logging, and SAML. This chapter describes these topics:

Requirements for an Access Manager Client SDK Deployment

Requirements for an Access Manager Client SDK deployment include:

- An Access Manager server must be running on a remote server. To configure the Client SDK, you will need the following information from this remote installation:
  - Protocol (`http` or `https`) used by web container instance on which the Access Manager server is deployed.
  - Fully qualified domain name (FQDN) of the host on which the Access Manager server is deployed.
  - Port on which the Access Manager server is running.
  - Deployment URI for the services web application (default is `amserver`).
  - Password encryption key used by the Access Manager server. The Access Manager Client SDK must use the same password encryption key as the Access Manager server.
- The Access Manager Client SDK can be used with a standalone application or 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 Release Notes*.

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 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 26.**

3. **If not already installed, install the web container that you plan to use for the Client SDK:**

   - Web Server or Application Server: Install the web container using the Java ES installer.
   - BEA WebLogic Server or IBM WebSphere Application Server: Follow the BEA or IBM documentation. See also Chapter 7, "Installing and Configuring Third-Party Web Containers."

     If you are not using a web container, skip this step.

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 131 for the values that you must specify during installation.

   If you are using BEA WebLogic Server or IBM WebSphere Application Server as the web container, you must use the Configure Later option.

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`.**

      On Windows systems, copy the `AMConfigurator.properties` file to `AMConfigurator-clientsdk.properties`.

For the variables that you need to set, see "Access Manager Client SDK Configuration Variables" on page 131.

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
```

On Windows systems, in the `amconfig.bat` file, change `AMConfigurator.properties` to `AMConfigurator-clientsdk.properties`, and then run the edited `amconfig.bat` file.

6 Restart the web container for the Access Manager Client SDK.

Access Manager Client SDK Configuration Variables

TABLE 10–1 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`,	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`	
`ADMIN_PORT`	Same value as the administration port of the web container on the host where the Client SDK is to be deployed.
`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"`.

TABLE 10–1 Access Manager Client SDK Configuration Variables *(Continued)*

Variable	Description
`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, 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 40. If you are not using a web container or if you don not want to configure the web container, set `WEB_CONTAINER` to one that is not installed.
`APPLICATION_USER`	User name for the application. Default: `anonymous`
`APPLICATION_PASSWD`	Password of the user for the application. Default: `anonymous`
`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 and HP-UX systems: `/var/opt/sun/identity/logs` Windows systems: *AccessManager-base*`/identity/debug`
`BASEDIR`	Base directory where the Access Manager Client SDK is installed. The default values for `BASEDIR` are: Solaris systems: `/opt` Linux and HP-UX systems: `/opt/sun` Windows systems: *AccessManager-base*
`CONSOLE_HOST`, `CONSOLE_PORT`, and `CONSOLE_PROTOCOL`	Corresponding values for the host on which the Access Manager console has been deployed.
`CONSOLE_REMOTE`	Specifies whether the Access Manager Console is on a different web container than the Access Manager server. 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
```

Running the Client SDK Samples

After you deploy the Client SDK using either the Java ES installer or the `amconfig` script with `DEPLOY_LEVEL=9`, the Client SDK samples are available in the following directory:

- Solaris systems: *AccessManager-base*`/SUNWam/war/clientsdk-samples`
- Linux and HP-UX systems: *AccessManager-base*`/identity/war/clientsdk-samples`
- Windows systems: *AccessManager-base*`\identity\war\clientsdk-samples`

To run the Client SDK command-line samples and standalone applications, follow the instructions in the `README.clientsdk file` in the following directory:

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

AccessManager-base represents the Access Manager base installation directory. The default base installation directory depends on your platform:

- Solaris systems: `/opt`
- Linux systems: `/opt/sun`

◆ ◆ ◆ CHAPTER 11

Deploying a Distributed Authentication UI Server

A Distributed Authentication UI server 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 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.

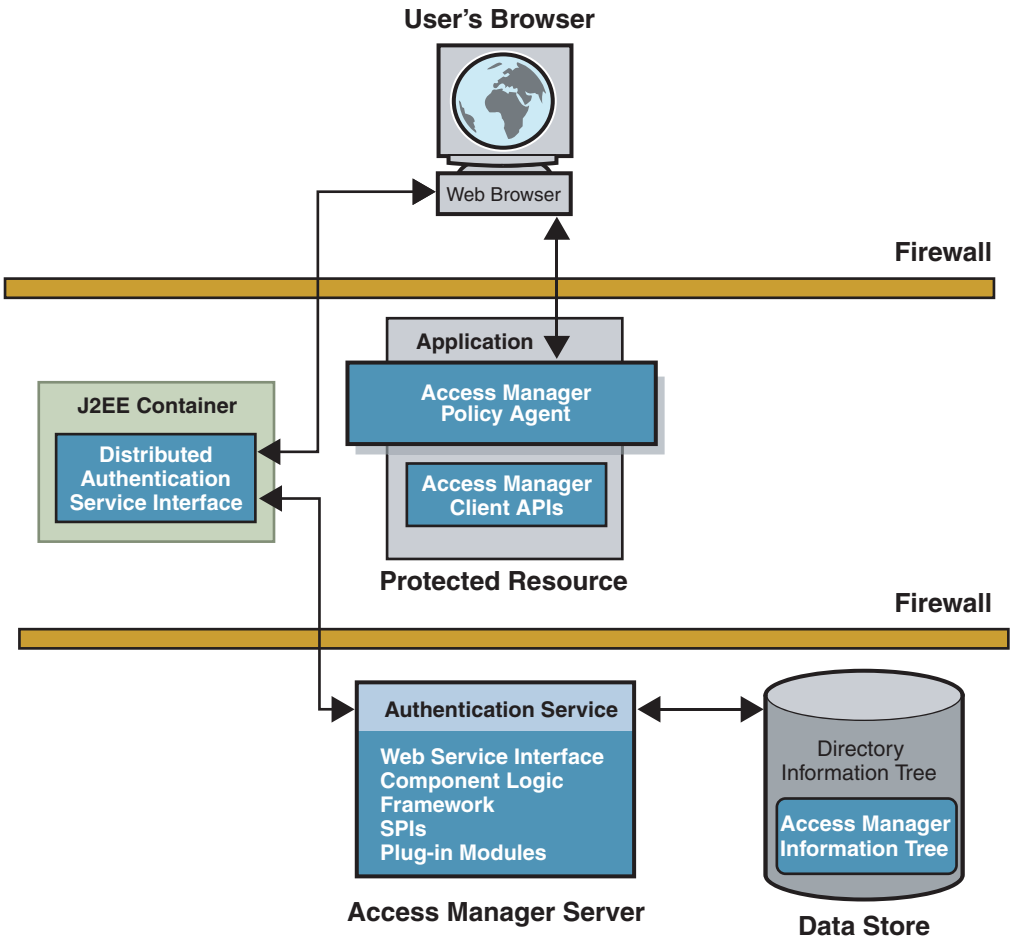

FIGURE 11–1 Distributed Authentication UI Server Deployment Scenario

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 Using the Java ES Installer

Installing and configuring (or reconfiguring) a Distributed Authentication UI server involves running the Java ES installer and the `amconfig` script on the server. One or more Access Manager full server instances must be installed and running remotely 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 26.

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 140 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`.**

On Windows systems, copy the `AMConfigurator.properties` file to `AMConfigurator-distauth.properties`.

For the variables that you need to set, see "Distributed Authentication UI Server Configuration Variables" on page 140.

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
```

On Windows systems, in the `amconfig.bat` file, change `AMConfigurator.properties` to `AMConfigurator-distauth.properties`, and then run the edited `amconfig.bat` file.

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

Example 11–1 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 11–1 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, 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 40.
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 11–1 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 and HP-UX systems: /var/opt/sun/identity/logs Windows systems: *javaes-install-dir*\identity\logs *javaes-install-dir* represents the Java ES 5 installation directory. The default value is C:\Program Files\Sun\JavaES5.
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	Specifies whether the Access Manager Console is on a different web container than the Access Manager server. The default value is false.
DISTAUTH_DEPLOY_URI	Deployment URI that will be used on the local host by the Distributed Authentication UI server. The default value is /amdistauth.

Deploying a Distributed Authentication UI Server WAR File

Deploying a Distributed Authentication UI server WAR file involves these steps:

- "Getting the amauthdistui.war File" on page 141
- "Copying and Unzipping the amDistAuth.zip File" on page 142
- "Building the amauthdistui.war File" on page 143
- "Deploying the Distributed Authentication UI Server WAR File" on page 144

You can also deploy a Distributed Authentication UI server using the Java ES installer and amconfig script. For more information, see "Installing and Configuring a Distributed Authentication UI Server Using the Java ES Installer" on page 138.

Getting the amauthdistui.war File

The amauthdistui.war file is in the amDistAuth.zip file, which is part of the Access Manager 7. 1 ZIP file.

▼ To Get the `amauthdistui.war` File:

1. **Create a new directory to download and unzip the Access Manager 7. 1 ZIP file.**
2. **Download the Access Manager 7. 1 ZIP file to the new directory you created in Step 1 from "Identity Management > Access Manager" on the following web site:**

   `http://www.sun.com/download/index.jsp`
3. **Unzip the Access Manager 7. 1 ZIP file.**

   The `amDistAuth.zip` file contains the `amauthdistui.war` file as well as other files required to configure the WAR file.

   For the layout of the Access Manager 7. 1 ZIP file, see Table 12–2.

Copying and Unzipping the `amDistAuth.zip` File

If you downloaded and unzipped the Access Manager 7. 1 ZIP file on the host server where Access Manager server is (or will be) deployed, you must copy the `amDistAuth.zip` file to the server where you plan to deploy the `amauthdistui.war` file.

▼ To Copy and Unzip the `amDistAuth.zip` File:

1. **On the server where you plan to deploy the WAR file, create a directory for the ZIP file.**
2. **Copy the `amDistAuth.zip` file to the new directory you created in Step 1.**
3. **Unzip the `amDistAuth.zip` file.**

   Table 11–2 shows the `amDistAuth.zip` file layout. The directory where you unzip the file is represented by *zip_root*.

Layout of the `amDistAuth.zip` File

TABLE 11–2 Layout of the `amDistAuth.zip` File

Directory	Description
zip_root	`README.distAuthUI` describes the contents of the ZIP file. `amauthdistui.war` is the Distributed Authentication UI server WAR file. Setup scripts are used to build the properties files and Distributed Authentication UI server web application: ■ Solaris and Linux systems: `setup.sh` ■ Windows systems: `setup.bat`
zip_root`/lib/`	`setup.jar` is a JAR file used by the setup scripts.
zip_root`/WEB-INF/classes/`	`AMConfigTemplate.properties` is the configuration template file used to update the `AMConfig.properties` file in the `amauthdistui.war` file. **Important**: Do not edit this file manually.

Building the `amauthdistui.war` File

Before you can deploy the `amauthdistui.war` file, you must run the `setup` script to add the configuration values to the `AMConfig.properties` configuration file in the `amauthdistui.war` file. The setup script uses the `WEB-INF/classes/AMConfigTemplate.properties` file to generate the `AMConfig.properties` file.

Note – Before you run the `setup` script, make sure that your `JAVA_HOME` environment variable is set to the JDK installation directory for the version of the JDK that you are using.

▼ To Build the `amauthdistui.war` File:

1. **Change to the directory on the server where you copied and unzipped the WAR file.**
2. **Change the permissions on the appropriate `setup` script to allow the script to execute:**
   - Solaris and Linux systems: `setup.sh`
   - Windows systems: `setup.bat`
3. **Invoke the appropriate `setup` script, depending on your platform.**

   For example, on Solaris systems:

   ```
   # ./setup.sh
   ```

4 **When the `setup` script prompts you, enter values for the following items:**

- Debug directory where the debug files will be created
- Application user name and password
- Access Manager server protocol. For example: `http` or `https`
- Access Manager server fully qualified host name
- Access Manager server port
- Access Manager server deployment URI. For example: `amserver`. Do not specify the slash (`/`).
- Access Manager server naming URL to get the naming service
- Distributed Authentication UI server protocol
- Distributed Authentication UI server fully qualified host name
- Distributed Authentication UI server port
- Distributed Authentication UI server deployment URI. For example: `distauth`. Do not specify the slash (`/`).
- Notification URL where notifications will be sent

After you provide these values, the `setup` script updates the `AMConfig.properties` file in the `amauthdistui.war` file.

More Information

WAR File Name

Some web containers require the WAR file name to use the same name as the deployment URI. If so, rename the `amauthdistui.war` file to the Distributed Authentication UI server deployment URI that you provided when you ran the `setup` script in the previous Step 4.

Deploying the Distributed Authentication UI Server WAR File

Deploy the Distributed Authentication UI server WAR file (`amauthdistui.war`, or the name you are using for the WAR file, if you changed the name), to one of the following web containers:

- Sun Java System Web Server 7
- Sun Java System Application Server Enterprise Edition (EE) 8.2
- BEA WebLogic Server
- IBM WebSphere Application Server

For the supported web container versions, see the *Sun Java System Access Manager 7.1 Release Notes*.

▼ To Deploy the Distributed Authentication UI Server WAR File:

Before You Begin Before you deploy the WAR file, the web container must be installed and running on the server where you plan to deploy the WAR file.

1. **Login as (or become) superuser (`root`) on the server where you plan to deploy the WAR file.**
2. **Deploy the `amauthdistui.war` file (or the name you are using for the WAR file, if you changed the name) using either the web container administration console or CLI command.**

Example 11–2 Deploying the Distributed Authentication UI Server WAR File

The following examples use the web container CLI commands. You can also deploy the WAR file using the web container administration console.

Web Server 7

If Web Server 7 is the web container, use the `wadm` command to deploy the WAR file. For example, on Solaris systems:

```
# cd /opt/SUNWwbsvr7/bin
# ./wadm add-webapp --user=admin --host=dist-auth-server-host
--port=dist-auth-port --config=web-server-configuration-name
--vs=web-server-virtual-server --uri=/dist-auth-deploy-uri
zip_root/amauthdistui.war

# ./wadm deploy-config --user=admin --host=dist-auth-server-host
--port=dist-auth-port --restart=true web-server-configuration-name
```

Enter the Web Server 7 administration password when you are prompted.

Application Server EE 8.2

If Application Server EE 8.2 is the web container, first create a password file to be used when you deploy the WAR file. For example: `/tmp/pwdfile`.

Set the following variable in the password file:

```
AS_ADMIN_PASSWORD=application-server-admin-password
```

Then, use the `asadmin deploy` command to deploy the WAR file. For example, on Solaris systems:

```
# cd /opt/SUNWappserver/appserver/bin
# ./asadmin deploy --user appserver-admin
--passwordfile /tmp/pwdfile --port 4849
--contextroot dist-auth-deploy-uri --name dist-auth-deploy-uri
--target dist-auth-server-host
zip_root/amauthdistui.war
```

See Also

- Web Server `wadm` command: Chapter 9, "Deploying Web Applications," in *Sun Java System Web Server 7.0 Developer's Guide to Java Web Applications*.
- Application Server `asadmin deploy` command: "Deploying an Application" in *Sun Java System Application Server Enterprise Edition 8.2 Quick Start Guide*
- BEA WebLogic Server documentation: `http://www.bea.com/`
- IBM WebSphere Application Server documentation: `http://www-306.ibm.com/software/webservers/appserv/was/`
- Issues and workarounds that apply to WebLogic Server or WebSphere Application Server: *Sun Java System Access Manager 7.1 Release Notes*

Tuning the Web Container

After you deploy the Distributed Authentication UI server on a web container, consider tuning the web container by running the Access Manager tuning scripts. The following tuning scripts set the JVM and other tuning options for the respective web containers:

TABLE 11–3 Access Manager Web Container Tuning Scripts

Web Container	Tuning Script Called by `amtune` Script
Web Server 7.0	`amtune-ws7`
Web Server 6.1 2005Q4 SP5	`amtune-ws61`
Application Server Enterprise Edition 8.2	`amtune-as8`
Application Server Enterprise Edition 8.1	
Application Server 7	`amtune-as7`

▼ To Tune a Web Container for a Distributed Authentication UI Server

Before You Begin Install and configure the Distributed Authentication UI server on the web container.

1. **Edit the parameters in the `amtune-env` configuration file to specify the web container and tuning options.**

   To run the script in review mode, set the `AMTUNE_MODE` variable to `REVIEW` in the `amtune-env` file.

2. **Run the `amtune` script in review mode, which calls the appropriate web container script based on values in the `amtune-env` file.**

   In review mode, the `amtune` script suggests tuning recommendations but does not make any changes to the deployment.

3. **Review the tuning recommendations in the debug log file. If needed, make changes to the `amtune-env` file based on this run.**
4. **To make tuning changes, set the `AMTUNE_MODE` variable to `CHANGE` in the `amtune-env` file.**
5. **Run the `amtune` script in change mode to make the tuning changes to the deployment.**

See Also For more information about running the `amtune` script to tune an Access Manager web container, see Chapter 2, "Access Manager Tuning Scripts," in *Sun Java System Access Manager 7.1 Performance Tuning and Troubleshooting Guide*.

Accessing the Distributed Authentication User Interface Web Application

To access the Distributed Authentication UI server web application, 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
```

Note – In a production environment, the Distributed Authentication UI server web application is usually deployed in the DMZ layer. So, always set the successful redirect URL to an absolute URL. For example: "`goto=`*absolute-successful-redirect-URL*".

For testing purposes, if you use the server returned default successful redirect URL (which is server Access Manager Admin Console URL) , make sure that you change this URL from its relative value to the absolute value before your move to a production environment by using the server Admin Console (Authentication Configuration > Properties).

◆ ◆ ◆ CHAPTER 12

Deploying Access Manager as a Single WAR File

This chapter describes how to deploy Access Manager 7.1 as an application (single WAR file), including:

Getting an Access Manager 7.1 War File

You can get an Access Manager 7.1 WAR file from the following sources:

- Downloading the Access Manager 7.1 ZIP file from the "Sun Download Site" on page 152. This ZIP file contains both the Access Manager 7.1 WAR file (`amserver.war`) and Distributed Authentication UI server WAR file (`amauthdistui.war`).
- Downloading the Access Manager 7.1 WAR file from the "Java EE 5 SDK Web Site" on page 153. A Distributed Authentication UI server WAR file is not available on this site.
- "Generating an Access Manager 7.1 WAR File Using the Java ES Installer" on page 154

Requirements for an Access Manager Single WAR File Deployment

The following table lists the requirements for creating and deploying an Access Manager WAR file.

TABLE 12–1 Requirements for a Single WAR File Deployment of Access Manager

Item	Requirement
Access Manager web container	One of the following web containers must be running on the host server where you plan to deploy an Access Manager WAR file: ■ Sun Java System Web Server 7 ■ Sun Java System Application Server Enterprise Edition 8.2 ■ BEA WebLogic Server ■ IBM WebSphere Application Server For the versions of WebLogic Server and WebSphere Application Server that are supported as web containers for Access Manager 7.1, see the *Sun Java System Access Manager 7.1 Release Notes*.
Directory Server	To store Access Manager configuration data, Directory Server Enterprise Edition 6 is required only for a production deployment. In a test or evaluation environment, you can use the `File System` option to store the Access Manager configuration data. The Java ES installer might enforce the Directory Server dependency for Access Manager, but Directory Server is not required if you select the `File System` option when you configure Access Manager after you deploy the WAR file. For more information, see "Configuring Access Manager 7.1 Using the Configurator" on page 161. **Multiple server deployment**: If you are deploying multiple Access Manager instances in a multiple server deployment: ■ All Access Manager instances must access the same instance of Directory Server. ■ The `File System` option to store the Access Manager configuration data is not supported. The Java ES 5 release includes Sun Java System Directory Server Enterprise Edition 6.

TABLE 12–1 Requirements for a Single WAR File Deployment of Access Manager *(Continued)*

Item	Requirement
Password encryption key	**Multiple server deployment**: If you are using the same WAR file to deploy multiple Access Manager instances in a multiple server deployment, you must use the same password encryption key value for each instance. Copy the encryption key value from the first instance and use this value when you configure each additional instance. You can determine this value from the `am.encryption.pwd` attribute in the `AMConfig.properties` file after you deploy the first instance.
Web container runtime user permissions	If the runtime user of the Access Manager web container instance is a non-root user, this user must be able to write to its own home directory. For example, when installing Web Server 7, the default runtime user for the Web Server instance is `webservd`. On Solaris systems, the `webservd` user has the following entry in the `/etc/passwd` file: `webservd:x:80:80:WebServer Reserved UID:/:` The `webservd` user does not have permission to write to its default home directory (`/`). Therefore, you must change the permissions to allow the `webservd` user to write to its default home directory. Otherwise, the `webservd` user will encounter an error after you configure Access Manager using the Configurator (`configurator.jsp`).
`LANG` environment variable	To run the Configurator, the code set in the `LANG` environment variable must be set to `ISO8859-1`.
Access Manager mode	An Access Manager instance deployed from an Access Manager 7.1 WAR file is always in Realm Mode (`AM_REALM=enabled`).
Sun Java Enterprise System (Java ES) installer	To generate an Access Manager 7.1 WAR file, see "Generating an Access Manager 7.1 WAR File Using the Java ES Installer" on page 154. For information about the installer, see "Overview of the Installation Process" on page 25.

Where to Find More Information

The following table shows where you can find more information if you are deploying an Access Manager 7.1 WAR file.

Component	Where to find more information
Access Manager 7.1	Access Manager 7.1 documentation collection: `http://docs.sun.com/coll/1292.2`
Access Manager Console	Access Manager 7.1 Console online help after you deploy the WAR file.

Component	Where to find more information
Sun Java System Web Server 7.0	Web Server 7.0 documentation collection: `http://docs.sun.com/coll/1308.3`
Sun Java System Application Server Enterprise Edition 8.2	Application Server EE 8.2 documentation collection: `http://docs.sun.com/coll/1310.3`
Sun Java System Directory Server 6	Directory Server 6 documentation collection: `http://docs.sun.com/coll/1224.1`

Downloading an Access Manager 7.1 WAR File

You can download a Sun Java System Access Manager 7.1 WAR file from the following sites:

- "Sun Download Site" on page 152
- "Java EE 5 SDK Web Site" on page 153

Sun Download Site

You can download an Access Manager 7.1 WAR file and a Distributed Authentication UI server WAR file as part of the Access Manager 7.1 ZIP file under "Identity Management > Access Manager" on the following web site:

`http://www.sun.com/download/index.jsp`

The ZIP file name is `AccessManager7_1`*release*`.zip`, where *release* specifies the Access Manager release. For example, `AccessManager7_1RTM.zip` is the initial release of Access Manager 7.1.

Table 12–2 describes the files in the Access Manager 7.1 ZIP file. The directory where you unzip the file is represented by *zip_root*.

TABLE 12–2 Layout of the Access Manager 7.1 ZIP File

Directory	Description
zip_root	`README` describes the contents of the ZIP file.
	`Software_License_Agt_SLA.txt` is the Software License Agreement.

TABLE 12–2 Layout of the Access Manager 7.1 ZIP File *(Continued)*

Directory	Description
zip_root/applications	README is a brief explanation of the web applications. amDistAuth.zip contains the files to configure and deploy a Distributed Authentication UI server WAR file (amauthdistui.war). For more information, see "Deploying a Distributed Authentication UI Server WAR File" on page 141.
zip_root/applications/jdk14	Contains the Access Manager 7.1 WAR file (amserver.war) for web containers running under JDK 1.4.x. For more information, see "Deploying an Access Manager 7.1 WAR File" on page 155.
zip_root/applications/jdk14/jarFix	Contains the following JAR files required for specific deployments: commons-logging.jar, dom.jar, jaxrpc-api.jar, jaxrpc-ri.jar, xalan.jar, and xercesImpl.jar.
zip_root/applications/jdk15	Contains the Access Manager 7.1 WAR file (amserver.war) for web containers running under JDK 1.5.x. For more information, see "Deploying an Access Manager 7.1 WAR File" on page 155.
zip_root/samples	README provides instructions about the Access Manager samples.
zip_root/tools	README describes the contents of the tools ZIP files. amAdminTools.zip contains: ■ Files to run the Access Manager CLI utilities and scripts such as amadmin, ampassword, amtune and amsfoconfig. ■ Properties files for various locales, including English, French, German, Spanish, Japanese, Korean, Simplified Chinese, and Traditional Chinese. amSessionTools.zip contains the files to install Sun Java System Message Queue and the Berkeley DB, which then allows you to configure Access Manager session failover.
zip_root/legal	Contains locale specific legal files

Java EE 5 SDK Web Site

You can also download a Sun Java System Access Manager 7.1 WAR file (and other components) from the Java EE 5 SDK web site:

http://java.sun.com/javaee/downloads/index.jsp

Note – This web site has the Access Manager 7.1 WAR file (`amserver.war`). If you also need the Distributed Authentication UI server WAR file (`amauthdistui.war`), download the Access Manager 7.1 ZIP file from the "Sun Download Site" on page 152.

Generating an Access Manager 7.1 WAR File Using the Java ES Installer

To generate an Access Manager 7.1 WAR file (`amserver.war`), you first install Access Manager by running the Java ES installer with the Configure Later option. You then set variables in the `amsamplesilent` file (or a copy of the file) and run the `amconfig` script.

▼ To Generate an Access Manager WAR File Using the Java ES Installer

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

2 **Install Access Manager by running the Java ES installer with the Configure Later option.**

The installer installs the `amconfig` script and `amsamplesilent` file in the following directory:

- Solaris systems: *AccessManager-base*`/SUNWam/bin`
- Linux and HP-UX systems: *AccessManager-base*`/identity/bin`
- Windows systems: *AccessManager-base*`\identity\bin`

  On Windows systems, the files are `amconfig.bat` and `AMConfigurator.properties`.

The default value for *AccessManager-base* is `/opt` on Solaris systems or `/opt/sun` on Linux systems.

3 **Make a copy of the `amsamplesilent` configuration file.**

The following examples use `amwardeploy` as the configuration file name. On Windows systems, the examples use `AMConfigurator-singlewar.properties` as the configuration file name.

4 **Set the following variables in the `amwardeploy` configuration file.**

Variable	Description
DEPLOY_LEVEL=10	Causes the amconfig script to generate an Access Manager 7.1 WAR file as follows, depending on your platform: ■ Solaris systems: /opt/SUNWam/amserver.war ■ Linux and HP-UX systems: /opt/sun/identity/amserver.war ■ Windows systems: *AccessManager-base*\identity\amserver.war
JAVA_HOME	Specifies the path to the JDK installation directory and the JDK version used by Access Manager. The JDK version must be 1.5 or later. Default: /usr/jdk/entsys-j2se
SERVER_DEPLOY_URI	Specifies the name of the new Access Manager WAR file: $SERVER_DEPLOY_URI.war Default: amserver

Note – An Access Manager instance deployed from an Access Manager 7.1 WAR file is always in Realm Mode (AM_REALM=enabled). If you set AM_REALM=disabled, the amconfig script ignores the variable.

5 **Run the** amconfig **script with the edited** amwardeploy **configuration file.**

For example, on Solaris systems with Access Manager installed in the default directory:

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

On Windows systems, in the amconfig.bat file, change AMConfigurator.properties to AMConfigurator-singlewar.properties, and then run the edited amconfig.bat file.

The amconfig script or amconfig.bat file generates the Access Manager WAR file as follows.

- Solaris systems: /opt/SUNWam/amserver.war
- Linux and HP-UX systems: /opt/sun/identity/amserver.war
- Windows systems: *AccessManager-base*\identity\amserver.war

Deploying an Access Manager 7.1 WAR File

Deploy the Access Manager 7.1 WAR file, depending on the web container you are using:

- "Deploying an Access Manager 7.1 WAR File in Sun Java System Web Server 7" on page 156
- "Deploying an Access Manager 7.1 WAR File in Sun Java System Application Server Enterprise Edition 8.2" on page 157
- "Deploying the Access Manager WAR File in BEA WebLogic Server" on page 158
- "Deploying an Access Manager 7.1 WAR File in IBM WebSphere Application Server" on page 158

- "Adding Access Manager Permissions to the Server Policy File" on page 160

Note: Samples and Javadocs are not provided after you deploy the Access Manager 7.1 WAR file.

Deploying an Access Manager 7.1 WAR File in Sun Java System Web Server 7

Before you deploy the Access Manager WAR file, Web Server 7 must be installed and running on the host server.

▼ To Deploy the Access Manager WAR File in Web Server 7

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

2 **Copy the `amserver.war file` to the host server where you want to deploy Access Manager.**

To get the `amserver.war` file, see "Getting an Access Manager 7.1 War File" on page 149.

For example, copy the WAR file to the `/opt/SUNWam/amwar_staging` directory.

3 **Backup the `server.policy` file and then add the Java security permissions to the file, as shown in "Adding Access Manager Permissions to the Server Policy File" on page 160.**

4 **Restart the Web Server instance for the new entries to take effect.**

5 **Deploy the Access Manager `amserver.war` file using the Web Server Admin Console or CLI command:**

- For example, the following Web Server 7 `wadm` command deploys the WAR file on Solaris systems:

```
cd /opt/SUNWwbsvr7/bin
./wadm add-webapp --user=admin --host=${SERVER_HOST}
--port=${WS_ADMIN_PORT} --config=${WS_CONFIG}
--vs=${WS_VIRTUAL_SERVER} --uri=/${SERVER_DEPLOY_URI}
/opt/SUNWam/amwar_staging/amserver.war

./wadm deploy-config --user admin --host=${SERVER_HOST}
--port=${WS_ADMIN_PORT} --restart=true ${WS_CONFIG}
```

Enter the Web Server administration password when you are prompted.

For more information about the `wadm` command, see Chapter 9, "Deploying Web Applications," in *Sun Java System Web Server 7.0 Developer's Guide to Java Web Applications*.

6. **Depending on your platform, add the following JavaHelp JAR file (`jhall.jar`) to the `classpath` so the Access Manager Console online help is accessible:**
   - Solaris systems: `/usr/jdk/packages/javax.help-2.0/lib/jhall.jar`
   - Linux systems: `/usr/java/packages/javax.help-2.0/javahelp/lib/jhall.jar`
7. **Continue with "Configuring Access Manager 7.1 Using the Configurator" on page 161.**

Deploying an Access Manager 7.1 WAR File in Sun Java System Application Server Enterprise Edition 8.2

Before you deploy the Access Manager WAR file, Application Server 8.2 must be installed and running on the host server.

▼ To Deploy the Access Manager 7.1 WAR File in Application Server 8.2

1. **Login as (or become) superuser (`root`).**
2. **Copy the `amserver.war file` to the host server where you want to deploy Access Manager.**

   To get the `amserver.war` file, see "Getting an Access Manager 7.1 War File" on page 149.

   For example, copy the WAR file to the `/opt/SUNWam/amwar_staging` directory.
3. **Backup the `server.policy` file and then add the Java security permissions to the file, as shown in "Adding Access Manager Permissions to the Server Policy File" on page 160.**
4. **Restart the Application Server instance for the new entries to take effect.**
5. **Create a file containing the Application Server administration password.**

   For example, if you use `/tmp/pwdfile` as the password file:

   ```
   echo "AS_ADMIN_PASSWORD=application-server-administration-password" > /tmp/pwdfile
   ```
6. **Deploy the `amserver.war` file using the Application Server Admin Console or the `asadmin deploy` command.**

   For example, the following `asadmin deploy` command deploys the WAR file on Solaris systems:

   ```
   # cd /opt/SUNWappserver/appserver/bin
   # ./asadmin deploy --user appserver-admin
   --passwordfile /tmp/pwdfile --port 4849
   --contextroot amserver --name amserver
   --target server /opt/SUNWam/amwar_staging/amserver.war
   ```
7. **Continue with "Configuring Access Manager 7.1 Using the Configurator" on page 161.**

Deploying the Access Manager WAR File in BEA WebLogic Server

Before you deploy the Access Manager WAR file, WebLogic Server must be installed and running on the host server.

For more information, see the WebLogic Server documentation: `http://www.bea.com/`.

For the versions of WebLogic Server that are supported as web containers for Access Manager 7.1, see the *Sun Java System Access Manager 7.1 Release Notes*.

Also, check the *Release Notes* for any issues and workarounds that apply to WebLogic Server.

▼ To Deploy an Access Manager 7.1 WAR File in WebLogic Server

1. **On the host server where you want to deploy Access Manager, create a staging directory for the WAR file.**

   For example, on a Solaris system: `/opt/SUNWam/amwar_staging`

2. **Copy the `amserver.war file` to the staging area.**

   To get the `amserver.war` file, see "Getting an Access Manager 7.1 War File" on page 149.

3. **Backup the `weblogic.policy` file and then add the Java security permissions to this file, as shown in "Adding Access Manager Permissions to the Server Policy File" on page 160.**

4. **Restart the WebLogic Server instance for the new entries to take effect.**

5. **Deploy the `amserver.war` file using either the WebLogic Server Admin Console or the CLI.**

6. **Depending on your platform, add the following JavaHelp JAR file (`jhall.jar`) to the `CLASSPATH` so the Access Manager Console online help is accessible:**

   - Solaris systems: `/usr/jdk/packages/javax.help-2.0/lib/jhall.jar`
   - Linux systems: `/usr/java/packages/javax.help-2.0/javahelp/lib/jhall.jar`

7. **Continue with "Configuring Access Manager 7.1 Using the Configurator" on page 161.**

Deploying an Access Manager 7.1 WAR File in IBM WebSphere Application Server

Before you deploy the Access Manager WAR file, WebSphere Application Server must be installed and running on the host server.

For more information, see the WebSphere Application Server documentation: `http://www-306.ibm.com/software/webservers/appserv/was/`.

For the versions of WebSphere Application Server that are supported as web containers for Access Manager 7.1, see the *Sun Java System Access Manager 7.1 Release Notes*.

Also, check the *Release Notes* for any issues and workarounds that apply to WebSphere Application Server.

▼ To Deploy an Access Manager 7.1 WAR File in WebSphere Application Server

1. **On the host server where you want to deploy Access Manager, create a staging directory for the WAR file.**

   For example, on a Solaris system: `/opt/SUNWam/amwar_staging`

2. **Copy the `amserver.war file` to the staging area.**

   To get the `amserver.war` file, see "Getting an Access Manager 7.1 War File" on page 149.

3. **Modify the `server.xml` file as follows:**

   a. **Add the following JVM entries to allow Access Manager to function:**

   ```
   genericJvmArguments="-Djava.awt.headless=true
   -DamCryptoDescriptor.provider=IBMJCE -DamKeyGenDescriptor.provider=IBMJCE"/>
   ```

   b. **If you are using SSL, add the following properties and JVM entry:**

   ```
   </cacheGroups>
   </services>
   <properties xmi:id="Property_1120370477732" name="amCryptoDescriptor.provider"
   value="IBMJCE" required="false"/>
   <properties xmi:id="Property_1120370511939" name="amKeyGenDescriptor.provider"
   value="IBMJCE" required="false"/>

   genericJvmArguments="-Djava.awt.headless=true
   -Djava.protocol.handler.pkgs=com.ibm.net.ssl.internal.www.protocol
   -DamCryptoDescriptor.provider=IBMJCE -DamKeyGenDescriptor.provider=IBMJCE"/>
   ```

4. **Backup the `server.policy` file and then add the Java security permissions to the file, as shown in "Adding Access Manager Permissions to the Server Policy File" on page 160.**

5. **Restart the WebSphere instance for the new entries to take effect.**

6. **Deploy the `amserver.war` file using either the WebSphere Application Server Admin Console or the CLI.**

7 **Depending on your platform, add the following JavaHelp JAR file (`jhall.jar`) to the `classpath` so the Access Manager Console online help is accessible:**

- Solaris systems: `/usr/jdk/packages/javax.help-2.0/lib/jhall.jar`
- Linux systems: `/usr/java/packages/javax.help-2.0/javahelp/lib/jhall.jar`

8 **Continue with "Configuring Access Manager 7.1 Using the Configurator" on page 161.**

Adding Access Manager Permissions to the Server Policy File

If Security Manager is enabled, add the Access Manager 7.1 permissions to the server policy file for the web container on which Access Manager will be deployed. The name of the server policy depends on the web container you are using.

EXAMPLE 12–1 Access Manager Permissions in the Server Policy File

The following permissions apply to all Access Manager web containers.

```
// ADDITIONS FOR Access Manager
grant {
  permission java.net.SocketPermission "*", "connect,accept,resolve";
  permission java.util.PropertyPermission "*", "read, write";
  permission java.lang.RuntimePermission "modifyThreadGroup";
  permission java.lang.RuntimePermission "setFactory";
  permission java.lang.RuntimePermission "accessClassInPackage.*";
  permission java.util.logging.LoggingPermission "control";
  permission java.lang.RuntimePermission "shutdownHooks";
  permission javax.security.auth.AuthPermission "getLoginConfiguration";
  permission javax.security.auth.AuthPermission "setLoginConfiguration";
  permission javax.security.auth.AuthPermission "modifyPrincipals";
  permission javax.security.auth.AuthPermission "createLoginContext.*";
  permission java.io.FilePermission "<<ALL FILES>>", "execute,delete";
  permission java.util.PropertyPermission "java.util.logging.config.class", "write";
  permission java.security.SecurityPermission "removeProvider.SUN";
  permission java.security.SecurityPermission "insertProvider.SUN";
  permission javax.security.auth.AuthPermission "doAs";
  permission java.util.PropertyPermission "java.security.krb5.realm", "write";
  permission java.util.PropertyPermission "java.security.krb5.kdc", "write";
  permission java.util.PropertyPermission "java.security.auth.login.config", "write";
  permission java.util.PropertyPermission "user.language", "write";
  permission javax.security.auth.kerberos.ServicePermission "*", "accept";
  permission javax.net.ssl.SSLPermission "setHostnameVerifier";
  permission java.security.SecurityPermission "putProviderProperty.IAIK";
  permission java.security.SecurityPermission "removeProvider.IAIK";
  permission java.security.SecurityPermission "insertProvider.IAIK";
```

EXAMPLE 12–1 Access Manager Permissions in the Server Policy File *(Continued)*

```
  permission java.security.SecurityPermission "getProperty.ocsp.*";
    };
// END OF ADDITIONS FOR Access Manager
```

Modifying the Server Policy File For Specific Applications

You can also specify that the permissions apply only to a specific application in a specific web container. For example, the following statement grants security permissions only to Access Manager deployed on Sun Java System Application Server. For other web containers, refer to the respective web container documentation for more information.

EXAMPLE 12–2 Additions to the Server Policy File For Sun Java System Application Server

```
// ADDITIONS FOR Access Manager on Sun Java System Application Server
grant codeBase "file:\${com.sun.aas.instanceRoot}/applications/j2ee-modules/amserver/-"
{

... // Permissions from the previous example

}
```

Also, if you deploy Access Manager using a name other than `amserver`, change that name in the `grant` statement.

Configuring Access Manager 7.1 Using the Configurator

Access Manager 7.1 includes the Configurator (`configurator.jsp`) to configure Access Manager after you deploy a WAR file.

Caution – Before you run the Configurator, make sure that the code set in the `LANG` environment variable is set to `ISO8859-1`. For example, to set the code set for U.S. English if you are using the `sh` or `ksh` shell:

```
# LANG=en_US.ISO8859-1
```

To launch Access Manager 7.1, specify the following URL in your browser:

`http://`*host.domain*`:`*port*`/amserver`

When you launch Access Manager 7.1, if you have not already configured the Access Manager instance, you will be directed to the Configurator page. If the Access Manager 7.1 instance is already configured successfully, you will be directed to the Access Manager Console login page.

▼ To Configure Access Manager 7.1 Using the Configurator

1 Enter the following values for the Access Manager Settings (or accept the default values).

The **Server Settings** are independent of the datastore that you select (File System or Directory Server) to store the Access Manager configuration data.

Server Settings	
Server URL	Host server where you plan to deploy Access Manager. Can be one of the following: ■ Host name. For example: `amhost1` ■ Fully qualified domain name (FQDN). For example: `http://amhost1.example.com` If you plan to use the Access Manager client SDK or a policy agent, you must specify the FQDN. ■ `localhost` Default: Host where you are deploying Access Manager.
Cookie Domain	Name of the trusted DNS domain that Access Manager returns to a browser when it grants a SSO token to a user. Specify a value only if the FQDN is used as the Server URL. For example, if the FQDN for Server URL is `http://amhost1.example.com`, the default value is `.example.com`. If you selected only the host name or `localhost` for the Server URL, Cookie Domain is set to blank, and any value you enter is ignored.
Administrator	
Name	`amAdmin` (read-only)
Password	Access Manager administrator (`amAdmin`) password. Enter and then retype to confirm the password. The password must be at least 8 characters long.
General Settings	

Configuration Directory	Base directory where the Access Manager configuration data is stored. The base directory applies to either File System or Directory Server, which you select under **Configuration Store Settings**. For example: `/am_configuration_data` Access Manager creates the following files and directories under the Configuration Directory: ■ `AMConfig.properties` file ■ `serverconfig.xml` file ■ `LDIF` files (if you select Directory Server to store the service configuration data) ■ *deploy-uri* directory ■ *deploy-uri*`/log` directory ■ *deploy-uri*`/stats` directory ■ *deploy-uri*`/debug` directory ■ *deploy-uri*`/idRepo` directory: All users are created under this directory, even if you select Directory Server to store the service configuration data, since it is the default data store. ■ `/`*deploy-uri*`/sms/` directory: Directories for the service configuration schema XML files *deploy-uri* is the Access Manager server deployment URI. The default is `/amserver`. The Access Manager instance determines the location of the Configuration Directory using the "Access Manager 7.1 Single WAR Bootstrap File" on page 165.
Platform Locale	Default language subtype for Access Manager. Default: `en_US` (US English)
Encryption Key	Random number that is used to encrypt passwords. Either accept the default encryption key value or specify a new value. The encryption key should be at least 12 characters long. **Multiple server deployment**: If you are using the same WAR file to deploy multiple Access Manager instances in a multiple server deployment, you must use the same password encryption key value for each instance. See "Requirements for an Access Manager Single WAR File Deployment" on page 150.

2 Select either of the following options to store the Access Manager configuration data:

Configuration Store Settings	
File System	Access Manager stores the service configuration data in directories under the *ConfigurationDirectory*`/amserver/sms` directory. For example: `/am_configuration_data/amserver/sms` Default is File System. **Note**: If you use an Access Manager server deployment URI other than `amserver`, that value is used instead of `amserver` for the directory name.
Directory Server	Access Manager stores the service configuration data in Sun Java System Directory Server 6. Directory Server 6 must be installed and running before you deploy the Access Manager 7.1 WAR file. **Note**: All users are created under the `/idRepo` directory, even if you select Directory Server 6 to store the service configuration data.

3 If you selected Directory Server in Step 2, provide values for the following settings:

Server Settings	
Name	Fully qualified host name of Directory Server. For example: `ds.example.com`
Port	Port at which Directory Server is running. Default: `389`
Suffix to store configuration data	Initial or root suffix in the directory where Access Manager configuration data will be stored. This value must exist in the Directory Server you are using. For example: `dc=ds,dc=example,dc=com`
Directory Server Administrator	
Directory Administrator DN	Distinguished Name (DN) of the Directory Server Administrator. Default: `cn=Directory Manager`
Password	Directory Server administrator password. Enter and then retype to confirm the password. The password must be at least eight characters long.

Load User Management Schema	Load Access Manager SDK Schema If checked, the Configurator loads the Access Manager SDK schema object classes and attributes from `sunone_schema2.ldif`, `ds_remote_schema.ldif`, `plugin.ldif`, `index.ldif` and `install.ldif` into Directory Server. Otherwise, the Configurator loads only the Access Manager service management services (SMS) object classes and attributes from the `am_sm_ds_schema.ldif` file into Directory Server.

4 **Click Configure.**

(To reset all values, click Reset.)

Next Steps The Configurator displays the configuration status:

- **Succeeded**: The Configurator displays a link to redirect you to the Access Manager Console login page. Login as `amAdmin` and the password you specified during the configuration.
- **Failed**: The Configurator displays an error message that describes the failure. If a configuration error occurred (such as an invalid password or host name), Access Manager returns to the Configurator page. Correct the error and continue. For some errors, the message will point to the Access Manager log files to help you to determine the error.

  Depending on when a failure occurs, the debug logs might not be created in their default locations. In this situation, check the logs for the following directory under the Access Manager web container:

  ```
  @BASE_DIR@@SERVER_URI@/@DEBUG_SUBDIR@
  ```

Note – If configuration was successful, you cannot reconfigure Access Manager using the Configurator. If you subsequently invoke the Configurator, Access Manager displays either the login page or the Console. If you are already logged in and have a valid session, you are redirected to the console. If you do not have a valid session, Access Manager displays the login page.

Access Manager 7.1 Single WAR Bootstrap File

An Access Manager instance deployed from a WAR file uses a bootstrap file to determine the location of its configuration data. The bootstrap file is an ASCII text file containing a single entry that specifies the location of the configuration directory for the specific Access Manager instance.

Each configured Access Manager instance on a host server has a unique bootstrap file. When you run the Configurator, a bootstrap file is created with the following name for the specific Access Manager instance:

user-home-directory/AccessManager/AMConfig_*deployed-instance-server-path_deploy-uri*

Where:

- *user-home-directory* is the home directory of the user who deployed the Access Manager instance from the WAR file.
- *deployed-instance-server-path* is the path of the deployed Access Manager instance.
- *deploy-uri* is the Access Manager server deployment URI.

For example, an Access Manager instance deployed by superuser (root) with Sun Java System Web Server 7 as the web container would have the following bootstrap file:

```
/AccessManager/AMConfig_var_opt_
SUNWwbsvr7_https-amhost.example.com_web-app_amhost.example.com_amserver
```

Each time the Access Manager web container is restarted, the Access Manager instance accesses the single WAR bootstrap file to determine the location of its configuration data. If the single WAR bootstrap file is deleted, Access Manager displays the Configurator page instead of the login page, which allows you to reconfigure the Access Manager instance.

The value in the bootstrap file is determined from the value you enter in the Configurator Configuration Directory field. For example:

```
/am_configuration_data
```

Specifying a Bootstrap File in a Different Directory

If you prefer, you can specify that the bootstrap file be created in a directory other than the user's home directory.

▼ To Specify a Bootstrap File in a Different Directory:

1. **Create a staging area for the Access Manager WAR file (amserver.war) on the host server. For example: /amwar.**

2. **Extract all files from the amserver.war file in the staging area. For example:**

   ```
   # cd /amwar
   # jar -xvf zip_root/applications/jdk15/amserver.war
   ```

   Where *zip_root* is the directory where you unzipped the Access Manager 7.1 WAR file.

3. **Add the following entry to the WEB-INF/web.xml file:**

   ```
   <context-param>
   <param-name>com.sun.identity.bootClassPath</param-name>
   <param-value>/user_defined_directory</param-value>
   </context-param>
   ```

Where *user_defined_directory* is the new location of the bootstrap file.

4 **Create a new `amserver.war` file. For example:**

```
# mkdir ../newamwar
# jar -cvf ../newamwar/amserver.war *
```

5 **Deploy the new Access Manager WAR file.**

In this example, if *user_defined_directory* is `programs`, the location of the bootstrap file would be:

```
/programs/AccessManager/AMConfig_var_opt_
SUNWwbsvr7_https-amhost.example.com_web-app_amhost.example.com_amserver
```

Considerations for an Access Manager WAR File Deployment

If you deploy an Access Manager 7.1 WAR file, consider the following:

- **Access Manager mode**. Access Manager is deployed as a single web application in Realm Mode.
- **Data Stores**. The user data store is configured to File System (flat file repository) by default, even if you specify Directory Server to store the Access Manager configuration data. The users under the File System directory are sample users. To configure a different user data store, perform the following steps:
  1. Login to the Access Manager Console.
  2. Click the realm under Realm Name.
  3. Under the realm, click Data Stores.
  4. Remove the Files data store.
  5. Add either Access Manager Repository, if you loaded the Access Manager schema during configuration, or any LDAP v3 data store.

     For information abut configuring an LDAP v3 data store, see Appendix B, "Access Manager User LDAP Entries."

     Alternatively, click Authentication under Module Instances and change to LDAP authentication instead of DataStore authentication.
- **Monitoring**. The Java Enterprise System (Java ES) monitoring framework, which is available through the Java Management Extensions (JMX), is disabled for Access Manager.
- **Client Detection**. The Client Detection service is disabled for an Access Manager WAR file deployment. If you need this feature, install Access Manager 7.1 using the Java ES installer (package-based installation).

Using the Access Manager Utilities and Scripts with an Access Manager WAR File Deployment

After you have deployed and configured the Access Manager 7.1 from the WAR file, you will probably need to perform various administrative and configurations tasks. For example, you might need to run the `amadmin` utility or to configure Access Manager session failover. The Access Manager 7.1 ZIP file provides utilities, scripts, libraries, and other supporting files in the following zip files, available for you to download:

- The `amAdminTools.zip` file contains the files to run the Access Manager CLI utilities and scripts such as `amadmin`, `ampassword`, `amtune`, and `amsfoconfig`. This zip file also contains properties files for various locales, including English, French, German, Spanish, Japanese, Korean, Simplified Chinese, and Traditional Chinese.
- The `amSessionTools.zip` file contains the files to install Sun Java System Message Queue and the Berkeley DB, which then allows you to configure Access Manager session failover.

Each zip file contains files to support the following platforms:

- Solaris SPARC and x86 based systems
- Linux systems
- Windows systems

For the specific versions that are supported for each platform, see the *Sun Java System Access Manager 7.1 Release Notes*.

Using the Utilities and Scripts in the `amAdminTools.zip` File

▼ To Use the Utilities and Scripts in the `amAdminTools.zip` File

Before You Begin

Caution – To run the `setup` utility, you must be using the Java Runtime Environment (JRE) 1.4 or later. Make sure that your `JAVA_HOME` and `PATH` environment variables point to the JDK installation directory for the version of the JDK that you are using.

1. **On Solaris and Linux systems, issue the following command before running the setup script:**

   ```
   # chmod +x setup
   ```

2. **Create a new directory to unzip the files. For example:** `amtools`

3. **Download the `amAdminTools.zip` file to the new directory and unzip the files.**

4 **In the directory (`amtools`) where you unzipped the files, run the `setup` utility.**

On Windows systems, run the `setup.bat` utility.

On Solaris and Linux systems, use this syntax to run the `setup` utility:

```
setup -p | --path aminstancedir
```

where *aminstancedir* is the path to the Access Manager configuration files, which includes the `AMConfig.properties` and `serverconfig.xml` files.

If you run the `setup` utility without any options, the script prompts you for the path to the Access Manager configuration directory.

If the path to the Access Manager configuration files contains a space, run the `setup` utility without any options and then provide the path when you are prompted.

To display the help for the `setup` utility:

```
setup -h | --help
```

Next Steps You can now run the Access Manager CLI utilities and scripts from the directory where you unzipped the `amAdminTools.zip` file.

Troubleshooting

- For more information, see the `amAdminTools.zip` README file.
- To run the `setup` utility, you must be using the JRE1.4 or later.

Using the `amSessionTools.zip` File For Access Manager Session Failover

▼ To Use the Scripts and Related Files in the `amSessionTools.zip` File

Before You Begin

Caution – To run the `setup` utility, you must be using the Java Runtime Environment (JRE) 1.4 or later. Make sure that your `JAVA_HOME` and `PATH` environment variables point to the JDK installation directory for the version of the JDK that you are using.

1 **Create a new directory to unzip the `amSessionTools.zip` file. For example: `amsfotools`**

2 **Download the `amSessionTools.zip` file to the new directory and unzip the files.**

3 **In the directory (`amsfotools`) where you unzipped the files, run the `setup` utility.**

On Windows systems, run the `setup.bat` utility.

On Solaris and Linux systems, use this syntax to run the `setup` utility:

```
setup -p | --path desireddir
```

where *desireddir* is the directory where the `setup` utility unzips the session failover scripts and related files.

If you run the `setup` utility without any options, the script prompts you for a path. If the path contains a space, run the `setup` utility without any options and then provide the path when you are prompted.

The `setup` utility preforms these functions:

- Unzips the session failover scripts and related files in the directory indicated by *desireddir*.
- Unzips the files for Sun Java System Message Queue in the *desireddir*`/jmq` directory.
- Unzips the files for BerkeleyDB in the *desireddir*`/bdb` directory.

To display the help for the `setup` utility:

```
setup -h | --help
```

Next Steps You are now ready to configure Access Manager session failover. For more information, see "Configuring Access Manager for Session Failover" on page 80.

Managing an Access Manager 7.1 WAR File Deployment

After you deploy an Access Manager WAR file, you might need to perform the following tasks:

Redeploying an Access Manager Instance

In this scenario, you want to redeploy an Access Manager instance using the web container administration console or CLI commands, without having to reconfigure the Access Manager instance.

Access Manager uses the same datastore (either Directory Server or File System) that was configured to store the configuration data before the redeployment. The location of the configuration directory is not changed.

▼ To Redeploy an Access Manager Instance

1. **Undeploy the Access Manager instance.**
2. **Restart the Access Manager web container.**

3 **Redeploy the Access Manager instance.**

After a successful redeployment, Access Manager accesses its configuration data either from Directory Server or File System by using the single WAR bootstrap file and then displays the login page.

Removing an Access Manager Instance

In this scenario, you want to completely remove an existing configured Access Manager instance that was deployed from a WAR file.

▼ To Completely Remove an Access Manager Instance

1 **Undeploy Access Manager using the web container administration console or CLI command.**

2 **Manually remove the Access Manager related additions from the server policy file.**

3 **If you deployed Access Manager on IBM WebSphere Application Server, manually remove the Access Manager related entries from the web container's `server.xml` configuration file.**

4 **From the Access Manager single WAR bootstrap file, determine the location of configuration directory for the instance.**

For information about the bootstrap file, see "Access Manager 7.1 Single WAR Bootstrap File" on page 165.

5 **Delete the Access Manager configuration directory.**

6 **Delete the Access Manager instance specific single WAR bootstrap file.**

7 **Restart the Access Manager web container for these changes to take effect.**

Migrating From File System Configuration to Directory Server Configuration

In this scenario, you deployed Access Manager from a WAR file using the File System option to store the configuration data and you want to migrate the data to Directory Server. Command-line utilities are not provided to migrate the configuration data. Directory Server must be installed and running before you perform the following steps.

▼ To Migrate From File System to Directory Server to Store Configuration Data

1. **From the Access Manager single WAR bootstrap file, determine the location of configuration directory for the instance.**

   For information about the bootstrap file, see "Access Manager 7.1 Single WAR Bootstrap File" on page 165.

2. **Delete the configuration directory for the Access Manager instance.**

3. **Restart the Access Manager web container using the web container administration console or CLI command.**

4. **Reconfigure Access Manager using the Configurator and specify Directory Server to store the configuration data.**

Uninstalling Access Manager Using the Java ES Uninstaller

Consider the following scenario:

1. You installed Access Manager 7.1 by running the Java ES installer with the Configure Later option.
2. You create an Access Manager WAR file by running the `amconfig` script with `DEPLOY_LEVEL=10`.
3. You deployed the WAR file into a web container using the container's CLI or Admin console.
4. You now want to uninstall Access Manager using the Java ES uninstaller.

The Java ES uninstaller uses the `com.sun.identity.webcontainer` property in the `AMConfig.properties` file to determine the Access Manager web container. For this scenario, this property is always set to `WEB_CONTAINER`, regardless of the web container where the Access Manager WAR file is actually deployed. During uninstallation, the uninstaller displays the Access Manager panel to gather Web Server information, even though the WAR file might be deployed on Sun Java System Application Server, BEA WebLogic Server, or IBM WebSphere Application Server.

To continue with the uninstallation, accept the default values in Access Manager Web Server uninstaller panel and click Force Uninstallation.

◆ ◆ ◆ CHAPTER 13

Changing the Password Encryption Key

Sun Java™ System Access Manager 7.1 uses a password encryption key to encrypt user passwords. All Access Manager subcomponents must use the same password encryption key value. If you plan to deploy multiple instances of Access Manager, you must use the same password encryption key for all instances.

Installation Considerations

When you install Access Manager, the Sun Java Enterprise System (Java ES) installer generates a default password encryption key string. You can either accept this default value or specify another value produced by a J2EE random number generator. The installer stores the password encryption key value in the `am.encryption.pwd` property in the `AMConfig.properties` file.

If you specify a value for the password encryption key, the string must be at least 12 characters long.

To deploy multiple instances of Access Manager, save the password encryption key value from the `am.encryption.pwd` property after you install the first instance. Then, use this key value to set the value when you deploy additional instances:

- If you run the Java ES installer, copy this value into the Password Encryption Key field on the Access Manager: Administration page.
- If you run the `amconfig` script, set the `AM_ENC_PWD` variable to this value in the `amsamplesilent` configuration file (or copy of the file) before you run the script.
- On Windows systems, if you run `amconfig.bat`, set the `AM_ENC_PWD` variable in the `AmConfigurator.properties` configuration file (or copy of the file).

Changing the Encryption Key Value

The following scenarios explain why you might need to retrieve and change the password encryption key. In these scenarios, all Access Manager instances use the same Directory Server.

- If you are doing a multiple server installation of Access Manager and you did not save the password encryption key when you installed the first Access Manager instance, you must retrieve the key to use when you deploy additional instances.
- If you have deployed an additional Access Manager instance that uses a different password encryption key from the first Access Manager instance, you must modify the encryption key value to match the first instance.

Passwords and the password encryption key must be consistent throughout a deployment. If you change a password in one place or instance, you must also update the password in all other places and instances.

The `serverconfig.xml` file contains the encrypted user passwords, which are identified by the `<DirPassword>` element. For example:

```
<DirPassword>
Adfhfghghfhdghdfhdfghrteutru
</DirPassword>
```

The `puser` and `dsameuser` passwords in `serverconfig.xml` are encrypted using the password encryption key defined in `am.encryption.pwd` in the `AMConfig.properties` file. If you change the password encryption key, you must also re-encrypt these passwords in the `serverconfig.xml` file using the `ampassword` utility (or `ampassword.bat` on Windows systems).

For information about the `ampassword` utility, see Chapter 2, "The ampassword Command Line Tool," in *Sun Java System Access Manager 7.1 Administration Reference*.

Note – If you are changing the password encryption key value on a Windows system, follow the next procedure, but run `amconfig.bat` with configuration parameters specified in the `AMConfigurator.properties` file (or a copy of the file).

▼ To change the password encryption key value

1. **Log in as or become superuser (`root`) on the host server where the first Access Manager instance is installed.**

2. **In the `AMConfig.properties` file for the first Access Manager instance, save the values of the following properties:**

- Password encryption key: `am.encryption.pwd`
- Shared secret: `com.iplanet.am.service.secret`

The `AMConfig.properties` file is installed in the following directory, depending on your platform:

- Solaris systems: `/etc/opt/SUNWam/config`
- Linux and HP-UX systems: `/etc/opt/sun/identity/config`
- Windows systems: *javaes-install-dir*`\identity\config`

  javaes-install-dir represents the Java ES 5 installation directory. The default value is `C:\Program Files\Sun\JavaES5`.

3 Log in as or become superuser (`root`) on the server where the second Access Manager instance is deployed.

4 As a precaution, back up the `AMConfig.properties` and `serverconfig.xml` files, which are in the `/config` directory.

5 Stop the web container for the second Access Manager instance.

For example, on a Solaris system, with Sun Java System Web Server as the web container:

```
# cd /opt/SUNWwbsvr/https-host2-name
#./stop
```

6 Edit the `AMConfig.properties` file and replace the values for `am.encryption.pwd` and `com.iplanet.am.service.secret` with the values that you saved from the first Access Manager instance in Step 2.

7 Because the encryption key defined in `am.encryption.pwd` is changed, you must run the `ampassword` utility to re-encrypt and replace the passwords in the `serverconfig.xml` file. The passwords in `serverconfig.xml` are identified by the `<DirPassword>` element. Consider the following cases:

Passwords are the same. If the password for `puser` and `dsameuser` is the same as the `amadmin` password in `serverconfig.xml`, run `ampassword` to re-encrypt the `amadmin` password. For example on Solaris systems:

```
# cd /opt/SUNWam/bin
# ./ampassword --encrypt password
```

where *password* is the password you used for `amadmin` when you installed the first instance. Use the `ampassword` output (new encrypted password) to replace the two passwords in the `serverconfig.xml` file for the second instance.

Passwords are different. If the passwords for `puser` and `dsameuser` are different from the `amadmin` password in `serverconfig.xml`, run `ampassword` to re-encrypt each password (`type="proxy"` and `type="admin"`).

Use the `ampassword` output (new encrypted passwords) to replace the `puser` and `dsameuser` passwords in `serverconfig.xml` for the second instance.

8 Restart the web container for the second Access Manager instance. For example, on a Solaris system, with Web Server as the web container:

```
# cd /opt/SUNWwbsvr/https-host2-name
# ./start
```

Next Steps Repeat Step 3 through Step 8 for any additional instances of Access Manager in the deployment.

◆ ◆ ◆ CHAPTER 14

Removing Access to the Access Manager Console

In this scenario, you want to remove access to the Access Manager Administration Console, to prevent unauthorized users from accessing the Console.

Removing Access to the Console

▼ To Remove Access to the Console

1 **Locate the `WEB-INF/web.xml` file for your specific web container.**

2 **In the `web.xml` file, either comment out or remove all 11 references to the Access Manager Console servlets. For example:**

```
...
<!--
       <servlet-mapping>
              <servlet-name>AuthServlet</servlet-name>
              <url-pattern>/authentication/*</url-pattern>
       </servlet-mapping>
       <servlet-mapping>
              <servlet-name>AMBaseServlet</servlet-name>
              <url-pattern>/base/*</url-pattern>
       </servlet-mapping>
       <servlet-mapping>
              <servlet-name>FSServlet</servlet-name>
              <url-pattern>/fed/*</url-pattern>
       </servlet-mapping>
       <servlet-mapping>
              <servlet-name>WSServlet</servlet-name>
              <url-pattern>/webservices/*</url-pattern>
       </servlet-mapping>
```

```
        <servlet-mapping>
                <servlet-name>SCServlet</servlet-name>
                <url-pattern>/service/*</url-pattern>
        </servlet-mapping>
        <servlet-mapping>
                <servlet-name>RMServlet</servlet-name>
                <url-pattern>/realm/*</url-pattern>
        </servlet-mapping>
        <servlet-mapping>
                <servlet-name>PMServlet</servlet-name>
                <url-pattern>/policy/*</url-pattern>
        </servlet-mapping>
        <servlet-mapping>
                <servlet-name>IDMServlet</servlet-name>
                <url-pattern>/idm/*</url-pattern>
        </servlet-mapping>
        <servlet-mapping>
                <servlet-name>UMServlet</servlet-name>
                <url-pattern>/user/*</url-pattern>
        </servlet-mapping>
        <servlet-mapping>
                <servlet-name>DelegationServlet</servlet-name>
                <url-pattern>/delegation/*</url-pattern>
        </servlet-mapping>
        <servlet-mapping>
                <servlet-name>DMServlet</servlet-name>
                <url-pattern>/dm/*</url-pattern>
        </servlet-mapping>
-->
...
```

3 Restart the web container for the changes in the edited `web.xml` file to take effect.

◆ ◆ ◆ APPENDIX A

Directory Server Considerations

Access Manager 7.1 requires a Directory Server to store user information and Access Manager configuration data. Considerations include these topics:

Configuring a Directory Server That is Not Provisioned With User Data

In this deployment scenario, you installed Access Manager by running the Java ES installer and your Directory Server is not yet provisioned with user data. In this deployment scenario, you must configure Directory Server as follows:

- "Indexing Access Manager Attributes in Directory Server" on page 182
- "Enabling the Directory Server Referential Integrity Plug-in" on page 183

You can now provision users in Directory Server for the deployment.

To perform these tasks, use either the Directory Server 6.0 Directory Service Control Center (DSCC) or the `ldapmodify` utility. For more information, see the *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide*.

Configuring a Directory Server That is Provisioned With User Data

In this deployment scenario, Sun Java System Directory Server is installed with an existing directory information tree (DIT), but the schema does not include the Sun organization and user naming attributes (that is, the `sunISManagedOrganization` object class is not in the root suffix).

You installed Access Manager 7.1 on a host server using either of these methods:

- You ran the Java ES installer with the Configure Now option but did not load the DIT into your Directory Server.
- You ran the Java ES installer with the Configure Later option and then ran the `amconfig` script with `DIRECTORY_MODE` set to 3 or 4.

In this deployment scenario, you must load the following Access Manager LDIF files into Directory Server:

LDIF File	Description
`sunone_schema2.ldif` and `ds_remote_schema.ldif`	Access Manager schema changes
`sunAMClient_schema.ldif` and `sunAMClient_data.ldif`	Access Manager client data and schema changes
`installExisting.ldif`	Access Manager entries

The Access Manager LDIF files are located in the following directory, depending on your platform:

- Solaris systems: `/etc/opt/SUNWam/config/ldif`
- Linux and HP-UX systems: `/etc/opt/sun/identity/config/ldif`
- Windows systems:*javaes-install-dir*`\identity\config\ldif`

  javaes-install-dir represents the Java ES 5 installation directory. The default value is `C:\Program Files\Sun\JavaES5`.

▼ To Configure the Directory Server Schema For Access Manager

Before You Begin To modify the Directory Server schema, you must have the appropriate Directory Server administrator privileges and know the administrator password.

To load the LDIF files, use either the Directory Service Control Center (DSCC) or the `ldapmodify` utility. For information about these options, see "Deciding When to Use DSCC and When to Use the Command Line" in *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide*.

1. **Load the `sunone_schema2.ldif` and `ds_remote_schema.ldif` files for the Access Manager schema changes.**

2. **Load the `sunAMClient_schema.ldif` and `sunAMClient_data.ldif` files for the Access Manager client data and schema changes.**

3. **In the `installExisting.ldif` file, edit the passwords (`userPassword` entry) for the following users:**

   - `puser`
   - `dsameuser`
   - `amldapuser`
   - `amAdmin`

   Note: The passwords for `puser`, `dsameuser`, and `amAdmin` and can be the same value, but the password for `amldapuser` must be a different value.

4. **Load the `installExisting.ldif` file.**

5. **Add the Directory Server indexes and enable the referential integrity plug-in, as described in the following sections:**

   - "Indexing Access Manager Attributes in Directory Server" on page 182
   - "Enabling the Directory Server Referential Integrity Plug-in" on page 183

6. **Load the Access Manager services using the `amserveradmin` script:**

   a. **Change to the directory where the `amserveradmin` script is located:**

      - Solaris systems: `/etc/opt/SUNWam/config/ums`
      - Linux systems: `/etc/opt/sun/identity/config/ums`

   b. **Check the `umsExisting.xml` file and make any changes to the naming attribute values as required for your Directory Server implementation.**

   c. **Edit the `amserveradmin` script and replace `ums.xml` with `umsExisting.xml`.**

   d. **Run the `amserveradmin` script. For example:**

      ```
      # ./amserveradmin "cn=amadmin,ou=people,dc=example,dc=com" "amadmin_password"
      ```

7 **Restart the Access Manager web container.**

You should now be able to login to the Access Manager Admin Console.

Indexing Access Manager Attributes in Directory Server

Directory Server indexes improve the performance of searches of Directory Server data. The following table lists the recommended attributes that you should consider indexing for Access Manager (if they are not already indexed).

TABLE A–1 Recommended Access Manager Attributes to Index in Directory Server

Attribute	Index Type
`nsroledn`	Equality, Presence, and Substring
`memberof`	Equality and Presence
`iplanet-am-static-group-dn`	Equality
`iplanet-am-modifiable-by`	Equality
`iplanet-am-user-federation-info-key`	Equality
`sunxmlkeyvalue`	Equality and Substring
`o`	Equality, Presence, and Substring
`ou`	Equality, Presence, and Substring
`sunPreferredDomain`	Equality, Presence, and Substring
`associatedDomain`	Equality, Presence, and Substring
`sunOrganizationAlias`	Equality, Presence, and Substring

▼ To Add Indexes to Directory Server

1 **Make sure that Directory Server is configured and running.**

2 **Add indexes using either the Directory Server Console or the `ldapmodify` command-line utility.**

See Table A–1 for a list of the recommended Access Manager attributes to index.

If you use the `ldapmodify` utility, load the Access Manager `index.ldif` file, which is available in the following directory, depending on your platform:

- Solaris systems: `/etc/opt/SUNWam/config/ldif`
- Linux and HP-UX systems: `/etc/opt/sun/identity/config/ldif`
- Windows systems:*javaes-install-dir*`\identity\config\ldif`

javaes-install-dir represents the Java ES 5 installation directory. The default value is `C:\Program Files\Sun\JavaES5`.

3 **Restart Directory Server.**

Enabling the Directory Server Referential Integrity Plug-in

When enabled, the Directory Server Referential Integrity plug-in performs integrity updates on specified attributes immediately after a delete or rename operation. This process ensures that relationships between related entries are maintained throughout the database. If the Referential Integrity plug-in is not already enabled, perform the following procedure.

▼ To Enable the Referential Integrity Plug-in

1 **Make sure that Directory Server is configured and running.**

2 **Enable the Referential Integrity plug-in using either the Directory Server Console or the `ldapmodify` command-line utility.**

If you use the `ldapmodify` utility, load the Access Manager `plugin.ldif` file, which is available in the following directory, depending on your platform:

- Solaris systems: `/etc/opt/SUNWam/config/ldif`
- Linux and HP-UX systems: `/etc/opt/sun/identity/config/ldif`
- Windows systems:*javaes-install-dir*`\identity\config\ldif`

  javaes-install-dir represents the Java ES 5 installation directory. The default value is `C:\Program Files\Sun\JavaES5`.

3 **Restart Directory Server to enable the plug-in.**

Disabling Persistent Searches in Directory Server

Access Manager uses persistent searches to receive information about Sun Java System Directory Server entries that change. By default, Access Manager creates the following persistent search connections during server startup:

- `aci` - To receive changes to the `aci` attribute, with the search using the LDAP filter (`aci=*`).
- `sm` - To receive changes in the Access Manager information tree (service management node), which includes objects with the `sunService` or `sunServiceComponent` marker object class. For example, creation of a new policy to define access privileges for a protected resource or changes to the rules, subjects, conditions, or response providers for an existing policy.

- `um` - To receive changes in the user directory (user management node). For example, changes to a user's name or address.

Persistent searches can cause performance overhead on Directory Server. If you determine that improving performance is critical in a production environment, disable persistent searches using the `com.sun.am.event.connection.disable.list` property.

Caution – Do not disable persistent searches unless the performance improvement is required for your deployment. The `com.sun.am.event.connection.disable.list` property was introduced primarily to avoid overhead on Directory Server when multiple version 2.1 J2EE agents are used, because each of these agents establishes these persistent searches. The version 2.2 J2EE agents no longer establish these persistent searches.

For example, if you disable persistent searches for changes in the user directory (`um`), the Access Manager server will not receive notifications from Directory Server. Therefore, an agent would not get notifications from Access Manager to update its local user cache with the new values for the user attribute. Then, if an application queries the agent for the user attributes, it might receive the old value for that attribute.

Or, if you know that Service Configuration changes (related to changing values to any of services such as Session Service and Authentication Services) will not happen in production environment, you can disable the persistent search to the Service Management (`sm`) component. However, if any changes do occur for any of the services, a server restart would be required. The same condition also applies to other persistent searches, as specified by the `aci` and `um` values.

▼ To Disable Persistent Searches

1 **Set the `com.sun.am.event.connection.disable.list` property in the `AMConfig.properties` file to one or more of the following values, previously described in this section: `aci`, `sm`, `um`.**

Values are case insensitive. To specify multiple values, separate each value with a comma. For example:

```
com.sun.am.event.connection.disable.list=sm,um
```

2 **Restart the Access Manager web container for the new property value to take effect.**

More Information

Enabling a Persistent Search

If you later want to enable a persistent search that you have disabled, set the property to a blank value for the specific search. For the previous example, to enable the search for Access Manager information tree (service management node) changes but leave the search disabled for user directory (user management node) changes, set the property as follows:

```
com.sun.am.event.connection.disable.list=um
```

Configuring a User Directory on a Directory Server Instance Different From the Access Manager Information Tree Node

In this deployment scenario, the Access Manager information tree is in one Sun Java System Directory Server instance, but the user directory node is in a different Directory Server instance. You want Access Manager to write to user profiles in the user directory node in order to support features such as account locking or account lockout.

In this scenario, the user directory node requires the schema that is installed into the Directory Server instance that contains the Access Manager information tree. Therefore, you must update the schema manually by loading the following two files, in order, into the Directory Server instance that contains the user directory node:

- `sunone_schema2.ldif`
- `ds_remote_schema.ldif`

These files are available in the following directory, depending on your platform:

- Solaris systems: `/etc/opt/SUNWam/config/ldif`
- Linux and HP-UX systems: `/etc/opt/sun/identity/config/ldif`
- Windows systems:*javaes-install-dir*`\identity\config\ldif`

  javaes-install-dir represents the Java ES 5 installation directory. The default value is `C:\Program Files\Sun\JavaES5`.

If you are using a directory other than Sun Java System Directory Server to store your users (for example, Microsoft® Active Directory), you must add specific object classes and attributes to that directory schema. For a list of these object classes and attributes, see Appendix B, "Access Manager User LDAP Entries."

Configuring Different Root Suffixes for the Access Manager Information Tree and User Directory Nodes

In Sun Java System Directory Server, you can separate Access Manager configuration data in the Access Manager information tree (or service management node) from the user data in the user directory (or user management node) by specifying a different root suffix for each node.

This scenario applies to deployments that want to separate the Access Manager configuration data from user data but do not support an LDAPv3 data repository. For example, deployments with Sun Java System Communications Suite products use the Access Manager SDK (AMSDK) to access user data.

If you deploying this scenario and are using the AMSDK to access user data in a Realm Mode deployment, a corresponding organization or sub-organization must exist for each realm or

sub-realm. To have Access Manager create an organization or sub-organization for each realm or sub-realm, enable the Copy Realm Configuration attribute (`sun-idrepo-amSDK-config-copyconfig-enabled`) in the Access Manager Console for the default (top-level realm).

The following figure shows the directory structure for this scenario.

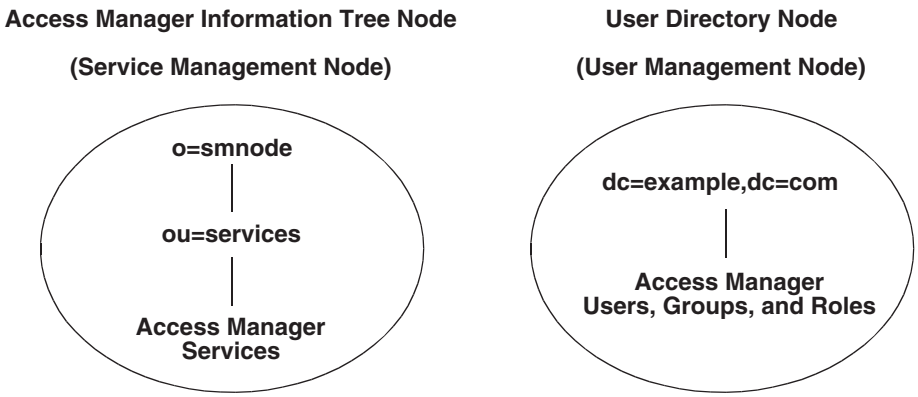

FIGURE A–1 Access Manager Information Tree and User Directory Nodes

▼ To Configure Different Root Suffixes for the Access Manager Information Tree and User Directory Nodes

To configure Access Manager with different suffixes for the Access Manager Information Tree (service management node) and user directory node, first install Access Manager by running the Java ES installer with the Configure Later option. Then, configure Access Manager by running the `amconfig` script with configuration values specified in the `amsamplesilent` file (or a copy of the file).

Important: Before you configure the two suffixes in the procedure below:

- The Access Manager Information Tree (service management node), which is specified by the `SM_CONFIG_BASEDN` variable in the `amsamplesilent` file, must exist in the directory. Create this node in the directory using a tool of your choice.
- The administrator and associated password, which are specified by the `CONFIG_ADMINDN` and `CONFIG_ADMINPASSWD` variables in the `amsamplesilent` file, must exist in the directory and must have read and write permissions to the Access Manager Information Tree (service management node), which is specified by the `SM_CONFIG_BASEDN` variable. During installation, Access Manager does not create this user or any ACIs in the directory.

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

2. **Install Access Manager by running the Java ES installer with the Configure Later option.**

3. **In the `amsamplesilent` file (or copy of the file), set the root suffixes as follows:**

   - Set the `SM_CONFIG_BASEDN` variable to the root suffix of the Access Manager information tree node (service management node).

     Note: The value indicated by `SM_CONFIG_BASEDN` must already exist in the directory, created using Directory Server tools.

   - Set the `ROOT_SUFFIX` variable to the initial or root suffix of Directory Server.

4. **Set the `CONFIG_*` variables as follows:**

   - Set `CONFIG_AD` to `false` (the default), since Sun Java System Directory Server is the configuration data store. The Directory Server schema will be loaded.
   - Set `CONFIG_SERVER` to the fully qualified domain name of the Directory Server host where the Access Manager Information Tree (service management data) is stored. The suffix on this host is indicated by the `SM_CONFIG_BASEDN` variable. The default is the value of `DS_HOST`.
   - Set `CONFIG_PORT` to the port for the Directory Server indicated by the `CONFIG_SERVER` variable. The default is the value of `DS_PORT`.
   - Set `CONFIG_ADMINDN` to the `DN` that is used to connect to the directory indicated by the `CONFIG_SERVER` variable. The default is "`cn=dsameuser,ou=DSAME Users`".
   - Set `CONFIG_ADMINPASSWD` to the password for `CONFIG_ADMINDN`. The default is the value of the `ADMINPASSWD` variable.

5. **Set any other variables in the `amsamplesilent` file (or copy of the file) as required for your deployment.**

6. **Run the `amconfig` script with the edited `amsamplesilent` file (or copy of the file).**

   For example, on a Solaris system with Access Manager installed in the default directory:

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

7. **Restart the Access Manager web container.**

Configuring Access Manager With Directory Server in MMR Mode

This deployment scenario includes the following components:

- Two Directory Server instances are installed on separate machines and configured in multi-master replication (MMR) mode. Directory Proxy Server (DPS) or a load balancer for the Directory Server instances is not used. To install the Directory Server instances, use the Java ES installer.

  The Directory Server instances used in the following examples are `ds1.example.com` and `ds2.example.com`.
- Two Access Manager instances are installed on separate host servers, accessing the Directory Server instances in MMR mode. To install the Access Manager instances, use either the Java ES installer (Realm Mode or Legacy Mode) or deploy the Access Manager 7.1 WAR file (Realm Mode only). When you install each Access Manager instance, point to the first Directory Server instance (`ds1.example.com`).

  The Access Manager instances used in the following examples are `amserver1.example.com` and `amserver2.example.com`.

  Optionally, configure the Access Manager instances for session failover, if required for your deployment. For information, see Chapter 6, "Implementing Session Failover."

Depending on whether you installed Access Manager in Realm Mode or Legacy Mode, perform the following configuration steps for **each** Access Manager instance:

- To Configure Each Access Manager Instance in Realm Mode
- To Configure Each Access Manager Instance in Legacy Mode

▼ To Configure Each Access Manager Instance in Realm Mode

Before You Begin Start the Directory Server instance (`ds1.example.com`) on the first machine only. Add the Access Manager indexes to the first Directory Server instance, as described in "Indexing Access Manager Attributes in Directory Server" on page 182.

1 Log in as or become superuser (`root`) on the server where Access Manager is installed.

2 Backup the `serverconfig.xml` file.

The `serverconfig.xml` file is in the following directory, depending on your platform:

- Solaris systems: `/etc/opt/SUNWam/config`
- Linux and HP-UX systems: `/etc/opt/sun/identity/config`
- Windows systems: `C:\Program Files\Sun\JavaES5\identity\config`

3 In the `serverconfig.xml` file, add the secondary Directory Server instance. For example:

```
...
<iPlanetDataAccessLayer>
    <ServerGroup name="default" minConnPool="1" maxConnPool="10">
```

```
                <Server name="Server1" host=" ds1.example.com" port="389" type="SIMPLE" />
                <Server name="Server2" host=" ds2.example.com" port="389" type="SIMPLE" />
...
```

4. **Login to the Access Manager Realm Mode Console as** `amadmin`**.**

5. **Click** `Access Control > Realm Name` *realm-name* `General .`

   a. **Add both Access Manager instances to the Realm/DNS Aliases list. For example:**

      ```
      amserver1.example.com
      amserver2.example.com
      ```

   b. **Save the changes.**

6. **Click** `Access Control > Realm Name` *realm-name* `> Authentication Module Instances – LDAP.`

   a. **Add the secondary Directory Server instance to Secondary LDAP Server. For example:** `ds2.example.com:389`

   b. **Save the change.**

7. **After you have performed the changes on both Access Manager instances, restart the Access Manager web container on both host servers.**

8. **On the secondary Directory Server instance, add the Access Manager indexes as follows:**

   a. **Start the secondary Directory Server instance.**

   b. **Add the Access Manager indexes using either the Directory Server 6.0 Directory Service Control Center (DSCC) or the** `ldapmodify` **utility.**

      For information about adding indexes, see "Indexing Access Manager Attributes in Directory Server" on page 182.

   c. **Restart the secondary Directory Server instance.**

▼ To Configure Each Access Manager Instance in Legacy Mode

Before You Begin Start the Directory Server instance (`ds1.example.com`) on the first machine only. Add the Access Manager indexes to the first Directory Server instance, as described in "Indexing Access Manager Attributes in Directory Server" on page 182.

1. **Log in as or become superuser (`root`) on the server where Access Manager is installed.**

2. **Backup the `serverconfig.xml` file.**

   The `serverconfig.xml` file is in the following directory, depending on your platform:

   - Solaris systems: `/etc/opt/SUNWam/config`
   - Linux and HP-UX systems: `/etc/opt/sun/identity/config`
   - Windows systems: `C:\Program Files\Sun\JavaES5\identity\config`

3. **In the `serverconfig.xml` file, add the secondary Directory Server instance. For example:**

```
...
<iPlanetDataAccessLayer>
    <ServerGroup name="default" minConnPool="1" maxConnPool="10">
            <Server name="Server1" host=" ds1.example.com" port="389" type="SIMPLE" />
            <Server name="Server2" host=" ds2.example.com" port="389" type="SIMPLE" />
...
```

4. **Login to the Access Manager Legacy Mode Console as `amadmin`.**

5. **Click `Directory Management > Organizations` *organization-name*.**

   a. **Make sure that Organization Aliases includes both Access Manager instances. Add the instances, if necessary. For example:**

      ```
      amserver1.example.com
      amserver2.example.com
      ```

   b. **Add both Access Manager instances to the DNS Aliases Names list.**

   c. **Save the changes.**

6. **Click `Configuration > Authentication Service Name – LDAP`.**

   a. **Add the secondary Directory Server instance to Secondary LDAP Server. For example:**
      `ds2.example.com:389`

   b. **Save the change.**

7. **After you have performed the changes on both Access Manager instances, restart the Access Manager web container on both host servers.**

8. **On the secondary Directory Server instance, add the Access Manager indexes as follows:**

   a. **Start the secondary Directory Server instance.**

b. **Add the Access Manager indexes using either the Directory Server 6.0 Directory Service Control Center (DSCC) or the `ldapmodify` utility.**

   For information about adding indexes, see "Indexing Access Manager Attributes in Directory Server" on page 182.

c. **Restart the secondary Directory Server instance.**

Specifying a User Naming Attribute Other Than the User ID (`uid`)

If you are using the Access Manager SDK to create users, you might want to specify an attribute other than the default user ID (`uid`) as the naming attribute. For example, you might want to use the user's email (`mail`) or common name (`cn`) attribute. Or, you might want to use a different attribute altogether, such as an application generated user ID. This section describes these topics:

Changing the Naming Attribute Before Running the `amconfig` Script

In this scenario, you install Access Manager with the Java ES installer Configure Later option and then run the `amconfig` script to set the user naming attribute (as well as other attributes). You want to change the user naming attribute before you run the `amconfig` script.

▼ To Specify a User Naming Attribute Other Than the User ID (`uid`)

1 **In the `amsamplesilent` file (or copy of the file), set the `USER_NAMING_ATTR` variable to the new attribute you want to use.**

For example, for the mail attribute: `USER_NAMING_ATTR=mail`

Specify a valid naming attribute supported by Directory Server and in the default Access Manager supported naming attribute list. Or, if the naming attribute you want to use is not in the list of Access Manager supported attributes, add the attribute to the `ums.xml` and `amUser.xml` files, as described in the following steps.

2 **In the `ums.xml` file, add the attribute to the list in the `CreationTemplate` for the `BasicUser`. For example, to use the `mail` attribute:**

```
<SubConfiguration name="CreationTemplates" >
                    <SubConfiguration name="BasicUser" id="CreationUmsObjects">
                        <AttributeValuePair> <Attribute name="name" />
```

```
            <Value>BasicUser</Value>
        </AttributeValuePair>
        <AttributeValuePair> <Attribute name="javaclass" />
            <Value>com.iplanet.ums.User</Value>
        </AttributeValuePair>
        <AttributeValuePair> <Attribute name="required" />
            <Value>objectClass=top</Value>
            <Value>objectClass=person</Value>
            <Value>objectClass=organizationalPerson</Value>
            <Value>objectClass=inetOrgPerson</Value>
            <Value>objectClass=iPlanetPreferences</Value>
            <Value>objectClass=iplanet-am-user-service</Value>
            <Value>objectClass=inetuser</Value>
            <Value>objectClass=inetAdmin</Value>
            <Value>objectClass=iplanet-am-managed-person</Value>
            <Value>objectClass=sunAMAuthAccountLockout</Value>
            <Value>cn=default</Value>
            <Value>sn=default</Value>
            <Value>uid</Value>
            <Value>inetuserstatus=Active</Value>
            <Value>mail</Value>
        </AttributeValuePair>
        <AttributeValuePair> <Attribute name="optional" />
            <Value>*</Value>
        </AttributeValuePair>
        <AttributeValuePair> <Attribute name="namingattribute" />
            <Value>uid</Value>
        </AttributeValuePair>
    </SubConfiguration>
```

3. **Also in the `ums.xml` file, add the attribute to the `BasicUserSearch` template.**

4. **In the `amUser.xml` file, add the attribute (such as `mail`) to the `<User>` schema (if it is not already in the schema).**

5. **Run the `amconfig` script with the `amsamplesilent` file (or copy of the file) from Step 1.**

Changing the Naming Attribute After Installation

In this scenario, you have installed and configured Access Manager and you want to change the user naming attribute. You must modify the `ums.xml` file and then reload the DAI service using the `amadmin` utility.

▼ To Change the Naming Attribute After Installation

1 **In the `ums.xml` file (used for the DAI service), add the attribute to the list in the `CreationTemplate` for the `BasicUser`. For example, to use the `mail` attribute:**

```
<SubConfiguration name="CreationTemplates" >
                    <SubConfiguration name="BasicUser" id="CreationUmsObjects">
                        <AttributeValuePair> <Attribute name="name" />
                            <Value>BasicUser</Value>
                        </AttributeValuePair>
                        <AttributeValuePair> <Attribute name="javaclass" />
                            <Value>com.iplanet.ums.User</Value>
                        </AttributeValuePair>
                        <AttributeValuePair> <Attribute name="required" />
                            <Value>objectClass=top</Value>
                            <Value>objectClass=person</Value>
                            <Value>objectClass=organizationalPerson</Value>
                            <Value>objectClass=inetOrgPerson</Value>
                            <Value>objectClass=iPlanetPreferences</Value>
                            <Value>objectClass=iplanet-am-user-service</Value>
                            <Value>objectClass=inetuser</Value>
                            <Value>objectClass=inetAdmin</Value>
                            <Value>objectClass=iplanet-am-managed-person</Value>
                            <Value>objectClass=sunAMAuthAccountLockout</Value>
                            <Value>cn=default</Value>
                            <Value>sn=default</Value>
                            <Value>uid</Value>
                            <Value>inetuserstatus=Active</Value>
                            <Value>mail</Value>
                        </AttributeValuePair>
                        <AttributeValuePair> <Attribute name="optional" />
                            <Value>*</Value>
                        </AttributeValuePair>
                        <AttributeValuePair> <Attribute name="namingattribute" />
                            <Value>uid</Value>
                        </AttributeValuePair>
                    </SubConfiguration>
```

2 **Delete the DAI service using the `amadmin` command. For example, on Solaris systems:**

```
# # cd /opt/SUNWam/bin
# ./amadmin -u amadmin -w amadminpassword -r DAI
```

3 **Reload the DAI service, again using the `amadmin` command. For example:**

```
# ./amadmin -u amadmin -w amadminpassword
-s /etc/opt/SUNWam/config/xml/ums.xml
```

4 **Restart the Access Manager web container.**

◆ ◆ ◆ APPENDIX B

Access Manager User LDAP Entries

A Sun Java™ System Access Manager deployment that stores users in an LDAP directory other than Sun Java System Directory Server must add the following object classes and attributes to the directory schema:

For example, if you have configured a generic LDAPv3 repository plug-in or a Microsoft® Active Directory plug-in for a realm, you must create and add the user schema to the datastore. You must perform this operation manually, because pre-populated LDIF files are not currently available to use.

Object Classes

iplanet-am-session-service Object Class

Supported by: Access Manager

Definition: Contains session service related attributes.

Superior Class: top

Object Class Type: auxiliary

Required Attributes: none

Allowed Attributes:

- “iplanet-am-session-max-session-time” on page 200
- “iplanet-am-session-max-idle-time ” on page 200
- “iplanet-am-session-max-caching-time” on page 200
- “iplanet-am-session-quota-limit” on page 200
- “iplanet-am-session-service-status” on page 201
- “iplanet-am-session-get-valid-sessions” on page 201
- “iplanet-am-session-destroy-sessions” on page 201
- “iplanet-am-session-add-session-listener-on-all-sessions” on page 201

iplanet-am-user-service Object Class

Supported by: Access Manager

Definition: Contains the Access Manager attributes necessary to manage user accounts.

Superior Class: top

Object Class Type: auxiliary

Required Attributes: none

Allowed Attributes:

- “iplanet-am-user-admin-start-dn” on page 201
- “iplanet-am-user-alias-list” on page 202
- “iplanet-am-user-auth-config” on page 202
- “sunIdentityMSISDNNumber” on page 202
- “iplanet-am-user-failure-url” on page 202
- “iplanet-am-user-success-url” on page 202
- “iplanet-am-user-login-status” on page 202
- “iplanet-am-user-password-reset-force-reset” on page 202
- “iplanet-am-user-password-reset-options” on page 203
- “iplanet-am-user-password-reset-question-answer” on page 203
- “iplanet-am-user-service-status” on page 203
- “iplanet-am-user-federation-info-key” on page 203
- “iplanet-am-user-federation-info” on page 203

iplanet-am-managed-person Object Class

Supported by: Access Manager

Definition: Contains Access Manager attributes used to manage users.

Superior Class: top

Object Class Type: auxiliary

Required Attributes: none

Allowed Attributes:

- "iplanet-am-modifiable-by" on page 203
- "iplanet-am-role-aci-description" on page 204
- "iplanet-am-static-group-dn" on page 204
- "iplanet-am-user-account-life" on page 204

sunAMAuthAccountLockout Object Class

Supported by: Access Manager

Definition: Contains Access Manager attributes used to manage invalid login attempts and user lock out.

Superior Class: top

Object Class Type: auxiliary

Required Attributes: none

Allowed Attributes:

- "sunAMAuthInvalidAttemptsData" on page 204

inetUser Object Class

Supported by: Sun One Directory Server

Definition: Auxiliary class that has to be present in an entry for delivery of subscriber services.

Superior Class: top

Object Class Type: auxiliary

Required Attributes: none

Allowed Attributes:

- “inetUserStatus” on page 205

iplanet-am-saml-service Object Class

Supported by: Access Manager

Definition: Contains SAML service related attributes.

Superior Class: top

Object Class Type: auxiliary

Required Attributes: none

Allowed Attributes:

- “iplanet-am-saml-user ” on page 205
- “iplanet-am-saml-password” on page 205

sunIdentityServerDiscoveryService Object Class

Supported by: Access Manager

Definition: Contains Discovery Service related attributes.

Superior Class: top

Object Class Type: auxiliary

Required Attributes: none

Allowed Attributes:

- “sunIdentityServerDynamicDiscoEntries” on page 205

sunIdentityServerLibertyPPService Object Class

Supported by: Access Manager

Definition: Contains session service related personal profile (PP) attributes.

Superior Class: top

Object Class Type: auxiliary

Required Attributes: none

Allowed Attributes:

Attributes

iplanet-am-session-service Object Class Attributes

iplanet-am-session-max-session-time

Syntax: string

Description: Specifies the maximum session service Time

iplanet-am-session-max-idle-time

Syntax: string

Description: Specifies the maximum session idle time.

iplanet-am-session-max-caching-time

Syntax: string

Description: Specifies the maximum session caching time.

iplanet-am-session-quota-limit

Syntax: string

Description: Specifies the session quota constraints.

iplanet-am-session-service-status

Syntax: string

Description: Specifies the maximum session service status.

iplanet-am-session-get-valid-sessions

Syntax: string

Description: Specifies the get valid sessions.

iplanet-am-session-destroy-sessions

Syntax: string

Description: Specifies destroy session.

iplanet-am-session-add-session-listener-on-all-sessions

Syntax: string

Description: Specifies add session listener on all sessions.

iplanet-am-user-service Object Class Attributes

iplanet-am-user-admin-start-dn

Supported by: Access Manager

Syntax: dn, single-valued

Description: Specifies the starting point node (DN) displayed in the starting view of the Access Manager Console when this administrator logs in.

iplanet-am-user-alias-list

Syntax: string

Description: Specifies the user alias names list.

iplanet-am-user-auth-config

Syntax: string

Description: Specifies the user authentication configuration.

sunIdentityMSISDNNumber

Syntax: string

Description: Specifies the user Mobile Station Integrated Services Digital Network (MSISDN) number.

iplanet-am-user-failure-url

Syntax: string

Description: Specifies the redirection URL for a failed user authentication.

iplanet-am-user-success-url

Syntax: string

Description: Specifies the redirection URL for a successful user authentication.

iplanet-am-user-login-status

Syntax: string, single-valued

Description: Specifies the user login status:

- `Active` - User is allowed to authenticate through the Access Manager.
- `Inactive` - User is not allowed to authenticate through the Access Manager.

iplanet-am-user-password-reset-force-reset

Syntax: string

Description: Specifies the Password Reset Force Reset password.

iplanet-am-user-password-reset-options

Supported by: Access Manager

Syntax: string, single-valued

Description: Specifies options used by the Access Manager password reset module.

iplanet-am-user-password-reset-question-answer

Supported by: Access Manager

Syntax: string, single-valued

Description: Specifies the password question and answer used to prompt a user who has forgotten the password. The format is question answer.

iplanet-am-user-service-status

Supported by: Access Manager

Syntax: dn, single-valued

Description: Specifies the status of the user for various services.

iplanet-am-user-federation-info-key

Syntax: string

Description: Specifies the user Federation information key.

iplanet-am-user-federation-info

Syntax: string

Description: Specifies user Federation information.

iplanet-am-managed-person Object Class Attributes

iplanet-am-modifiable-by

Supported by: Access Manager

Syntax: dn, multi-valued

Description: Specifies the `role-dn` of the administrator who has access rights to modify this user entry. By default, the value is set to the `role-dn` of the administrator who created the account.

iplanet-am-role-aci-description

Supported by: Access Manager

Syntax: string, multi-valued

Description: Specifies the description of the ACI that belongs to this role.

iplanet-am-static-group-dn

Supported by: Access Manager

Syntax: dn, multi-valued

Description: Defines the DNs for the static groups that this user belongs to.

iplanet-am-user-account-life

Syntax: date string, single-valued

Description: Specifies the account expiration date in the following format:

```
yyyy/mm/dd hh:mm:ss
```

sunAMAuthAccountLockout Object Class Attributes

sunAMAuthInvalidAttemptsData

Syntax: string

Description: Specifies XML data for invalid login attempts.

inetUser Object Class Attributes

inetUserStatus

Syntax: string

Possible values: "`active`", "`inactive`", or "`deleted`"

Description: Specifies the status of a user.

iplanet-am-saml-service Object Class Attributes

iplanet-am-saml-user

Syntax: string

Description: Specifies the SAML user ID.

iplanet-am-saml-password

Syntax: string

Description: Specifies the SAML user password.

sunIdentityServerDiscoveryService Object Class Attributes

sunIdentityServerDynamicDiscoEntries

Syntax: string

Description: Specifies the dynamic disco entries.

sunIdentityServerLibertyPPService Object Class Attributes

sunIdentityServerPPCommonNameCN

Syntax: string

Description: Specifies the Liberty PP common name.

sunIdentityServerPPCommonNameAltCN

Syntax: string

Description: Specifies the Liberty PP alternate common name.

sunIdentityServerPPCommonNameFN

Syntax: string

Description: Specifies the Liberty PP common name first name.

sunIdentityServerPPCommonNameSN

Syntax: string

Description: Specifies the Liberty PP common name surname.

sunIdentityServerPPCommonNamePT

Syntax: string

Description: Specifies the Liberty PP common name first name personal title.

sunIdentityServerPPCommonNameMN

Syntax: string

Description: Specifies the Liberty PP common name middle name.

sunIdentityServerPPInformalName

Syntax: string

Description: Specifies the Liberty PP informal name.

sunIdentityServerPPLegalIdentityLegalName

Syntax: string

Description: Specifies the Liberty PP legal name.

sunIdentityServerPPLegalIdentityDOB

Syntax: string

Description: Specifies the Liberty PP date of birth.

sunIdentityServerPPLegalIdentityMaritalStatus

Syntax: string

Description: Specifies the Liberty PP marital status.

sunIdentityServerPPLegalIdentityGender

Syntax: string

Description: Specifies the Liberty PP gender.

sunIdentityServerPPLegalIdentityAltIDType

Syntax: string

Description: Specifies the Liberty PP alternate identity type.

sunIdentityServerPPLegalIdentityAltIDValue

Syntax: string

Description: Specifies the Liberty PP alternate identity value.

sunIdentityServerPPLegalIdentityVATIDType

Syntax: string

Description: Specifies the Liberty PP legal identity VATID type.

sunIdentityServerPPLegalIdentityVATIDValue

Syntax: string

Description: Specifies the Liberty PP legal identity VATID value.

sunIdentityServerPPEmploymentIdentityJobTitle

Syntax: string

Description: Specifies the Liberty PP job title.

sunIdentityServerPPEmploymentIdentityOrg

Syntax: string

Description: Specifies the Liberty PP employment organization.

sunIdentityServerPPEmploymentIdentityAltO

Syntax: string

Description: Specifies the Liberty PP alternate employment organization.

sunIdentityServerPPAddressCard

Syntax: string

Description: Specifies the Liberty PP address card.

sunIdentityServerPPMsgContact

Syntax: string

Description: Specifies the Liberty PP message contact.

sunIdentityServerPPFacadeMugShot

Syntax: string

Description: Specifies the Liberty PP façade mug shot.

sunIdentityServerPPFacadeWebSite

Syntax: string

Description: Specifies the Liberty PP façade website.

sunIdentityServerPPFacadeNamePronounced

Syntax: string

Description: Specifies the Liberty PP façade name pronounced.

sunIdentityServerPPFacadeGreetSound

Syntax: string

Description: Specifies the Liberty PP façade greet sound.

sunIdentityServerPPFacadeGreetMeSound

Syntax: string

Description: Specifies the Liberty PP façade greet me sound.

sunIdentityServerPPDemographicsDisplayLanguage

Syntax: string

Description: Specifies the Liberty PP demographics display language.

sunIdentityServerPPDemographicsLanguage

Syntax: string

Description: Specifies the Liberty PP demographics language.

sunIdentityServerPPDemographicsBirthday

Syntax: string

Description: Specifies the Liberty PP demographics birthday.

sunIdentityServerPPDemographicsAge

Syntax: string

Description: Specifies the Liberty PP demographics age.

sunIdentityServerPPDemographicsTimeZone

Syntax: string

Description: Specifies the Liberty PP demographics time zone.

sunIdentityServerPPSignKey

Syntax: string

Description: Specifies the Liberty PP signing key.

sunIdentityServerPPEncryptKey

Syntax: string

Description: Specifies the Liberty PP encryption key.

sunIdentityServerPPEmergencyContact

Syntax: string

Description: Specifies the Liberty PP emergency contact.

◆ ◆ ◆ APPENDIX C

Using Active Directory as the User Data Store

This appendix describes how to use Microsoft Active Directory as the user data store for Access Manager 7.1. First review the "Overview of Using Active Directory as the User Data Store" on page 211 and check the "Requirements to Use Active Directory as the User Data Store" on page 212. Then follow the steps in these sections:

Overview of Using Active Directory as the User Data Store

By default, Access Manager 7.1 defines a set of object classes and attributes. These object classes and attributes are required in your Active Directory server if you want Access Manager to manage your Active Directory server.

The Access Manager Console provides user management functionality based on the Access Manager's predefined set of object classes and attributes, as specified through the Access Manager XML files. If the Active Directory server you are trying to access does not have these required object classes or the attributes defined, access involving the missing object class or attributes will fail, unless you change the user XML files to match the attributes defined for your Active Directory server.

For example, when you create a user via the Access Manager Console, the Console writes out to the Active Directory server the predefined set of Access Manager object classes and attributes for the user. If the Active Directory server is not configured with the same set of user object classes and attributes, the user create operation will fail. When you use the Console's user information page to edit a user's information, unless the Active Directory server has the same set of attributes and/or object classes defined for the user as Access Manager does, the operation will fail.

The Access Manager 7.1 Identity Repository (IdRepo) LDAPv3 plug-in provides attribute name mapping. You can refer to an attribute name as one name in Access Manager and a different name in your Active Directory server. As a result, you need not have all Access Manager attributes defined in Active Directory if you use attribute name mapping. However, if Access Manager has more attributes than you have in your Active Directory server, you cannot do one-to-one mapping, and some Access Manager read or write operations will fail due to missing attributes in the Active Directory server.

Requirements to Use Active Directory as the User Data Store

To use Active Directory as the user data store, your deployment must meet these requirements:

- Access Manager 7.1 requires patch 1 for your specific platform. For information about patch 1, see "Access Manager 7.1 Patch Releases" in *Sun Java System Access Manager 7.1 Release Notes*.
- Active Directory must be running on Windows Server 2003 with "Windows Server 2003 forest functional level" enabled. For more information, see:

  `http://support.microsoft.com/?id=322692#4`

Configuring Active Directory With Access Manager Schema Files

The Access Manager 7.1 Identity Repository (IdRepo) LDAPv3 plug-in must be able to assign the service's object class name to the user's object class attribute, so it can tell if a user has been assigned a given service. The following procedure describes how to load the Access Manager schema files into Active Directory and then to configure Access Manager to enable the Access Manager services.

▼ To Configure Active Directory with Access Manager Schema Files

1. **Make sure that Active Directory has "Windows Server 2003 forest functional level" enabled.**

2. **Edit the `am_remote_ad_schema.ldif` file by replacing `@ROOT_SUFFIX@` with the actual root suffix of your Active Directory installation.**

   After you have installed Access Manager 7.1 patch 1, this file is available in the following directory, depending on your platform:

   - Solaris systems: `/etc/opt/SUNWam/config/ldif`

- Linux systems: `/etc/opt/sun/identity/config/ldif`
- Windows systems: `C:\Program Files\Sun\JavaES5\identity\config\ldif`

3 **Using Active Directory tools (or another tool of your choice), load the `am_remote_ad_schema.ldif` file from the previous step into Active Directory.**

4 **In the Access Manager Administration Console:**

- Under **Attribute Name Mapping**, remove `iplanet-am-user-alias-list=objectGUID` and `portalAddress=sAMAccountName`.
- In the datastore configuration page's **LDAP User Attributes** field, add the attribute names defined in the above LDIF files.

5 **If you are writing your own service with dynamic user attributes, the `service.ldif` file for Active Directory must NOT have the following lines:**

```
dn: CN=User,CN=Schema,CN=Configuration,ROOT_SUFFIX
changetype: modify
add: auxiliaryClass
auxiliaryClass: yourClassname
```

Otherwise, Access Manager will not be able to assign the service's object class name to the user's object class attribute.

Configuring an Access Manager Identity Repository LDAPv3 Data Store For Active Directory

Using an example, this section shows how you can configure an Access Manager 7.1 Identity Repository (IdRepo) LDAPv3 data store to point a freshly installed Active Directory, including:

- "Configuration Example" on page 213
- "Operational Notes" on page 218
- "Configuring an Authentication Module to Login Through Active Directory" on page 219

Configuration Example

The following configuration example assumes:

- You have a freshly installed Active Directory.
- You have not made any changes to the Access Manager 7.1 patch 1 schema, attributes, or XML files.

Note – This section shows an example. Some additional modifications might be required for your actual environment.

In the Access Manager Administration Console, set the following Active Directory attributes. For information about an attribute, refer to the Console online Help.

Primary LDAP Server: Active Directory server name and port number that you want to connect to. For example: `myADServer.example.com:389`

LDAP Bind DN: `CN=Administrator,CN=Users,DC=example,DC=com`

LDAP Bind Password: Password for `CN=Administrator,CN=Users,DC=example,dc=com`

LDAP Organization DN: `DC=example,DC=com` — Organization `DN` that this datastore will map to. This will be the base `DN` of all operations performed in this data store.

Enable LDAP SSL: Select if the Active Directory server is in SSL mode.

LDAP Connection Pool Minimum Size: Initial number of connections in the connection pool. The use of connection pool avoids having to create a new connection each time.

LDAP Connection Pool Maximum Size: Maximum number of connections allowed.

Maximum Results Returned from Search: Maximum number of search results to return. This value should be based on the size of your LDAP organization. The maximum number returned cannot exceed the `ns` size limit configured for the Active Directory server.

Search Timeout: Maximum time in seconds to wait for results on a search operation.

LDAP Follows Referral: Option specifying whether or not referrals to other LDAP servers are followed automatically.

LDAPv3 Repository Plugin Class Name: Where to find the class file that implements the LDAPv3 repository.

Attribute Name Mapping: Allows for common attributes known to the framework to be mapped to the native data store. Map the attributes as follows:

- `mail=userPrincipalName`
- `iplanet-am-user-alias-list=objectGUID`
- `employeeNumber=distinguishedName`
- `uid=sAMAccountName`
- `portalAddress=sAMAccountName`
- `telephonenumber=displayName`

LDAPv3 Plugin Supported Types and Operations: No change is needed.

LDAP Users Search Attribute: cn — Naming attribute of user.

LDAP Users Search Filter: `(objectclass=person)`

LDAP User Object Class: Object classes for user. When a user is created, this list of user object classes will be added to the user's attributes list. Therefore, it is important that the object classes you entered here actually exist in the Active Directory server; otherwise, you will get an object class violation (error=65).

Enter the following object classes (names are not case sensitive):

- `top`
- `person`
- `organizationalPerson`
- `user`

LDAP User Attributes: Definitive list of attributes associated with a user. If an attribute is not on this list, it will not be sent or read. Therefore, if there is any possibility that the user entry can contain this attribute, you should list it here. Or, if the attribute is not defined in the Active Directory server, you should not enter it here; otherwise, you will get an error when Access Manager tries to write this attribute to Active Directory. Enter the following attributes (names are not case sensitive):

- `cn`, `description`, `displayName`, `distinguishedName`, `dn`, `employeeNumber`, `givenName`, `mail`, `manager`, `memberOf`, `name`, `objectClass`, `objectGUID`, `postalAddress`, `sAMAccountName`, `sAMAccountType`, `sn`, `streetAddress`, `telephoneNumber`, `userAccountControl`, `userpassword`, `userPrincipalname`
- `iplanet-am-auth-configuration`, `iplanet-am-auth-login-success-url`, `iplanet-am-auth-login-failure-url`, `iplanet-am-auth-post-login-process-class`
- `iplanet-am-session-add-session-listener-on-all-sessions`, `iplanet-am-session-get-valid-sessions`, `iplanet-am-session-destroy-sessions`, `iplanet-am-session-max-caching-time`, `iplanet-am-session-max-idle-time`, `iplanet-am-session-max-session-time`, `iplanet-am-session-quota-limit`, `iplanet-am-session-service-status`
- `iplanet-am-user-auth-modules`, `iplanet-am-user-login-status`, `iplanet-am-user-admin-start-dn`, `iplanet-am-user-auth-config`, `iplanet-am-user-alias-list`, `iplanet-am-user-success-url`, `iplanet-am-user-failure-url`, `iplanet-am-user-password-reset-options`
- `iplanet-am-user-password-reset-question-answer`, `iplanet-am-user-password-reset-force-reset`, `sunIdentityServerDiscoEntries`, `iplanet-am-user-federation-info-key`, `iplanet-am-user-federation-info` `sunIdentityMSISDNNumber`
- `iplanet-am-user-admin-start-dn`, `iplanet-am-user-account-life`, `iplanet-am-user-alias-list`, `iplanet-am-user-auth-config`, `iplanet-am-user-failure-url`, `iplanet-am-user-login-status`,

`iplanet-am-user-password-reset-force-reset`,
`iplanet-am-user-password-reset-options`,
`iplanet-am-user-password-reset-question-answer`, `iplanet-am-user-success-url`

- `sunAMAuthInvalidAttemptsData`
- `sunIdentityServerDeviceKeyValue`, `sunIdentityServerDeviceStatus`, `sunIdentityServerDeviceType`, `sunIdentityServerDeviceVersion`, `sunxmlkeyvalue`
- `sunIdentityServerPPFacadeNamePronounced`, `sunIdentityServerPPSignKey`, `sunIdentityServerPPDemographicsBirthday`, `sunIdentityServerPPCommonNameFN`, `sunIdentityServerPPDemographicsDisplayLanguage`, `sunIdentityServerPPCommonNameMN`, `sunIdentityServerPPLegalIdentityAltIDType`, `sunIdentityServerPPCommonNameAltCN`, `sunIdentityServerPPAddressCard`, `sunIdentityServerPPLegalIdentityAltIDValue`, `sunIdentityServerPPLegalIdentityMaritalStatus`, `sunIdentityServerPPLegalIdentityDOB`, `sunIdentityServerPPLegalIdentityVATIDValue`, `sunIdentityServerPPEncryptKey`, `sunIdentityServerPPMsgContact`, `sunIdentityServerPPDemographicsTimeZone`, `sunIdentityServerPPCommonNamePT`, `sunIdentityServerPPLegalIdentityGender`, `sunIdentityServerPPLegalIdentityVATIDType`, `sunIdentityServerPPDemographicsAge`, `sunIdentityServerPPFacadeGreetSound`, `sunIdentityServerPPEmploymentIdentityOrg`, `sunIdentityServerPPEmergencyContact`, `sunIdentityServerPPDemographicsLanguage`, `sunIdentityServerPPFacadeMugShot`, `sunIdentityServerPPFacadeGreetMeSound`, `sunIdentityServerPPFacadeWebSite`, `sunIdentityServerPPCommonNameCN`, `sunIdentityServerPPCommonNameSN`, `sunIdentityServerPPInformalName`, `sunIdentityServerPPEmploymentIdentityJobTitle`, `sunIdentityServerPPLegalIdentityLegalName`, `sunIdentityServerPPEmploymentIdentityAltO`

User Status Attribute: `userAccountControl` — Attribute to check to determine if a user is active or inactive. When a user is created, the default user's active or inactive status is assigned based on the value in this field:

- User Status Active Value: 544
- User Status Inactive Value: 546

LDAP Groups Search Attribute: `cn` — Naming attribute of a group. This attribute name will be used to construct the group's `dn` and search filter.

LDAP Groups Search Filter: `(objectclass=group)` — Filter employed when doing a search for groups. The LDAP Groups Search Attribute will be prepended to this field to form the actual group search filter.

LDAP Groups Container Naming Attribute: `cn` — Naming attribute for a group container if groups resides in a container; otherwise, leave it blank.

LDAP Groups Container Value: `users` — Value for the group container.

LDAP Groups Object Class: `objectclasses` for group. When a group is created, this list of group object classes will be added to the group's attributes list. Enter the following object classes (names are not case sensitive):

- `group`
- `top`

LDAP Groups Attributes: Definitive list of attributes associated with a group. Any attempt to read or write group attributes that are not on this list is not allowed. Therefore, you should enter all possible attributes. Enter the following attributes (names are not case sensitive):

- `objectClass`
- `sAMAccountName`
- `distinguishedName`
- `member`
- `objectCategory`
- `dn`
- `cn`
- `sAMAccountType`
- `name`

Attribute Name for Group Membership: `memberOf` — Name of the attribute whose values are the names of all the groups that this `dn` belongs to.

Attribute Name of Unique Member: `member` — Attribute name whose value is a `dn` belonging to this group.

Attribute Name of Group Member URL: `memberUrl` — Name of the attribute whose value is an LDAP URL that resolves to members belonging to this group.

LDAP People Container Naming Attribute: `cn` — Naming attribute of people container if user resides in a people container.

LDAP People Container Value: `users`

LDAP Agents Search Attribute: `cn` — Naming attribute of an agent. This attribute name will be used to construct the agent's `dn` and search filter.

LDAP Agents Container Naming Attribute: `cn` — Naming attribute of agent container if agent resides in an agent container.

LDAP Agents Container Value: `users` — Value of the agent container.

LDAP Agents Search Filter: `(objectClass=sunIdentityServerDevice)` — Filter employed when searching for an agent.

LDAP Agents Object Class: `ojectclasses` for agents. When an agent is created, this list of user object classes will be added to the agent's attributes list. Enter the following object classes (names are not case sensitive):

- person
- organizationalPerson
- sunIdentityServerDevice
- top

LDAP Agents Attributes: Definitive list of attributes associated with a user. Any attempt to read or write user attributes that are not on this list is not allowed. Enter the following attributes (names are not case sensitive):

- cn
- dn
- name
- objectClass
- userPassword
- sunIdentityServerDeviceVersion
- sunIdentityServerDeviceType
- sunIdentityServerDeviceKeyValue
- sunIdentityServerDeviceStatus
- sunxmlkeyvalue
- description

Persistent Search Base DN: `DC=example,DC=com` — Base `DN` to use for a persistent search. For Active Directory, this needs to be the root suffix.

Persistent Search Maximum Idle Time Before Restart: Restart the persistence search if it has been idle for this maximum allowed time. Default value is OK.

Maximum Number of Retries After Error Codes: Number of times to retry the persistent search operation if it encounters the error codes specified in LDAP Exception Error Codes to Retry On. Default value is OK.

Delay Time Between Retries: Time to wait before each retry. Applies only to a persistent search connection. Default value is OK.

LDAP Exception Error Codes to Retry On: Retry the persistent search operations if these errors are encountered. Default value is OK.

Operational Notes

The above configuration will allow you to list users and groups. It will also allow you to perform some basic user profile operations. You should be able to change the following user profile information in the Access Manager Console:

- emailaddress
- employeeNumber
- telephonenumber — Active Directory will add it.

- `postalAddress` — Home address in the Console. Active Directory will add it.
- `user` alias list

However, you cannot do the following operations because of missing attributes or object classes:

- Cannot create `firstname`, `lastname`, `fullname`.
- Cannot create a `group`.
- Cannot change the user authentication (`iplanet-am-user-auth-config`). No attribute exists.
- Cannot change the user status (`inetUserStatus`). No attribute exists.
- Cannot change the success URL (`iplanet-am-user-success-url`). No attribute exists.
- Cannot change the failure URL (`iplanet-am-user-failure-url`). No attribute exists.
- Cannot change the MSISDN number (`sunIdentityMSISDNNumber`). No attribute exists.
- Cannot create a user or agent in Access Manager Console. The user must be created in Active Directory.
- Cannot change the user or agent password. This change must be done in Active Directory.

Configuring an Authentication Module to Login Through Active Directory

▼ To Configure an Authentication Module to Login Through Active Directory

1 In the Access Manager 7.1 Administration Console, click realm for which you want to add the new authentication chain.

2 Click the Authentication tab.

3 Create a new module instance with the following data:

- Primary Active Directory server: *ADServer*:*ADServerPort*
- DN to Start User Search: `dc=example,dc=com`
- DN for Root User Bind: `cn=Administrator,cn=users,dc=`*RootUser*`,dc=com`
- Password for Root User Bind: *AdministratorPassword*
- Attribute Used to Retrieve User Profile: *sAMAccountName*
- Attributes Used to Search for a User to be Authenticated: *sAMAccountName*
- Search Scope: `SUBTREE`

4. **Create a new Authentication chaining instance:**
   a. **Add a new instance for the authentication instance created in the previous step.**
   b. **Set the criteria to Sufficient.**
5. **Change Default Authentication Chain to the new authentication chain you just created.**
6. **Click Save.**

Next Steps To login using Active Directory for authentication, specify the following URL:

`http://`*YourAccessManagerServer*`:`*port*`/amserver/UI/login?org=`*YourRealmName*

Index

T

U

V

W