Sun Java™ System

Access Manager 6 Administration Guide

2005Q1

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

Part No: 817-7647-11

Contents

About This Guide

The *Sun Java™ System Access Manager 2005Q1 Administration Guide* offers information on how manage Sun Java System Acceess Manager (formerly Sun™ ONE Identity Server) through the User and Command Line Interface.

This preface contains the following sections:

Who Should Use this Book

This *Administration Guide* is intended for use by IT administrators and software developers who implement an integrated identity management and web access platform using Sun Java System servers and software.

Readers of this guide should be familiar with the following concepts and technologies:

- Sun Java System Directory Server
- Lightweight Directory Access Protocol (LDAP) concepts

- Java™ technology
- JavaServer Pages™ (JSP) technology
- HyperText Transfer Protocol (HTTP)
- HyperText Markup Language (HTML)
- eXtensible Markup Language (XML)

Before You Read This Book

Access Manager is a component of the Sun Java Enterprise System, a software infrastructure that supports enterprise applications distributed across a network or Internet environment. You should be familiar with the documentation provided with Sun Java Enterprise System, which you can access online at:

http://docs.sun.com/prod/entsys.05q1

Because Sun Java System Directory Server is used as the data store in an Access Manager deployment, you should be familiar with the Directory Server documentation, which you can access online at:

http://docs.sun.com/coll/DirectoryServer_05q1

Conventions Used in This Book

The tables in this section describe the conventions used in this book.

Typographic Conventions

The following table describes the typographic changes used in this book.

Table 1 Typographic Conventions

Typeface	Meaning	Examples
AaBbCc123 (Monospace)	API and language elements, HTML tags, web site URLs, command names, file names, directory path names, onscreen computer output, sample code.	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`

Table 1 Typographic Conventions *(Continued)*

Typeface	Meaning	Examples
AaBbCc123 (Monospace bold)	What you type, when contrasted with onscreen computer output.	`%` **su** `Password:`
AaBbCc123 (Italic)	Book titles, new terms, words to be emphasized. A placeholder in a command or path name to be replaced with a real name or value.	Read Chapter 6 in the *User's Guide*. These are called *class* options. Do *not* save the file. The file is located in the *install-dir*`/bin` directory.

Symbols

The following table describes the symbol conventions used in this book.

Table 2 Symbol Conventions

Symbol	Description	Example	Meaning
`[ ]`	Contains optional command options.	`ls [-l]`	The `-l` option is not required.
`{ | }`	Contains a set of choices for a required command option.	`-d {y|n}`	The `-d` option requires that you use either the `y` argument or the `n` argument.
-	Joins simultaneous multiple keystrokes.	Control-A	Press the Control key while you press the A key.
+	Joins consecutive multiple keystrokes.	Ctrl+A+N	Press the Control key, release it, and then press the subsequent keys.
>	Indicates menu item selection in a graphical user interface.	File > New > Templates	From the File menu, choose New. From the New submenu, choose Templates.

Default Paths and File Names

The following table describes the default paths and file names used in this book:

Table 3 Default Paths and File Names

Term	Description
AccessManager-base	Represents the base installation directory for Access Manager. The Access Manager default base installation and product directory depends on your specific platform: Solaris™ systems: `/opt/SUNWam` Linux systems: `/opt/sun/identity`
DirectoryServer-base	Represents the base installation directory for Sun Java System Directory Server. Refer to the product documentation for the specific path name.
ApplicationServer-base	Represents the base installation directory for Sun Java System Application Server. Refer to the product documentation for the specific path name.
WebServer-base	Represents the base installation directory for Sun Java System Web Server. Refer to the product documentation for the specific path name.

Shell Prompts

The following table describes the shell prompts used in this book.

Table 4 Shell Prompts

Shell	Prompt
C shell on UNIX or Linux	*machine-name*%
C shell superuser on UNIX or Linux	*machine-name*#
Bourne shell and Korn shell on UNIX or Linux	$
Bourne shell and Korn shell superuser on UNIX or Linux	#
Windows command line	C:\

Related Documentation

To access Sun technical documentation online, go to http://docs.sun.com.

You can browse the documentation archive or search for a specific book title, part number, or subject.

Books in This Documentation Set

Table 5 Access Manager 6 2005Q1 Documentation Set

Book Title	Description
Technical Overview http://docs.sun.com/doc/817-7643	Provides a high-level overview of how Access Manager components work together to consolidate identity management and to protect enterprise assets and web-based applications. Explains basic Access Manager concepts and terminology
Deployment Planning Guide http://docs.sun.com/doc/817-7644	Provides information about planning a deployment within an existing information technology infrastructure
Administration Guide (this guide) http://docs.sun.com/doc/817-7647	Describes how to use the Access Manager console as well as manage user and service data via the command line.
Migration Guide http://docs.sun.com/doc/817-7645	Describes how to migrate existing data and Sun Java System product deployments to the latest version of Access Manager. (For instructions about installing and upgrading Access Manager and other products, see the *Sun Java Enterprise System 2005Q1 Installation Guide*.)
Performance Tuning Guide http://docs.sun.com/doc/817-7646	Describes how to tune Access Manager and its related components.
Federation Management Guide http://docs.sun.com/doc/817-7648	Provides information about Federation Management, which is based on the Liberty Alliance Project.
Developer's Guide http://docs.sun.com/doc/817-7649	Offers information on how to customize Access Manager and integrate its functionality into an organization's current technical infrastructure. Contains details about the programmatic aspects of the product and its API.
Developer's Reference http://docs.sun.com/doc/817-7650	Provides summaries of data types, structures, and functions that make up the Access Manager public C APIs.

Table 5 Access Manager 6 2005Q1 Documentation Set *(Continued)*

Book Title	Description
Release Notes http://docs.sun.com/doc/817-7642	Available after the product is released. Contains last-minute information, including a description of what is new in this current release, known problems and limitations, installation notes, and how to report issues with the software or the documentation.

Access Manager Policy Agent Documentation

Documentation for the Access Manager Policy Agents is available on the following documentation Web site:

http://docs.sun.com/coll/S1_IdServPolicyAgent_21

Policy Agents for Access Manager are available on a different schedule than the server product itself. Therefore, the documentation set for the policy agents is available outside the core set of Access Manager documentation. The following titles are included in the set:

- *Policy Agents For Web and Proxy Servers Guide* documents how to install and configure an Access Manager policy agent on various web and proxy servers. It also includes troubleshooting and information specific to each agent.
- *J2EE Policy Agents Guide* documents how to install and configure an Access Manager policy agent that can protect a variety of hosted J2EE applications. It also includes troubleshooting and information specific to each agent.
- The *Release Notes* are available online after a set of agents is released. The *Release Notes* include a description of what is new in the current release, known problems and limitations, installation notes, and how to report issues with the software or the documentation.

Other Server Documentation

For other server documentation, go to the following:

- Directory Server documentation
  http://docs.sun.com/coll/DirectoryServer_05q1
- Web Server documentation
  http://docs.sun.com/coll/WebServer_05q1

- Application Server documentation
  http://docs.sun.com/coll/ApplicationServer_05q1
- Web Proxy Server documentation
  http://docs.sun.com/prod/s1.webproxys#hic

Accessing Sun Resources Online

For product downloads, professional services, patches and support, and additional developer information, go to the following:

Download Center
http://wwws.sun.com/software/download/

Professional Services
http://www.sun.com/service/sunps/sunone/index.html

Sun Enterprise Services, Solaris Patches, and Support
http://sunsolve.sun.com/

Developer Information
http://developers.sun.com/prodtech/index.html

Contacting Sun Technical Support

If you have technical questions about this product that are not answered in the product documentation, go to:

http://www.sun.com/service/contacting.

Related Third-Party Web Site References

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.

Sun Welcomes Your Comments

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

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

For example, the title of this book is *Sun Java System Access Manager 6 2005Q1 Administration Guide*, and the part number is 817-7647.

Part I

Access Manager Configuration

This is part one of the *Sun Java™ System Access Manager 6 2005Q1 Administration Guide*. It discusses configuration options that you can perform after Access Manager installation. This part contains the following chapters:

Chapter 1

Access Manager 2005Q1 Configuration Scripts

This chapter describes how to configure and deploy Sun Java™ System Access Manager using the `amconfig` script and the sample silent mode input file (`amsamplesilent`). Topics include:

Access Manager 2005Q1 Installation Overview

For a new installation, always install the first instance of Access Manager 2005Q1 by running the Sun Java Enterprise System installer. When you run the installer, you can select either of these configuration options for Access Manager:

- The Configure Now option allows you to install and configure the first instance during the installation by the choices (or default values) that you select on the Access Manager installation panels.
- The Configure Later option installs the Access Manager 2005Q1 components, and then after installation, you must manually configure them or run the Access Manager scripts as described in Configuring and Reconfiguring an Instance of Access Manager. If you choose this option, then none of the products that you are currently installing will be configured. For example, if you choose to install Access Manager and Application Server and select the Configure Later option, neither application will be configured.

NOTE If you are installing BEA WebLogic 8.1.x or IBM WebSphere 5.1.x as the Access Manager web container, you must choose the Configure Later option when installing Access Manager. See "Installing and Configuring Third-Party Web Containers" on page 55 for more information.

For information about the installer, refer to the *Sun Java Enterprise System 2005Q1 Installation Guide* (`http://docs.sun.com/doc/819-0056`).

The Java Enterprise System installer installs the Access Manager 2005Q1 `amconfig` script and sample silent mode input file (`amsamplesilent`) in the *AccessManager-base*`/SUNWam/bin` directory on Solaris systems or the *AccessManager-base*`/identity/bin` directory on Linux systems.

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`. However, you can specify another directory, if you prefer, when you run the installer.

The `amconfig` script is a top-level script that calls other scripts as needed to perform the requested operation. For more information, see the Access Manager amconfig Script.

The sample configuration script input file (`amsamplesilent`) is a template that you can use to create the input file that you must specify when you run the `amconfig` script in silent mode.

This sample configuration script input file is an ASCII text file that contains Access Manager configuration variables. 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. The configuration variables are in the following format:

`variable-name=`*value*

For example:

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

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

CAUTION The format of the sample configuration script input file used when you run the `amconfig` script in silent mode does not follow the same format or necessarily use the same variable names as a Java Enterprise System silent installation state file. This file contains sensitive data, such as the admin password. Make sure to protect or delete this file as appropriate.

Access Manager amconfig Script Operations

After you install first instance of Access Manager using the Sun Java Enterprise System installer, you can run the `amconfig` script to perform the following operations, depending on the values of the variables in the silent mode input file:

- Deploy and configure the first instance of Access Manger or deploy and configure for additional instances of Access Manager on the same host system. For example, after you configure an additional instance of a web container, you can then deploy and configure a new Access Manager instance for that web container instance.
- Reconfigure both the first instance and any additional instances of Access Manager.

- Deploy and configure the Access Manager full server services or only the SDK services, which enables support for these products:
  - BEA WebLogic Server 6.1 SP4 and SP5
  - BEA WebLogic Server 8.1.x
  - IBM WebSphere 5.1.x
- Deploy and configure specific Access Manager components such as the console or Federation Management module.
- Uninstall instances and components of Access Manager that you deployed using the `amconfig` script.

Access Manager Sample Configuration Script Input File

After you run the Java Enterprise System installer, the Access Manager sample configuration script input file (`amsamplesilent`) is available in the *AccessManager-base*`/SUNWam/bin` directory on Solaris systems or the *AccessManager-base*`/identity/bin` directory on Linux systems.

To set configuration variables, first copy and rename the `amsamplesilent` file. Then set the variables in the copy for the operation you want to perform. For an example of this file, see "Example Configuration Script Input File" on page 52.

This sample silent mode input file contains the following configuration variables:

- Deployment Mode Variable
- Access Manager Configuration Variables
- Web Container Configuration Variables
- Directory Server Configuration Variables

Deployment Mode Variable

Table 1-1 describes the values for the required DEPLOY_LEVEL variable. This variable determines the operation you want the `amconfig` script to perform.

Table 1-1 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

Table 1-1 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
Re-install	21 = Redeploy all (console, password, services, and common) web applications.
(also referred to as re-deploy or re-configure)	26 = Undeploy all (console, password, services, and common) web applications.

Access Manager Configuration Variables

Table 1-2 describes the Access Manager configuration variables.

Table 1-2 Access Manager Configuration Variables

Variable	Description
BASEDIR	Base installation directory for Access Manager packages.
	Default: PLATFORM_DEFAULT
	For Solaris systems, PLATFORM_DEFAULT is `/opt`
	For Linux systems, PLATFORM_DEFAULT is `/opt/sun`
SERVER_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 7, this variable should match AS70_HOST.
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 7, this variable should match AS70_PORT.

Table 1-2 Access Manager Configuration Variables *(Continued)*

Variable	Description
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 7, this variable should match AS70_PROTOCOL.
CONSOLE_HOST	Fully qualified host name of the server where the console is installed. Default: Value provided for the Access Manager host (SERVER_HOST variable)
CONSOLE_PORT	Port of the web container where the console is installed and listens for connections. Default: Value provided for the Access Manager port (SERVER_PORT variable)
CONSOLE_PROTOCOL	Protocol of the web container where the console is installed. Default: Server protocol (SERVER_PROTOCOL variable)
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"`
DS_DIRMGRPASSWD	Password for the directory manager (DS_DIRMGRDN variable). See the note about special characters in the description of ADMINPASSWD.
ROOT_SUFFIX	Initial or root suffix of the directory. You must make sure that this value exists in the Directory Server you are using. See the note about special characters in the description of ADMINPASSWD.
ADMINPASSWD	Password for the administrator (`amadmin`). Must be different from the password for `amldapuser`. **Note**: If the password contains special characters such as a slash (/) or backslash (\), the special character must be enclosed by single quotes (‘). For example: `ADMINPASSWD='\\\\\####///'` However, the password cannot have a single quote as one of the actual password characters.
AMLDAPUSERPASSWD	Password for `amldapuser`. Must be different from the password for `amadmin`. See the note about special characters in the description of ADMINPASSWD.

Table 1-2 Access Manager Configuration Variables *(Continued)*

Variable	Description
CONSOLE_DEPLOY_URI	URI prefix for accessing the HTML pages, classes and JAR files associated with the Access Manager Administration Console subcomponent. Default: `/amconsole`
SERVER_DEPLOY_URI	URI prefix for accessing the HTML pages, classes, and JAR files associated with the Identity Management and Policy Services Core subcomponent. Default: `/amserver`
PASSWORD_DEPLOY_URI	URI that determines the mapping that the web container running Access Manager will use between a string you specify and a corresponding deployed application. Default: `/ampassword`
COMMON_DEPLOY_URI	URI prefix for accessing the common domain services on the web container. Default: `/amcommon`
COOKIE_DOMAIN	Names of the trusted DNS domains that Access Manager returns to a browser when it grants a session ID to a user. At least one value should be present. In general, the format is the server's domain name preceded with a period. Example: `.example.com`
JAVA_HOME	Path to the JDK installation directory. Default: `/usr/jdk/entsys-j2se.` This variable provides the JDK used by the command line interface's (such as `amadmin`) executables. The version must be 1.4.2 or later.
AM_ENC_PWD	Password encryption key: String that Access Manager uses to encrypt user passwords. Default: none. When the value is set to none, `amconfig` will generate a password encryption key for the user, so a password encryption will exist for the installation that is either specified by the user or created through `amconfig`. **Important**: If you are deploying multiple instances of Access Manager or the remote SDK, all instances must use the same password encryption key. When you deploy an additional instance, copy the value from the `am.encryption.pwd property` in the `AMConfig.properties` file for the first instance.
PLATFORM_LOCALE	Locale of the platform. Default: `en_US` (US English)
NEW_OWNER	New owner for the Access Manager files after installation. Default: `root`
NEW_GROUP	New group for the Access Manager files after installation. Default: `other` For a Linux installation, set NEW_GROUP to `root`.
XML_ENCODING	XML encoding. Default: `ISO-8859-1`
NEW_INSTANCE	Specifies whether the configuration script should deploy Access Manager to a new user-created web container instance: • true = To deploy Access Manager to a new user-created web container instance other than an instance that already exists. • false = To configure the first instance or re-configure an instance. Default: false

Web Container Configuration Variables

To specify the web container for Access Manager, set the WEB_CONTAINER variable in the silent mode input file, as described in Table 1-3.

Table 1-3 Access Manager WEB_CONTAINER Variable

Value	Web Container
WS6 (default)	Sun Java System Web Server 6.1 SP4
AS7	Sun Java System Application Server 7.0 Update 3 (Provided for compatibility with previous versions of Access Manager)
AS8	Sun Java System Application Server 8.1.x
WL6	BEA WebLogic Server 6.1 SP4 and SP5
WL8	BEA WebLogic Server 8.1
WAS4	IBM WebSphere 4.0.5 (Provided for compatibility with previous versions of Access Manager)
WAS5	IBM WebSphere 5.1

Sun Java System Web Server 6.1 SP4

Table 1-4 describes the configuration variables for Web Server 6.1 SP4 in the silent mode input file.

Table 1-4 Web Server 6.1 SP4 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 (SERVER_HOST variable)
WS61_HOME	Web Server base installation directory. Default: `/opt/SUNWwbsvr`
WS61_PROTOCOL	Protocol used by the Web Server instance set by the WS61_INSTANCE variable where Access Manager will be deployed: http or https. Default: Access Manager protocol (SERVER_PROTOCOL variable)

Table 1-4 Web Server 6.1 SP4 Configuration Variables *(Continued)*

Variable	Description
WS61_HOST	Fully qualified host name for the Web Server instance (WS61_INSTANCE variable). Default: Access Manager host instance (SERVER_HOST variable)
WS61_PORT	Port on which Web Server listens for connections. Default: Access Manager port number (SERVER_PORT 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"`
WS61_IS_SECURE	Specifies whether a secure port is enabled: • true: Secure port is enabled (HTTPS protocol). • false: Secure port is not enabled (HTTP protocol). Default: false (not enabled)

Sun Java System Application Server 7.0 Update 3

Table 1-5 describes the configuration variables for Application Server 7.0 Update 3 in the silent mode input file.

Table 1-5 Application Server 7.0 Update 3 Configuration Variables

Variable	Description
AS70_HOME	Path to the directory where Application Server 7.0 is installed. Default: `/opt/SUNWappserver7`
AS70_PROTOCOL	Protocol used by the Application Server instance: http or https. Default: Access Manager protocol (SERVER_PROTOCOL variable)
AS70_HOST	Fully qualified domain name (FQDN) on which the Application Server instance listens for connections. Default: Access Manager host (SERVER_HOST variable)
AS70_PORT	Port on which Application Server instance listens for connections. Default: Access Manager port number (SERVER_PORT variable)
AS70_ADMINPORT	Port on which the Application Server administration server listens for connections. Default: 4848

Table 1-5 Application Server 7.0 Update 3 Configuration Variables *(Continued)*

Variable	Description
AS70_ADMIN	Name of the user who administers the Application Server administration server for the domain into which Application Server is being displayed. Default: `admin`
AS70_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 ADMINPASSWD.
AS70_INSTANCE	Name of the Application Server instance that will run Access Manager. Default: `server1`
AS70_DOMAIN	Path to the Application Server directory for the domain to which you want to deploy this Access Manager instance. Default: `domain1`
AS70_INSTANCE_DIR	Path to the directory where Application Server stores files for the instance. Default: `/var/opt/SUNWappserver7/domains/domain1/server1`
AS70_DOCS_DIR	Directory where Application Server stores content documents. Default: `/var/opt/SUNWappserver7/domains/domain1/server1/docroot`
AS70_IS_SECURE	Specifies whether a secure port is enabled: • true: Secure port is enabled (HTTPS protocol). • false: Secure port is not enabled (HTTP protocol). Default: false (not enabled) During installation, if the Application Server admin port is SSL enabled, configuration will fail. Do not use the admin server in `https` mode.

Sun Java System Application Server 8.1.x

Table 1-6 describes the configuration variables for Application Server 8.1 in the silent mode input file.

Table 1-6 Application Server 8.1 Configuration Variables

Variable	Description
AS81_HOME	Path to the directory where Application Server 8.1 is installed. Default: `/usr/appserver1`
AS81_PROTOCOL	Protocol used by the Application Server instance: http or https. Default: Access Manager protocol (SERVER_PROTOCOL variable)

Table 1-6 Application Server 8.1 Configuration Variables *(Continued)*

Variable	Description
AS81_HOST	Fully qualified domain name (FQDN) on which the Application Server instance listens for connections. Default: Access Manager host (SERVER_HOST variable)
AS81_PORT	Port on which Application Server instance listens for connections. Default: Access Manager port number (SERVER_PORT 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 ADMINPASSWD.
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`
AS81_INSTANCE_DIR	Path to the directory where Application Server stores files for the instance. Default: `/var//appserver/domains/domain1`
AS81_DOCS_DIR	Directory where Application Server stores content documents. Default: `/var/appserver/domains/domain1/docroot`
AS81_IS_SECURE	Specifies whether a secure port is enabled: • true: Secure port is enabled (HTTPS protocol). • false: Secure port is not enabled (HTTP protocol). Default: false (not 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 6.1 SP4 and SP5

Table 1-7 describes the configuration variables for BEA WebLogic Server 6.1 in the silent mode input file.

Table 1-7 BEA WebLogic Server 6.1 SP4 and SP5 Configuration Variables

Variable	Description
WL61_HOME	WebLogic home directory. Default: `/export/bea61a`
WL61_PROJECT_DIR	WebLogic project directory. Default: `user_projects`
WL61_DOMAIN	WebLogic domain name. Default: `mydomain`
WL61_SERVER	WebLogic server name. Default: `myserver`
WL61_INSTANCE	WebLogic instance name. Default: WS61_HOME`/wlserver6.1`
WL61_PROTOCOL	WebLogic protocol. Default: http
WL61_HOST	WebLogic host name.
WL61_PORT	WebLogic port. Default: 7001
WL61_SSLPORT	WebLogic SSL port. Default: 7002
WL61_ADMIN	WebLogic administrator. Default: “system”
WL61_PASSWORD	WebLogic administrator password. See the note about special characters in the description of ADMINPASSWD.
WL61_JDK_HOME	WebLogic JDK home directory. Default: WS61_HOME`/jdk131`

BEA WebLogic Server 8.1

Table 1-8 describes the configuration variables for BEA WebLogic Server 8.1 in the silent mode input file.

Table 1-8 BEA WebLogic Server 8.1 Configuration Variables

Variable	Description
WL8_HOME	WebLogic home directory. Default: `/export/bea8`
WL8_PROJECT_DIR	WebLogic project directory. Default: `projects`
WL8_DOMAIN	WebLogic domain name. Default: `mydomain`
WL8_SERVER	WebLogic server name. Default: `myserver`
WL8_INSTANCE	WebLogic instance name. Default: `/export/bea8/weblogic81`
WL8_PROTOCOL	WebLogic protocol. Default: http
WL8_HOST	WebLogic host name. Default: none
WL8_PORT	WebLogic port. Default: 7001
WL8_SSLPORT	WebLogic SSL port. Default: 7002
WL8_ADMIN	WebLogic administrator. Default: `"system"`
WL8_PASSWORD	WebLogic administrator password. See the note about special characters in the description of ADMINPASSWD.
WL8_JDK_HOME	WebLogic JDK home directory. Default: WL8_HOME`/jdk141_03`
WL8_CONFIG_LOCATION	Should be set to the parent directory of the location of the WebLogic start script.
WL8_IS_SECURE	Specifies whether a secure port is enabled: • true: Secure port is enabled (HTTPS protocol). • false: Secure port is not enabled (HTTP protocol). Default: false (not enabled)

IBM WebSphere 5.1

Table 1-9 describes the configuration variables for IBM WebSphere Server 5.1 in the silent mode input file.

Table 1-9 IBM WebSphere 5.1 Configuration Variables

Variable	Description
WAS51_HOME	WebSphere home directory. Default: `/opt/WebSphere/AppServer`
WAS51_JDK_HOME	WebSphere JDK home directory. Default: `/opt/WebSphere/AppServer/java`
WAS51_CELL	WebSphere cell. Default: *`hostname`* value
WAS51_DOMAIN	WebSphere domain name. Default: `mydomain`
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: `sample`
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
WAS51_IS_SECURE	Specifies whether a secure port is enabled: • true: Secure port is enabled (HTTPS protocol). • false: Secure port is not enabled (HTTP protocol). Default: false (not enabled)

Directory Server Configuration Variables

Access Manager 2005Q1 supports Sun ONE Directory Server 5.1 and Sun Java System Directory Server 5 2005Q1. Table 1-10 describes the Directory Server configuration variables in the silent mode input file.

Table 1-10 Directory Server Configuration Variables

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

Access Manager amconfig Script

After your run the Java Enterprise System installer, the `amconfig` script is available in the *AccessManager-base*`/SUNWam/bin` directory on Solaris systems or the *AccessManager-base*`/identity/bin` directory on Linux systems.

The `amconfig` script reads a silent install input file and then calls other scripts in silent mode, as needed to perform the requested operation.

To run the `amconfig` script, use this syntax:

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

where:

-s runs `amconfig` in silent mode.

input-file is a silent install input file that contains the configuration variables for the operation you want to perform. For more information, see Access Manager Sample Configuration Script Input File.

NOTE In the Access Manager 2005Q1 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 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 mode input file and then running the `amconfig` script.

This section describes the following scenarios:

- Deploying Additional Instances of Access Manager
- Configuring and Reconfiguring an Instance of Access Manager
- Uninstalling an Access Manager Instance
- Uninstalling All Access Manager Instances

Deploying Additional Instances of Access Manager

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

- For Web Server 6.1 SP2, see:
  `http://docs.sun.com/coll/S1_websvr61_en`
- For Application Server 7.0 Update 3, see:
  `http://docs.sun.com/coll/s1_asseu3_en`

The steps described in this section only apply to an Access Manager instance that has been installed with the Configure Now option. If you are planning to use WebLogic or WebSphere as web containers, you must use the Configure Later option when installing Access Manager. See “Installing and Configuring Third-Party Web Containers” on page 55 for more information.

To Deploy an Additional Access Manager Instance

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

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

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

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

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

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

   - Access Manager Configuration Variables
   - Web Container Configuration Variables
   - Directory Server Configuration Variables

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

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

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

   ```
   cd AccessManager-base/SUNWam/bin/
   ./amconfig -s /newinstances/amnewws6instance
   ```

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

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

Configuring and Reconfiguring an Instance of Access Manager

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

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

To Configure or Reconfigure an Instance of Access Manager

1. Log in as an administrator, depending on the web container for the instance. For example, if Web Server 6.1 is the web container, log in either as superuser (root) or as the user account for Web Server Administration Server.
2. Copy the silent install input file you used to deploy the instance to a writable directory and make that directory your current directory. For example, to reconfigure an instance for Web Server 6.1, the following steps use an input file named `amnewinstanceforWS61` in the `/reconfig` directory.
3. In the `amnewinstanceforWS61` file, set the DEPLOY_LEVEL variable to one of the values described for a Re-install operation. For example, set DEPLOY_LEVEL=21 to reconfigure a full installation.
4. In the `amnewinstanceforWS61` file, set the NEW_INSTANCE variable to false:

   ```
   NEW_INSTANCE=false
   ```

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

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

   - Access Manager Configuration Variables
   - Web Container Configuration Variables
   - Directory Server Configuration Variables

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

   ```
   cd AccessManager-base/SUNWam/bin/
   ./amconfig -s /reconfig/amnewinstanceforWS61
   ```

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

Uninstalling an Access Manager Instance

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

To Uninstall an Instance of Access Manager

1. Log in as an administrator, depending on the web container for the instance. For example, if Web Server 6.1 is the web container, log in either as superuser (root) or as the user account for Web Server Administration Server.
2. Copy the silent install input file you used to deploy the instance to a writable directory and make that directory your current directory. For example, to unconfigure an instance for Web Server 6.1, the following steps use an input file named amnewinstanceforWS61 in the /unconfigure directory.
3. In the amnewinstanceforWS61 file, set the DEPLOY_LEVEL variable to one of the values described for an Uninstall (unconfigure) operation. For example, set DEPLOY_LEVEL=11 to uninstall (or unconfigure) a full installation.
4. Run the amconfig script, specifying your edited input file. For example, on Solaris systems:

   ```
   cd AccessManager-base/SUNWam/bin/
   ./amconfig -s /unconfigure/aminstanceforWS61
   ```

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

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

Uninstalling All Access Manager Instances

This scenario completely removes all Access Manager 2005Q1 instances and packages from a system.

To Completely Remove Access Manager 2005Q1 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 Uninstall (unconfigure) 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 Step 2. For example on Solaris systems:

   ```
   cd AccessManager-base/SUNWam/bin/
   ./amconfig -s /newinstances/amnewws6instance
   ```

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

   Repeat these steps for any other Access Manager instances you want to uninstall, except for the first instance, which is the instance you installed using the Java Enterprise System installer.
4. To uninstall the first instance and remove all Access Manager packages from the system, run the Java Enterprise System uninstaller. For information about the uninstaller, refer to the *Sun Java Enterprise System Installation Guide*.

Example Configuration Script Input File

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

Table 1-11 Sample Configuration Script Input File for WebLogic 8.1.x

```
DEPLOY_LEVEL=1
BASEDIR=/opt
SERVER_HOST=ide-56.example.company.com
SERVER_PORT=7001
SERVER_PROTOCOL=http
CONSOLE_HOST=$SERVER_HOST
CONSOLE_PORT=$SERVER_PORT
CONSOLE_PROTOCOL=$SERVER_PROTOCOL
CONSOLE_REMOTE=false
DS_HOST=ide-56.example.company.com
DS_PORT=389
DS_DIRMGRDN="cn=Directory Manager"
```

```
DEPLOY_LEVEL=1
DS_DIRMGRPASSWD=11111111
ROOT_SUFFIX=”dc=company,dc=com”
ADMINPASSWD=11111111
AMLDAPUSERPASSWD=00000000
CONSOLE_DEPLOY_URI=/amconsole
SERVER_DEPLOY_URI=/amserver
PASSWORD_DEPLOY_URI=/ampassword
COMMON_DEPLOY_URI=/amcommon
COOKIE_DOMAIN=.iplanet.com
JAVA_HOME=/usr/jdk/entsys-j2se
AM_ENC_PWD=””
PLATFORM_LOCALE=en_US
NEW_OWNER=root
NEW_GROUP=other
XML_ENCODING=ISO-8859-1
NEW_INSTANCE=false
WEB_CONTAINER=WL8
WL8_HOME=/export/bea8
WL8_PROJECT_DIR=user_projects
WL8_DOMAIN=mydomain
WL8_CONFIG_LOCATION=$WL8_HOME/$WL8_PROJECT_DIR/domains
WL8_SERVER=myserver
WL8_INSTANCE=/export/bea8/weblogic81
WL8_PROTOCOL=http
WL8_HOST=ide-56.example.company.com
WL8_PORT=7001
WL8_SSLPORT=7002
WL8_ADMIN=”weblogic”
WL8_PASSWORD=”11111111”
WL8_JDK_HOME=$WL8_HOME/jdk142_04
WL8_IS_SECURE=false
DIRECTORY_MODE=1
USER_NAMING_ATTR=uid
ORG_NAMING_ATTR=o
ORG_OBJECT_CLASS=examplemanagedorganization
USER_OBJECT_CLASS=inetorgperson
DEFAULT_ORGANIZATION=
```

Chapter 2

Installing and Configuring Third-Party Web Containers

This chapter describes the procedures for installing and configuring third-party web containers deployed with Sun Java™ System Access Manager. For this release, Access Manager supports BEA WebLogic 8.1 (and its current patches) and IBM WebSphere 5.1 (and its current patches).

WebLogic and WebSphere are not part of the Java Enterprise System, so you must install and configure them independently of the Java ES Install program. In general the procedures are:

1. Install, configure, and start the web container instance.
2. Install the Directory Server from the Java ES installer.
3. Install Access Manger from the Java ES Installer in Configure Later Mode, which will leave Access Manager in an unconfigured state.
4. Run the Access Manager configuration scripts to deploy Access Manager in the web container.
5. Restart the web container.

Installing and Configuring BEA WebLogic 8.1

Before you install WebLogic, make sure that your host domain is registered in DNS and verify that you are installing the correct version of the WebLogic software. For more information, go to the BEA product site at http://commerce.bea.com/index.jsp.

To Install and Configure WebLogic 8.1

1. Unpack the downloaded software image, either in `.zip` or `.gz` format. Make sure that the zip / gzip utility you are using is for the correct platform, or you may receive a checksum error during the unpackaging.

2. Run the installation program from a shell window of your target system.

   Follow the procedures provided by the WebLogic installation utility (detailed installation instructions can be found at `http://e-docs.bea.com/wls/docs81/`).

   During the installation process, make sure that you record the following information, as it will be used later in the Access Manager configuration:

   - FQDN (used in the `WL8_HOST` parameter)
   - installation location
   - port number

3. Once installation is complete, run the WebLogic configuration tool to configure the domain and server instance from the following location:

   WebLogic-base`/`*WebLogic-instance*`/common/bin/quickstart.sh`

   By default, WebLogic defines the server instance as `myserver` and the domain as `mydomain`, however it is unlikely that you will choose to use these defaults. If you create a new domain and instance, make sure that you record the information for Access Manager configuration and deployment. See the WebLogic 8.1 documentation for instructions.

4. If you are installing on an administration instance, start WebLogic by using the `startWebLogic.sh` utility from the following location:

   WebLogic-base`/`*WebLogic-Userhome*`/domains/`*WebLogic-domain*`/startWebLogic.sh`

   If you are installing on a managed instance, start WebLogic by using the following command:

   WebLogic-base`/`*WebLogic-Userhome*`/domains/`*WebLogic-domain*`/startManagedWebLogic` *WebLogic-managed-instancename admin-url*

Installing and Configuring IBM WebSphere 5.1

Before you install WebSphere, make sure that your host domain is registered in DNS and verify that you are installing the correct version of the WebSphere software for your platform. For more information, go to the IBM product support website at http://www-306.ibm.com/software/websphere/support/.

To Install and Configure WebSphere 5.1

1. Unpack the downloaded software image, either in `.zip` or `.gz` format. Make sure that the zip / gzip utility you are using is for the correct platform, or you may receive a checksum error during the unpackaging.

2. Run the installation program from a shell window of your target system. If you are planning on installing a patch, install the 5.1 version first and apply the patch later (see Step 5). Detailed installation instructions can be found at http://publib.boulder.ibm.com/infocenter/ws51help/index.jsp.

   During the installation process, make sure that you record the following information, as it will be used later in the Access Manager configuration:

   - hostname
   - domain name
   - cell name
   - node name
   - port number
   - installation directory
   - WebSphere instance name
   - administration port

   By default, WebSphere defines the server instance as `server1`, however it is unlikely that you will choose to use the default. If you create a new instance, make sure that you record the information for Access Manager configuration and deployment. See the WebSphere 5.1 documentation for instructions.

3. Verify that the installation was successful. To do so:

   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. Use the `startServer.sh` command to start the server. For example:

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

c. In a web browser, enter the corresponding URL of the following format to view the sample web application:

`http://`*`fqdn:portnumber/snoop`*

4. Once you have verified a successful installation, stop the server using the `stopServer.sh` utility. For example:

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

5. If you are installing WebSphere 5.1 patch, use the `updateWizard.sh` command line utility to install the patch over the original 5.1 instance.

6. Restart WebSphere and verify that the installation was successful (see Step 3).

Using Java ES to Install Directory Server and Access Manager

Access Manager installation involves two separate invocations of the Java Enterprise System (Java ES) Installer.

1. Run the first Java ES invocation to install Directory Server (either local or remote) with the Configure Now option. The Configure Now option allows you to configure the first instance during the installation by the choices (or default values) that you select.

2. Run the second Java ES invocation to install Access Manager with the Configure Later option. This option Installs the Access Manager 2005Q1 components, and then after installation, you must configure them.

   WebLogic and WebSphere are installed independently of Java ES, so the Installer does not contain the necessary configuration data to automatically deploy the containers. Because of this, you must select the Configure Later option when installing Access Manager. This option leaves your Access Manager deployment in the following state:

   - The active Directory Server (either Local or Remote) does not have Access Manager DIT data loaded.
   - Access Manager configuration files are not automatically loaded.
   - Access Manager web application `.war` files are not generated.

- Access Manager deployment and post-installation configuration processes are not automatically started and run.

For detailed installation instructions, refer to the Sun Java Enterprise System Installation Guide located at http://docs.sun.com/doc/819-0056.

Configuring Access Manager

After you have completed Access Manager installation on the target system's local drive, you need to manually configure Access Manager with either WebLogic 8.1 or WebSphere 5.1. This is a three-step process:

1. Edit the configuration script input file
2. Run the configuration script
3. Restart the web container

Creating the Configuration Script Input File

The Access Manager configuration script input file contains all of the deployment level, Access Manager, web container, and Directory Server variable definitions. Access Manager contains a sample configuration script input file template (`amsamplesilent`) which is available in the *AccessManager-base*`/SUNWam/bin` directory on Solaris systems or the *AccessManager-base*`/identity/bin` directory on Linux systems.

You can use the `amsamplesilent` template to construct your configuration script input file. Instructions for editing the file, as well as the variable definitions, are described in "Access Manager Sample Configuration Script Input File" on page 35.

Before you edit the file, make sure that you have the following information available from your web container installation:

BEA WebLogic and IBM WebSphere

- installation location
- instance name and location
- hostname
- FQDN
- port number to which it is listening

- administration ID
- protocol used

BEA WebLogic only

- administration password
- shared library location
- domain name and location
- project directory name
- JDK location

IBM WebSphere only

- cell name
- node name
- JDK location

Running the Configuration Script

When you have saved the configuration script input file, you run the `AMConfig` script to complete the configuration process. For example:

```
AccessManager-base/SUMWam/bin/amconfig -s silentfile
```

`silentfile` should be the absolute path to the configuration input file.

Running this script performs the following functions:

1. Loads the Access Manager schema to the active Directory Server instance.
2. Loads the Access Manager service data to the Directory Server instance.
3. Generates the Access Manager configuration files used by the active Access Manager instance.
4. Deploys the Access Manager web application data to the web container.
5. Customizes the web container configuration to match the Access Manager requirements.

The procedures for running the `AMConfig` script is described in "Configuring and Reconfiguring an Instance of Access Manager" on page 50.

Restarting the Web Container

After you have completed the configuration process, you must restart the web container. Refer to your product's documentation for instructions.

For BEA WebLogic 8.1, see http://e-docs.bea.com/wls/docs81.

For IBM WebSphere 5.1, see http://publib.boulder.ibm.com/infocenter/ws51help/index.jsp.

Chapter 3

Configuring Access Manager in SSL Mode

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

1. Configure Access Manager with a secure web container
2. Configure Access Manager to a secure Directory Server

The following sections describe these steps:

Configuring Access Manager With a Secure Sun Java System Web Server

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

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

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

Step 2 through Step 25 describe the Sun Java System Web Server.

2. Log on to the Web Server console. The default port is 58888.
3. Select the Web Server instance on which Access Manager is running, and click Manage.

   This displays a pop-up window explaining that the configuration has changed. Click OK.
4. Click on the Apply button located top right corner of the screen.
5. Click Apply Settings.

   The Web Server should automatically restart. Click OK to continue.
6. Stop the select Web Server instance.
7. Click the Security Tab.
8. Click on Create Database.
9. Enter the new database password and click OK.

   Ensure that you write down the database password for later use.
10. Once the Certificate Database has been created, click on Request a Certificate.
11. Enter the data in the fields provided in the screen.

    The Key Pair Field Password field is the same as you entered in Step 9. In the location field, you will need to spell out the location completely. Abbreviations, such as CA, will not work. All of the fields must be defined. In the Common Name field, provide the hostname of your Web Server.
12. Once the form is submitted, you will see a message such as:

```
--BEGIN CERTIFICATE REQUEST---

afajsdllwqeroisdaoi234rlkqwelkasjlasnvdknbslajowijalsdkjfalsdflasdf

alsfjawoeirjoi2ejowdnlkswnvnwofijwoeijfwiepwerfoiqeroijeprwpfrwl

--END CERTIFICATE REQUEST--
```

13. Copy this text and submit it for the certificate request.

    Ensure that you get the Root CA certificate.

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

```
--BEGIN CERTIFICATE---

afajsdllwqeroisdaoi234rlkqwelkasjlasnvdknbslajowijalsdkjfalsdflasdf

alsfjawoeirjoi2ejowdnlkswnvnwofijwoeijfwiepwerfoiqeroijeprwpfrwl

--END CERTIFICATE---
```

15. Copy this text into your clipboard, or save the text into a file.
16. Go to the Web Server console and click on Install Certificate.
17. Click on Certificate for this Server.
18. Enter the Certificate Database password in the Key Pair File Password field.
19. Paste the certificate into the provided text field, or check the radio button and enter the filename in the text box. Click Submit.

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

20. Click Install Certificate.
21. Click Certificate for Trusted Certificate Authority.

22. Install the Root CA Certificate in the same manner described in Step 16 through Step 21.
23. Once you have completed installing both certificates, click on the Preferences tab in the Web Server console.
24. Select Add Listen Socket if you wish to have SSL enabled on a different port. Then, select Edit Listen Socket.
25. Change the security status from Disabled to Enabled, and click OK to submit the changes.

Step 26 through Step 28 describe Access Manager.

26. Open the `AMConfig.properties` file. By default, the location of this file is `etc/opt/SUNWam/config`.
27. Replace all of the protocol occurrences of `http://` to `https://`, except for the Web Server Instance Directory. This is also specified in `AMConfig.properties`, but must remain the same.
28. Save the `AMConfig.properties` file.
29. In the Web Server console, click the ON / OFF button for the Access Manager hosting web server instance.

    The Web Server displays a text box in the Start / Stop page.
30. Enter the Certificate Database password in the text field and select Start.

Configuring Access Manager with a Secure Sun Java System Application Server

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

Setting Up Application Server 6.2 With SSL

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 Sun Java System 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 Sun Java System 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. (It is the same password you entered in Step 9).
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 in Step 10 through Step Step 22. However, in Step 18, 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 Sun Java System 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.

Setting Up Application Server 8.1 With SSL

To Secure the Application Server Instance:

1. Make sure that the Application Server instance is stopped.
2. Change the token password by using the `asadmin>change-master-password` command.
3. Go to the Application Server console and select Configuration>HTTP Service>HTTP Listeners.
4. Click the listener that you wish to enable and in the right pane, select Security:Enabled.
5. Check if `certutil` is installed.
   a. Go to `/usr/sfw/bin`.
   b. If it is not, install the `SUNWtlsu` package from the following directory:
   c. The shell environment variable, LD_LIBRARY_PATH

      `LD_LIBRARY_PATH has to have /usr/lib/mps/secv1`
6. Use `certutil` to check for installed certificates in `certdb`:
   a. Go to `/var/opt/SUNWappserver/domains/domain1/config`
   b. `certutil -L -d`

c. You will see following output:

```
Certificate Name                                                Trust
Attributes

Server-Cert                                                      u,u,u
s1as                                                               u,u,u
AM TrustCA                                                     CT,CT,CT
verisignc1g1                                                     T,T,T
verisignc1g2                                                     T,T,T
verisignc1g3                                                     T,T,T
verisignc2g1                                                     T,T,T
verisignc2g2                                                     T,T,T
verisignc2g3                                                     T,T,T
verisignc3g1                                                     T,T,T
verisignc3g2                                                     T,T,T
verisignc3g3                                                     T,T,T
verisignsecureserver                                             T,T,T

p    Valid peer
P    Trusted peer (implies p)
c    Valid CA
T    Trusted CA to issue client certs (implies c)
C    Trusted CA to certs(only server certs for ssl) (implies c)
u    User cert
w    Send warning
```

Application Server 8.1 installs self signed server certificate (nickname, s1as) in install time and uses it for ssl enabled ports 4848, 8181.

7. Generate the certificate request.The syntax to do so is:

```
certutil -R -s subj -o cert-request-file [-d certdir] [-P dbprefix]
[-p phone] [-a]
```

For example:

```
certutil -R -s "CN=test.company1.com, O=company1.com, C=US" -o
cert.req -d . -a
```

8. Retrieve the certificate from the CA and istall it by using the following command:

```
certutil -A -n cert-name -t trustargs [-d certdir] [-P dbprefix]
[-a] [-i input]
```

9. Save server certificate in a file, for example `cert.txt`.

10. Install the server certificate with the following command:

```
certutil -A -n “AMTrustCA” -t “CT<CT,CT” -d. -a -i cert.txt
```

11. List the `certdb` to ensure successful installation. Enter the following command:

    ```
    /var/opt/SUNWappserver/domains/domain1/config/% certutil -L -d
    ```

12. Go to the Application Server Admin Console and choose HTTP Listeners.

    Under General Settings, configure the HTTP Listener with the new server certificate.

13. Restart the Application Server.

Configuring Access Manager in SSL Mode Using JSS

To configure Access Manager in SSL mode using JSS (Network Security Service for Java):

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

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

---

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

   `/etc/opt/SUNWam/config`.

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

4. Save the `AMConfig.properties` file.

5. Restart the Application Server.

Configuring AMSDK with a Secure BEA WebLogic Server

The BEA WebLogic Server must first be installed and configured as a web container before you configure it with the AMSDK in SSL. For installation instructions, see the BEA WebLogic server documentation. To configure WebLogic as a web container for Access Manager, see Chapter 1, “Access Manager 2005Q1 Configuration Scripts” on page 31.

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 `vetri_csr.txt` CSR 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 "Greenday CA" -storepass changeit -file
   /opt/bea81/cacert.txt
   ```

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

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

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

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

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

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

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

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

    ```
    com.sun.identity.jss.donotInstallAtHighestPriority=true [ this is not required for AM 6.3 and above]
    ```

    ```
    com.iplanet.security.SecureRandomFactoryImpl=com.iplanet.am.util.SecureRandomFactoryImpl
    ```

    ```
    com.iplanet.security.SSLSocketFactoryImpl=netscape.ldap.factory.JSSESocketFactory
    ```

    ```
    com.iplanet.security.encryptor=com.iplanet.services.util.JCEEncryption2
    ```

    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 Using JSS” on page 72.

Configuring AMSDK with a Secure IBM WebSphere Application Server

The IBM WebShpere 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 1, “Access Manager 2005Q1 Configuration Scripts” on page 31.

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 IBMJSSE 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 Using JSS" on page 72.

Configuring Access Manager to Directory Server in SSL Mode

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

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

Configuring Directory Server in SSL Mode

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

`http://docs.sun.com/doc/817-5221`

You can also download a PDF of the manual from the following location:

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

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

Connecting Access Manager to the SSL-enabled Directory Server

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

1. In the Access Manager Console, go to the LDAP Authentication service in the Service Configuration module.
   a. Change the Directory Server port to the SSL port.
   b. Select the Enable SSL Access to LDAP Server attribute.
2. Go to the Membership Authentication service in the Service Configuration module.
   a. Change the Directory Server port to the SSL port.
   b. Select the Enable SSL Access to LDAP Server attribute.
3. Go to the Policy Configuration service located in Service Configuration.
   a. Change the Directory Server port to the SSL port.
   b. Select the Enable LDAP SSL attribute.
4. Open the `serverconfig.xml` in a text editor. The file is in the following location:

   `/etc/opt/SUNWam/config`

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

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

      `type`- change SIMPLE to SSL.

   b. Save and close `serverconfig.xml`.
5. Open the `AMConfig.properties` file from the following default location:

   /etc/opt/`SUNWam/config`.

   Change the following properties:

   a. `Directory Port = 636` (if using the default)
   b. `ssl.enabed = true`
   c. Save `AMConfig.properties`.

6. Restart the server

Part II

Managing Access Manager Through the Console

This is part two of the *Sun Java™ System Access Manager 6 2005Q1 Administration Guide*. It discusses the Access Manager graphical user interface and how to navigate through it. This section contains the following chapters:

Chapter 4

Identity Management

This chapter describes the identity management features of Sun Java™ System Access Manager 6 2005Q1. The Identity Management module interface provides a way to view, manage and configure all Access Manager objects and identities. This chapter contains the following sections:

The Access Manager Console

The Access Manager console is divided into three sections: the Location pane, the Navigation pane and the Data pane. By using all three panes, the administrator is able to navigate the directory, perform user and service configurations and create policies.

Header Pane

The Header pane runs along the top of the console. The tabs in the Header pane allow the administrator to switch between the different management module views:

- Identity Management module - allows for the creation and management of identity-related objects.
- Service Configuration module - allows for the configuration of Access Manager's default services.

- Current Sessions module - allows administrators to view current session information, as well as terminating any session.
- Federation Management module - allows for the utilization of the open standards for federated network identity being developed by the Liberty Alliance Project.

The *Location* field provides a trail to the administrator's position in the directory tree. This path is used for navigational purposes.

The *Welcome* field displays the name of the user that is currently running the console with a link to the user profile.

The *Search* link displays an interface that allows the user to search for entries of a specific Access Manager object type. Use the pull-down menu to select the object type and enter the search string.The Results are returned in the search table. Wildcards are accepted.

The *Help* link opens a browser window containing information on Identity Management, Current Sessions, Federation Management and Part IV of this documentation, the Attribute Reference.

The *Logout* link allows the user to log out of the Access Manager.

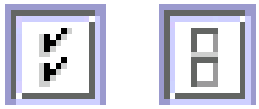
Navigation Pane

The Navigation pane is the left portion of the Access Manager console. The *Directory Object* portion (within the grey box) displays the name of the directory object that is currently open and its *Properties* link. (Most objects displayed in the Navigation pane will have a corresponding *Properties* link. Selecting this link will render the entry's attributes in the Data pane to the right.) The View menu lists the directories under the selected directory object. Depending on the number of sub-directories, a paging mechanism is provided.

Data Pane

The Data peneplain is the right portion of the console. This is where all object attributes and their values are displayed and configured and where entries are selected for their respective group, role or organization.

TIP You can select or deselect all of the items in a list by clicking the Select All, or Deselect All icons.



There are two basic views of the Access Manager graphical user interface. Depending on the roles of the user logging in, they might gain access to the Identity Management view or the User Profile view.

Identity Management View

When a user with an administrative role authenticates to the Access Manager, the default view is the Identity Management view. In this view the administrator can perform administrative tasks. Depending on the role of the administrator, this can include creating, deleting and managing objects (users, organizations, policies, and so forth), and configuring services.

User Profile View

When a user who has not been assigned an administrative role authenticates to the Access Manager, the default view is the user's own User Profile. In this view the user can modify the values of the attributes particular to the user's personal profile. This can include, but is not limited to, name, home address and password. The attributes displayed in the User Profile View can be extended. For more information on adding customized attributes for objects and identities, see the *Access Manager Developer's Guide*.

Properties Function

To view or modify an entry's properties, click the Properties arrow next to the object's name. Its attributes and corresponding values are displayed in the Data pane. Different objects display different properties.

See the *Access Manager Developer's Guide* for information on how to extend an entry's properties.

The Identity Management Interface

The Identity Management interface allows for the creation and management of identity-related objects. User, role, group, policies, organization, suborganization and container objects and more can be defined, modified or deleted using either the Access Manager console or the command line interface. The console has default administrators with varying degrees of privileges used to create and manage the organizations, groups, containers, users, services, and policies. (Additional administrators can be created based on roles.) The administrators are defined within the Directory Server when installed with Access Manager.

Managing Access Manager Objects

The User Management interface contains all the components needed to view and manage the Access Manager objects (organizations, groups, users, services, roles policies, container objects, and agents). This section explains the object types and details how to configure them.

For most Access Manager object types, you can optionally configure Display Options and Available Actions to show or hide the way in which the web interfaces are displayed in the Access Manager console. Configuration is done at the organization and role levels and users inherit the configuration from the organization in which they reside and the roles that are assigned to them. These settings are described at the end of this chapter.

Organizations

An *Organization* represents the top-level of a hierarchical structure used by an enterprise to manage its departments and resources. Upon installation, Access Manager dynamically creates a top-level organization (defined during installation) to manage the Access Manager enterprise configurations. Additional organizations can be created after installation to manage separate enterprises. All created organizations fall beneath the top-level organization.

To Create an Organization

1. Choose Organizations from the View menu in the Identity Management module.
2. Click New in the Navigation pane.
3. Enter the values for the fields. Only Name is required. The fields are:

   Name. Enter a value for the name of the Organization.

   Domain Name. Enter the full Domain Name System (DNS) name for the organization, if it has one.

   Organization Status. Choose a status of `active` or `inactive`.

   The default is `active`. This can be changed at any time during the life of the organization by selecting the Properties icon. Choosing `inactive` disables user access when logging in to the organization.

   Organization Aliases. This field defines alias names for the organization, allowing you to use the aliases for authentication with a URL login. For example, if you have an organization named `exampleorg`, and define `123` and `abc` as aliases, you can log into the organization using any of the following URLs:

   `http://machine.example.com/amserver/UI/Login?org=exampleorg`

   `http://machine.example.com/amserver/UI/Login?org=abc`

   `http://machine.example.com/amserver/UI/Login?org=123`

   Organization alias names must be unique throughout the organization. You can use the Unique Attribute List to enforce uniqueness.

   DNS Alias Names. Allows you to add alias names for the DNS name for the organization. This attribute only accepts “real” domain aliases (random strings are not allowed). For example, if you have a DNS named `example.com`, and define `example1.com` and `example2.com` as aliases for an organization named `exampleorg`, you can log into the organization using any of the following URLs:

   `http://machine.example.com/amserver/UI/Login?org=exampleorg`

   `http://machine.example1.com/amserver/UI/Login?=org=exampleorg`

   `http://machine.example2.com/amserver/UI/Login?org=exampleorg`

   Unique Attribute List. Allows you to add a list of unique attribute names for users in the organization. For example, if you add a unique attribute name

specifying an email address, you would not be able to create two users with the same email address. This field also accepts a comma-separated list. Any one of the attribute names in the list defines uniqueness. For example, if the field contains the following list of attribute names:

`PreferredDomain, AssociatedDomain`

and `PreferredDomain` is defined as `http://www.example.com` for a particular user, then the entire comma-separated list is defined as unique for that URL.

Uniqueness is enforced for all suborganizations.

4. Click OK.

   The new organization displays in the Navigation pane. To edit any of the properties that you defined during creation of the organization, click the Properties arrow of the organization you wish to edit, select General from the View menu in the Data pane, edit the properties and click OK. You can use the Display Options and Available Actions views to customize the appearance of the Access Manager console and to specify the behavior for any users that authenticate to this organization.

To Delete an Organization

1. Choose Organizations from the View menu in Identity Management.

   All created organizations are displayed. To display specific organizations, enter a search string and click Search.

2. Select the checkbox next to the name of the Organization to be deleted.

3. Click Delete.

NOTE There is no warning message when performing a delete. All entries within the organization will be deleted and you can not perform an undo.

To Add an Organization to a Policy

Access Manager objects are added to a policy through the policy's subject definition. When a policy is created or modified, organizations, roles, groups, and users can be defined as the subject in the policy's Subject page. Once the subject is defined, the policy will be applied to the object. For more information, see "Managing Policies" on page 132.

Groups

A *group* represents a collection of users with a common function, feature or interest. Typically, this grouping has no privileges associated with it. Groups can exist at two levels; within an organization and within other managed groups. Groups that exist within other groups are called *sub-groups*. Sub-groups are child nodes that "physically" exist within a parent group.

Access Manager also supports *nested groups*, which are "representations" of existing groups contained in a single group. As opposed to sub-groups, nested groups can exist anywhere in the DIT. They allow you to quickly set up access permissions for a large number of users.

When you create a group, you can create groups that use Membership By Subscription (*static group*) or Membership By Filter (*filtered groups*). This controls the way in which users are added to the group. Users can only be added to static groups. Dynamic groups control the addition of users through a filter. Nested or sub-groups, however, can be added to both.

Static Group (Membership By Subscription)

When you specify group membership by subscription, a static group is created based on the Managed Group Type you specify. If the Managed Group Type value is `static`, group members are added to a group entry using the `groupOfNames` or `groupOfUniqueNames` object class. If the Managed Group Type value is `dynamic`, a specific LDAP filter is used to search and return only user entries that contain the `memberof` attribute. For more information, see "Managed Group Type" on page 263.

NOTE By default, the managed group type is dynamic. You can change this default in the Administration service configuration.

Filtered Group (Membership By Filter)

A filtered group is a dynamic group that is created through the use of an LDAP filter. All entries are funneled through the filter and dynamically assigned to the group. The filter would look for any attribute in an entry and return those that contain the attribute. For example, if you were to create a group based on a building number, you can use the filter to return a list all users containing the building number attribute.

NOTE Access Manager should be configured with Directory Server to use the referential integrity plug-in. When the referential integrity plug-in is enabled, it performs integrity updates on specified attributes immediately after a delete or rename operation. This ensures that relationships between related entries are maintained throughout the database. Database indexes enhance the search performance in Directory Server. For more information on enabling the plug-in, see the *Sun Java System Access Manager Migration Guide*.

To Create a Static Group

1. Navigate to the organization, group or group container where the group will be created.
2. Choose Groups from the View menu.
3. Click New.
4. Select Membership By Subscription for the group type from within the Data pane.
5. Enter a name for the group in the Name field. Click Next.
6. Select the Users Can Subscribe to this Group attribute to allow users to subscribe to the group themselves.
7. If you have defined multiple group containers in your DIT and the Show Group Containers attribute (from the Administration Service) is not enabled, you can select the Parent Group Container to which the static group will belong. Otherwise, this field is not displayed.
8. Click Finish.

   Once the group is created, you can edit the Users Can Subscribe to this Group attribute by selecting General from the View menu in the Data pane.

To Add or Remove Members to a Static Group

1. Click the Properties arrow next to the group to which you will add members.

2. In the Data pane, select Members from the View menu.

   Choose an action to perform in the Select Action menu. The actions you can perform are as follows:

   New User. This action creates a new user and automatically adds the user to the group when the user information is saved.

   Add User. This action adds an existing user to the group. When you select this action, you create a search criteria which will specify users you wish to add. The fields used to construct the criteria use either an ANY or ALL operator. ALL returns users for all specified fields. ANY returns users for any one of the specified fields. If a field is left blank, it will match all possible entries for that particular attribute.

   Once you have constructed the search criteria, click Next. From the returned list of users, select the users you wish to add and click Finish.

TIP Click the Show Path button to view the complete organizational path of the user.

   Add Group. This action adds a nested group to the current group. When you select this action, you create a search criteria, including search scope, the name of the group (the "*" wildcard is accepted), and you can specify whether users can subscribe to the group themselves. Once you have entered the information, click Next. From the returned list of groups, select the group you wish to add and click Finish.

   Remove Members. This action will remove members (which includes users and groups) from the group, but will not delete them. Select the member(s) you wish to remove and choose Remove Members from the Available Actions list.

   Delete Members. This action will permanently delete the member you select. Select the member(s) you wish to delete and choose Delete Members from the Available Actions list.

To Create a Filtered Group

1. Navigate to the organization (or group) where the group will be created.
2. Choose Groups from the View menu.
3. Click New.
4. Select Membership By Filter for the group type from within the Data pane.

5. Enter a name for the group in the Name field. Click Next.
6. Construct the LDAP search filter.

   By default, Access Manager displays the Basic search filter interface. The Basic fields used to construct the filter use either an ANY or ALL operator. ALL returns users for all specified fields. ANY returns users for any one of the specified fields. If a field is left blank it will match all possible entries for that particular attribute.

   Alternatively, you can select the Advanced button to define the filter attributes yourself. For example,

   ```
   (&(uid=user1)(|(inetuserstatus=active)(!(inetuserstatus=*))))
   ```

   When you click Finish, all users matching the search criteria are automatically added to the group.

To Add or Remove Members to a Filtered Group

1. Click the Properties arrow next to the group to which you will add members.
2. In the Data pane, select Members from the View menu.

   Choose an action to perform in the Action menu. The actions you can perform are as follows:

   Add Group. This action adds a nested group to the current group. When you select this action, you create a search criteria, including search scope, the name of the group (the "*" wildcard is accepted), and you can specify whether users can subscribe to the group themselves. Once you have entered the information, click Next. From the returned list of groups, select the group you wish to add and click Finish.

   Remove Members. This action will remove members (which includes groups) from the group, but will not delete them. Select the member(s) you wish to remove and choose Remove Members from the Available Actions list.

   Delete Members. This action will permanently delete the member you select. Select the member(s) you wish to delete and choose Delete Members from the Available Actions list.

To Add a Group to a Policy

Access Manager objects are added to a policy through the policy's subject definition. When a policy is created or modified, organizations, roles, groups, and users can be defined as the subject in the policy's Subject page. Once the subject is defined, the policy will be applied to the object. For more information, see "Managing Policies" on page 132.

Users

A *user* represents an individual's identity. Through the Access Manager Identity Management module, users can be created and deleted in organizations, containers and groups and can be added or removed from roles and/or groups. You can also assign services to the user.

NOTE If a user in a suborganization is created with the same userid as `amadmin`, the login will fail for `amadmin`. If such a problem arises, the administrator should change the user's userid through the Directory Server console. This enables the administrator to login to the default organization. Additionally, the DN to Start User Search in the authentication service can be set to the people container DN to ensure that a unique match is returned during the login process.

To Create a User

1. Navigate to the organization, container or people container where the user is to be created.
2. Choose Users from the View menu.
3. Click New.

   This displays the New User page in the Data pane.
4. If there are services available to the users, select a service to which the user will subscribe from the Available Services page. If you wish to skip this page, click Next.

5. Enter data for the following default required values:

   UserId. This field takes the name of the user with which he or she will log into Access Manager. This property may be a non-DN value.

   First Name. This field takes the first name of the user. The First Name value and the Last Name value identify the user in the Currently Logged In field in the upper right corner of the Access Manager console. This is not a required value.

   Last Name. This field takes the last name of the user. The First Name value and the Last Name value identify the user in the Currently Logged In field in the upper right corner of the Access Manager console.

   Full Name. This field takes the full name of the user.

   Password. This field takes the password for the name specified in the User Id field.

   Password (Confirm). Confirm the password.

   User Status. This option indicates whether the user is allowed to authenticate through Access Manager. Only active users can authenticate through Access Manager. The default value is `Active.`

6. Click Finish.

To Add a User to Roles and Groups

1. Navigate to the Organization for the user that is to be modified.
2. Choose Users from the View menu.
3. In the Navigation pane, select the user you wish to modify and click the Properties arrow.
4. From the View menu in the Data pane, select Roles or Groups. Only the roles and groups that have already been assigned to the user are displayed. Click Add to see the list of available roles and groups from which to choose.
5. Select the role or group that to which you wish to add the user, and click Save.

To Add a Service to a User

1. Navigate to the Organization for the user that is to be modified.
2. Choose Users from the View menu in the Navigation pane.
3. In the Navigation pane, select the user you wish to modify and click the Properties arrow.

4. From the View menu in the Data pane, select Services. The list of services that are available to the user are displayed in the Add Services pages.
5. Select the services you wish to assign to the user.
6. Click OK.

   To edit a service's attributes, click the Edit link next to the service name. Only services that are editable will display the Edit link.

To Remove a User

1. From the View menu in the Data pane, select Roles or Groups.
2. From the Selected list, choose the role or group that from which you wish to remove the user, and click Remove. You can optionally remove the user from all available roles and groups by clicking Remove All.
3. Click Save to remove the user.

NOTE There is no warning message before the delete operation, and it can not be undone.

To Add a User to a Policy

Access Manager objects are added to a policy through the policy's subject definition. When a policy is created or modified, organizations, roles, groups, and users can be defined as the subject in the policy's Subject page. Once the subject is defined, the policy will be applied to the object. For more information, see "Managing Policies" on page 132.

Services

Activating a *service* for an organization or container (containers behave the same as organizations) is a two step process. In the first step you need to add the service to the organization. After you add the service, you must configure a template configured specifically for that organization.

NOTE New services must first be imported into the Access Manager through the command line's `amadmin`. Information on importing a service's XML schema can be found in the *Access Manager Developer's Guide*.

To Add a Service

1. Navigate to the Organization where you will add services.
2. Choose Services from the View menu.
3. Click Add.

   The Data pane will display a list of services available to add to this organization.
4. Select the checkbox next to each service to be added.
5. Click OK. The services that have been added are displayed in the Navigation pane.

NOTE Only the services that are added to the parent organization are displayed at the suborganization level.

To Create a Template for a Service

1. Navigate to the organization or role where the added service exists.

   Choose Organizations from the View menu in the Identity Management module and select the organization from the Navigation pane.
2. Choose Services from the View menu.
3. Click the properties icon next to the name of the service to be activated.

   The Data pane displays the message *A template does not currently exist for this service. Do you want to create one now?*
4. Click Yes.

   A template is created for this service for the parent organization or role. The Data pane displays the default attributes and values for this service. Descriptions for the attributes for the default services are described in the "Attribute Reference" on page 259.
5. Accept or modify the default values and click Save.

To Remove a Service

1. Navigate to the organization where you will remove services.

   Choose Organizations from the View menu in Identity Management module and select the organization from the Navigation pane.

2. Choose Services from the View menu.
3. Select the checkboxes for the services to remove.
4. Click Remove.

NOTE Services can not be removed from the parent organization level if they are registered at the sub organization level.

Roles

Roles are a Directory Server entry mechanism similar to the concept of a *group*. A group has members; a role has members. A role's members are LDAP entries that possess the role. The criteria of the role itself is defined as an LDAP entry with attributes, identified by the Distinguished Name (DN) attribute of the entry. Directory Server has a number of different types of roles but Access Manager can manage only one of them: the managed role.

NOTE The other Directory Server role types can still be used in a directory deployment; they just can not be managed by the Access Manager console. Other Directory Server types can be used in a policy's subject definition. For more information on policy subjects, see "Creating Policies" on page 129.

Users can possess one or more roles. For example, a contractor role which has attributes from the Session Service and the Password Reset Service might be created. When new contractors start, the administrator can assign them this role rather than setting separate attributes in the contractor entry. If the contractor is working in the Engineering department and requires services and access rights applicable to an engineering employee, the administrator could assign the contractor to the engineering role as well as the contractor role.

Access Manager uses roles to apply access control instructions. When first installed, Access Manager configures access control instructions (ACIs) that define administrator permissions. These ACIs are then designated in roles (such as Organization Admin Role and Organization Help Desk Admin Role) which, when assigned to a user, define the user's access permissions.

Users can view their assigned roles only if the Display User's Roles attribute is enabled in the Administration Service. For more information, see "Show Roles on User Profile Page" on page 271.

NOTE Access Manager should be configured with Directory Server to use the referential integrity plug-in. When the referential integrity plug-in is enabled, it performs integrity updates on specified attributes immediately after a delete or rename operation. This ensures that relationships between related entries are maintained throughout the database. Database indexes enhance the search performance in Directory Server. For more information on enabling the plug-in, see the *Sun Java System Access Manager Migration Guide*.

Similar to groups, roles can be created by a filter, or be created statically.

Static Role. In contrast to a filtered role, a static role can be created without adding users at the point of the role's creation. This gives you more control when adding specific users to a given role.

Filtered Role. A filtered role is a dynamic role created through the use of an LDAP filter. All users are funneled through the filter and assigned to the role at the time of the role's creation. The filter looks for any attribute value pair (for example, `ca=user*`) in an entry and automatically assign the users that contain the attribute to the role.

To Create a Static Role

1. In the Navigation pane go the organization where the role will be created.
2. Choose Roles from the View menu.

   A set of default roles are created when an organization is configured, and are displayed in the Navigation pane. The default roles are:

   Container Help Desk Admin. The Container Help Desk Admin role has read access to all entries in an organizational unit and write access to the `userPassword` attribute in user entries only in this container unit.

   Organization Help Desk Admin. The Organization Help Desk Administrator has read access to all entries in an organization and write access to the `userPassword` attribute.

NOTE When a suborganization is created, remember that the administration roles are created in the suborganization, not in the parent organization.

Container Admin. The Container Admin role has read and write access to all entries in an LDAP organizational unit. In Access Manager, the LDAP organizational unit is often referred to as a container.

Organization Policy Admin. The Organization Policy Administrator has read and write access to all policies, and can create, assign, modify, and delete all policies within that organization.

People Container Admin. By default, any user entry in an newly created organization is a member of that organization's People Container. The People Container Administrator has read and write access to all user entries in the organization's People Container. Keep in mind that this role DOES NOT have read and write access to the attributes that contain role and group DNs therefore, they cannot modify the attributes of, or remove a user from, a role or a group.

NOTE Other containers can be configured with Access Manager to hold user entries, group entries or even other containers. To apply an Administrator role to a container created after the organization has already been configured, the Container Admin Role or Container Help Desk Admin defaults would be used.

Group Admin. The Group Administrator has read and write access to all members of a specific group, and can create new users, assign users to the groups they manage, and delete the users the that they have created.

When a group is created, the Group Administrator role is automatically generated with the necessary privileges to manage the group. The role is not automatically assigned to a group member. It must be assigned by the group's creator, or anyone that has access to the Group Administrator Role.

Top-level Admin. The Top-level Administrator has read and write access to all entries in the top-level organization. In other words, this Top-level Admin role has privileges for every configuration principal within the Access Manager application.

Organization Admin. The Organization Administrator has read and write access to all entries in an organization. When an organization is created, the Organization Admin role is automatically generated with the necessary privileges to manage the organization.

3. Click New in the Navigation pane. The New Role template appears in the Data pane.
4. Select Static Role and enter a name. Click Next.
5. Enter a description of the role.
6. Choose the role type from the Type menu.

   The role can be either an Administrative role or a Service role. The role type is used by the console to determine and here to start the user in the Access Manager console. An administrative role notifies the console that the possessor of the role has administrative privileges; the service role notifies the console that the possessor is an end user.
7. Choose a default set of permissions to apply to the role from the Access Permission menu.The permissions provide access to entries within the organization. The default permissions shown are in no particular order. The permissions are:

   No permissions. No permissions are to be set on the role.

   Organization Admin. The Organization Administrator has read and write access to all entries in the configured organization.

   Organization Help Desk Admin. The Organization Help Desk Administrator has read access to all entries in the configured organization and write access to the `userPassword` attribute.

   Organization Policy Admin. The Organization Policy Administrator has read and write access to all policies in the organization. The Organization Policy Administrator can not create a referral policy to a peer organization.

   Generally, the No Permissions ACI is assigned to Service roles, while Administrative roles are assigned any of the default ACIs.
8. Click Finish.

   The created role is displayed in the Navigation pane and status information about the role is displayed in the Data pane.

   You can optionally configure the Display Options and Available Actions by selecting them in the View menu. For more information, see Display Options and Available Actions at the end of this chapter.

To Add Users to a Static Role

1. Select the role to modify and click on the Properties arrow.

2. Choose Users from the View menu in the Data pane.
3. Click Add.
4. Enter the information for the search criteria. You can choose to search for users based on one or more the displayed fields The fields are:

   Match. Allows you to include an operator for any the fields you wish to include for the filter. ALL returns users for all specified fields. ANY returns users for any one of the specified fields.

   First Name. Search for users by their first name.

   User Status. Search for users by their status (active or inactive).

   User ID. Search for a user by User ID.

   Last Name. Search for users by their last name.

   Full Name. Search for users by their full name.
5. Click Next to begin the search. The results of the search are displayed.
6. Choose the users from the names returned by selecting the checkbox next to the user name.
7. Click Finish.

   The Users are now assigned to the role.

To Create a Filtered Role

1. In the Navigation pane, go the organization where the role will be created.
2. Choose Roles from the View menu.

   A set of default roles are created when an organization is configured, and are displayed in the Navigation pane. The default roles are:

   Container Help Desk Admin. The Container Help Desk Admin role has read access to all entries in an organizational unit and write access to the `userPassword` attribute in user entries only in this container unit.

   Organization Help Desk Admin. The Organization Help Desk Administrator has read access to all entries in an organization and write access to the `userPassword` attribute.

NOTE When a suborganization is created, remember that the administration roles are created in the suborganization, not in the parent organization.

Container Admin. The Container Admin role has read and write access to all entries in an LDAP organizational unit. In Access Manager, the LDAP organizational unit is often referred to as a container.

Organization Policy Admin. The Organization Policy Administrator has read and write access to all policies, and can create, assign, modify, and delete all policies within that organization.

People Container Admin. By default, any user entry in an newly created organization is a member of that organization's People Container. The People Container Administrator has read and write access to all user entries in the organization's People Container. Keep in mind that this role DOES NOT have read and write access to the attributes that contain role and group DNs therefore, they cannot modify the attributes of, or remove a user from, a role or a group.

NOTE Other containers can be configured with Access Manager to hold user entries, group entries or even other containers. To apply an Administrator role to a container created after the organization has already been configured, the Container Admin Role or Container Help Desk Admin defaults would be used.

Group Admin. The Group Administrator has read and write access to all members of a specific group, and can create new users, assign users to the groups they manage, and delete the users the that they have created.

When a group is created, the Group Administrator role is automatically generated with the necessary privileges to manage the group. The role is not automatically assigned to a group member. It must be assigned by the group's creator, or anyone that has access to the Group Administrator Role.

Top-level Admin. The Top-level Administrator has read and write access to all entries in the top-level organization. In other words, this Top-level Admin role has privileges for every configuration principal within the Access Manager application.

Organization Admin. The Organization Administrator has read and write access to all entries in an organization. When an organization is created, the Organization Admin role is automatically generated with the necessary privileges to manage the organization.

3. Click New in the Navigation pane. The New Role template appears in the Data pane.
4. Select Filtered Role and enter the name. Click Next.
5. Enter a description for the role.
6. Choose the role type from the Type menu.

   The role can be either an Administrative role or a Service role. The role type is used by the console to determine and where to start the user in the Access Manager console. An administrative role notifies the console that the possessor of the role has administrative privileges; the service role notifies the console that the possessor is an end user.
7. Choose a default set of permissions to apply to the role from the Access Permission menu.
8. The permissions provide access to entries within the organization. The default permissions shown are in no particular order. The permissions are:

   No permissions. No permissions are to be set on the role.

   Organization Admin. The Organization Administrator has read and write access to all entries in the configured organization.

   Organization Help Desk Admin. The Organization Help Desk Administrator has read access to all entries in the configured organization and write access to the `userPassword` attribute.

   Organization Policy Admin. The Organization Policy Administrator has read and write access to all policies in the organization. The Organization Policy Administrator can not create a referral policy to a peer organization.

   Generally, the No Permissions ACI is assigned to Service roles, while Administrative roles are assigned any of the default ACIs.

9. Enter the information for the search criteria. The fields are:

   Match. Allows you to include an operator for any the fields you wish to include for the filter. ALL returns users for all specified fields. ANY returns users for any one of the specified fields.

   First Name. Search for users by their first name.

   User Status. Search for users by their status (active or inactive).

   User ID. Search for a user by User ID.

   Last Name. Search for users by their last name.

   Full Name. Search for users by their full name.

   Alternatively, you can select the Advanced button to define the filter attributes yourself. For example,

   ```
   (&(uid=user1)(|(inetuserstatus=active)(!(inetuserstatus=*))))
   ```

   If the filter is left blank, by default, the following role is created:

   ```
   (objectclass = inetorgperson)
   ```

   Click Cancel to cancel the role creation process.

10. Click Finish to initiate the search based on the filter criteria. The users defined by the filter criteria are automatically assigned to the role.

    You can optionally configure the Display Options and Available Actions by selecting them in the View menu. For more information, see Display Options and Available Actions at the end of this chapter.

NOTE You can add users to static roles through the Role profile page and/or the User profile page.

To Remove Users from a Role

1. Navigate to the Organization that contains the role to modify.

   Choose Organizations from the View menu in the Identity Management module and select the organization from the Navigation pane.

2. Choose Roles from the View menu.

3. Select the role to modify.

4. Choose Users from the View menu.

5. Select the checkbox next to each user to be removed.
6. Click Remove.

   The users are now removed from the role.

To Add a Role to a Policy

Access Manager objects are added to a policy through the policy's subject definition. When a policy is created or modified, organizations, roles, groups, and users can be defined as the subject in the policy's Subject page. Once the subject is defined, the policy will be applied to the object. For more information, see "Managing Policies" on page 132.

Customizing a Service to a Role

You can customize the services available to a role, and the access level for the service attributes, on a per-role basis. Each of the available services can be customized for a role by setting role-specific values to the attributes. You can also grant access for each of the services and to the services' attributes. There may be services that you wish only to be accessed by a specific type of user (for example, managers). To accomplish this, all users are assigned the service, but only the Manager type belonging to the role is allowed access to the specific service.

The same logic applies to service attributes. A user's account consists of many attributes, some of which the user may not be allowed to access; for example the account expiration date. The administrator of the account can be granted access to this attribute, but the user (the account owner) is not. Customizing the service and attribute access is accomplished through the role's Service view in the Navigation pane.

You must first add the services at the organization level in order to display the services. Users that are added to the role will inherit the role's service attributes.

To Configure Services

1. In the role's Service view, go to the section labeled Service Configuration for this Role.
2. Choose a service that is to be granted to the role by clicking on the Edit link next to the service name.

   If you have not created a service template, you will be prompted to do so. Click Yes.
3. Modify the Service attributes. For more information on specific Service attributes, see Part 3 of this manual, the *Attribute Reference Guide*.

4. Click Save.

NOTE When access to a service is denied (not checked), the service will not be displayed in the Access Manager console for the user possessing the role. Additionally, it is not possible to register or unregister a user, assign the service to a user, or create, delete, view or modify the Service template.

To Customize Attribute Access

1. In the role's Services view, go to the section labeled Service Access for this Role.
2. Choose the enable or disable status for the service you wish to modify. Enable allows the access modifications. Disable disallows the access modifications.
3. Click the Modify Access link.
4. Assign an access level to an attribute by selecting the Read / Write or Read Only check boxes.
5. Click OK and then Save.

For more information on specific Service attributes, see Part 4 of this manual, the *Attribute Reference.*

To Add a Role to a Policy

Access Manager objects are added to a policy through the policy's subject definition. When a policy is created or modified, organizations, roles, groups, and users can be defined as the subject in the policy's Subject page. Once the subject is defined, the policy will be applied to the object. For more information, see "Managing Policies" on page 132.

To Delete a Role

1. Navigate to the organization that contains the role to be deleted.
2. Choose Organizations from the View menu in Identity Management and select the organization from the Navigation pane. The Location path displays the default top-level organization and chosen organization.
3. Choose Roles from the View menu.
4. Select the checkbox next to the name of the role.
5. Click Delete.

Policies

Policies define rules to help protect an organization's web resources. Although policy creation, modification and deletion is performed through the Identity Management module, the procedures are described in "Creating Policies" on page 129.

Agents

Access Manager Policy Agents protect content on web servers and web proxy servers from unauthorized intrusions. They control access to services and web resources based on the policies configured by an administrator.

The *agent* object defines a Policy Agent profile, and allows Access Manager to store authentication and other profile information about a specific agent that is protecting an Access Manager resource. Through the Access Manager console, administrators can view, create, modify and delete agent profiles.

To Create an Agent

1. Navigate to the organization that contains the agent to be created.
2. Choose Agents from the View menu.
3. Click New.
4. Enter the values for the fields. The fields are:

   Name. Enter the name or identity of the agent. This is the name that the agent will use to log into Access Manager. Multi-byte names are not accepted.

   Password. Enter the agent password. This password must match the password used by the agent during LDAP authentication.

   Confirm Password. Confirm the password.

   Description. Enter a brief description of the agent. For example, you can enter the agent instance name or the name of the application it is protecting.

   Agent Key Value. Set the agent properties with a key/value pair. This property is used by Access Manager to receive agent requests for credential assertions about users. Currently, only one property is valid and all other properties will be ignored. Use the following format:

agentRootURL=http://*server_name:port/*

Device Status. Enter the device status of the agent. If set to Active, the agent will be able to authenticate to and communicate with Access Manager. If set to Inactive, the agent will not be able to authenticate to Access Manager.

To Delete an Agent

1. Navigate to the organization that contains the agent to be deleted.
2. Choose Agent from the View menu.
3. Select the checkbox next to the name of the agent.
4. Click Delete.

Creating a Unique Policy Agent Identity

By default, when you create multiple policy agents in a trusted environment, the policy agents contain the same UID and password. Because the UID and passwords are shared, Access Manager cannot distinguish between the agents, which may leave the session cookie open to interception.

The weakness may be present when an Identity Provider provides authentication, authorization and profile information about a user to applications (or Service Providers) that are developed by third parties or by unauthorized groups within the enterprise. Possible security issues are:

- All applications share the same `http` session cookie. This makes it possible for a rogue application to hijack the session cookie and impersonate the user to another application.
- If the application does not use the `https` protocol, the session cookie is prone to network eavesdropping.
- If just one application can be hacked, the security of the entire infrastructure is in jeopardy of being compromised.
- A rouge application can use the session cookie to obtain and possibly modify the profile attributes of a user. If the user has administrative privileges, the application would be able to do a lot more damage.

To Create a Unique Policy Identity

1. Use the Access Manager administration console to make an entry for each agent. For more information, see “To Create an Agent” on page 105.

2. Run the following command on the password that was entered in step 1b.

   AccessManager-base/SUNWam/agents/bin/crypt_util agent123

   This will give the following output:

   WnmKUCg/y3l404ivWY6HPQ==

3. Change AMAgent.properties to reflect the new value, and then and restart the agent. Example:

```
# The username and password to use for the Application authentication
module.


com.sun.am.policy.am.username = agent123
com.sun.am.policy.am.password = WnmKUCg/y3l404ivWY6HPQ==


# Cross-Domain Single Sign On URL

# Is CDSSO enabled.

com.sun.am.policy.agents.cdsso-enabled=true


# This is the URL the user will be redirected to after successful login

# in a CDSSO Scenario.

com.sun.am.policy.agents.cdcservletURL =
http://server.example.com:port/amserver/cdcservlet
```

4. Change AMConfig.properties to reflect the new values, and then and restart Access Manager. Example:

```
com.sun.identity.enableUniqueSSOTokenCookie=true

com.sun.identity.authentication.uniqueCookieName=sunIdentityServerAuthN
Server


com.sun.identity.authentication.uniqueCookieDomain=example.com
```

5. In the Access Manager administration console, choose Service Configuration>Platform.

6. In the Cookie Domains list, change the cookie domain name:

   a. Select the default `iplanet.com` domain, and then click Remove.

   b. Enter the host name of the Access Manager installation, and then click Add.

      Example: server.`example.com`

      You should see two cookies set on the browser:

Cookie	Host Name
`iPlanetDirectoryPro`	`server.example.com`
`sunIdentityServerAuthNServer`	`example.com`

Containers

The *container* entry is used when, due to object class and attribute differences, it is not possible to use an organization entry. It is important to remember that the Access Manager container entry and the Access Manager organization entry are not necessarily equivalent to the LDAP object classes `organizationalUnit` and `organization`. They are abstract Identity entries. Ideally, the organization entry will be used instead of the container entry.

NOTE The display of containers is optional. To view containers you must select Show Containers in View Menu in the Service Configuration module. For more information, see "Show Containers In View Menu" on page 263.

To Create a Container

1. Navigate to the Organization or Container where the new Container will be created.

   Select Containers from the View menu.

2. Click New.

   A Container template displays in the Data pane.

3. Enter the name of the Container to be created.

4. Click OK.

   You can optionally configure the Display Options and Available Actions by selecting them in the View menu. For more information, see Display Options and Available Actions at the end of this chapter.

To Delete a Container

1. Navigate to the organization or container which contains the container to be deleted.
2. Choose Containers from the View menu.
3. Select the checkbox next to the name of the container to be deleted.
4. Click Delete.

NOTE Deleting a container will delete all objects that exist in that Container. This includes all objects and sub containers.

People Containers

A *people container* is the default LDAP organizational unit to which all users are assigned when they are created within an organization. People containers can be found at the organization level and at the people container level as a sub People Container. They can contain only other people containers and users. Additional people containers can be added into the organization, if desired.

NOTE The display of people containers is optional. To view People Containers you must select Show People Containers in the Service Configuration module. For more information, see “Show People Containers” on page 262.

Create a People Container

1. Navigate to the organization or people container where the new people container will be created.

   Select People Containers from the View menu.

2. Click New.

   The People Container template displays in the Data pane.

3. Enter the name of the people container to be created.

4. Click OK.

Delete a People Container

1. Navigate to the organization or people container which contains the people container to be deleted.
2. Choose People Containers from the View menu.
3. Select the checkbox next to the name of the people container to be deleted.
4. Click Delete.

NOTE Deleting a people container will delete all objects that exist in that people container. This includes all users and sub people containers.

Group Containers

A *group container* is used to manage groups. It can contain only groups and other group containers. The group container Groups is dynamically assigned as the parent entry for all managed groups. Additional group containers can be added, if desired.

NOTE The display of group containers is optional. To view group containers you must select Show Group Containers in the Service Configuration module. For more information, see "Show Group Containers" on page 263.

To Create a Group Container

1. Navigate to the organization or the group container which contains the group container to be created.
2. Choose group containers from the View menu.

   The default Groups was created during the organization's creation.
3. Click New.
4. Enter a value in the Name field and click OK. The new group container displays in the Navigation pane.

To Delete a Group Container

1. Navigate to the organization which contains the group container to be deleted.

2. Choose Group Containers from the View menu.

   The default Groups and all created group containers display in the Navigation pane.

3. Select the checkbox next to the group container to be deleted.

4. Click Delete.

Display Options

For organizations, roles and containers, you can use Display Options view to customize the way in which Access Manager objects are displayed in the Access Manager console. Not all display options are available for all object types.

To Change the Display Options

1. Click on the Properties arrow of the organization for which you would like to change the display options.

2. Select Display Options from the View menu in the Data pane.

3. Edit the properties in the General section. The properties are:

   Generate Full Name Attribute. Select this attribute to enable Access Manager to always generate the user's full name, which is formed from the first and last name values in the user's profile.

   Always Select First Entry. Select this attribute for a search so that it automatically selects the first item of a given identity object type in the Navigation pane and displays it in the Data pane.

   User Profile Page Title. Choose an attribute from this pull-down menu to be used for the title in the User Profile Page.

   Disable Initial Search. This value disables the initial Access Manager search for one or more identity object types. Disabling the initial Search may enhance performance and reduce the likelihood of a timeout error.

4. Change the display options in the Display Configuration of Access Manager Objects section. This section allows you to customize how Access Manager containers and objects are displayed. The Access Manager Containers option allows you to specify which object views are displayed in the Navigation pane's View menu. The Access Manager Objects field allows you to specify which object views are displayed in the Data pane's View menu.

5. Click Save.

Available Actions

For certain Access Manager object types, you can define user access rights through the Available Actions view.

To Set Available Actions for Users

1. Click on the Properties arrow of the Identity object for which you would like to set available actions.
2. Select Available Actions from the View menu in the Data pane.
3. Choose the action type available for any Access Manager object. The action type defines the user's accessibility for each object. The action types are:

   No Access. The user has no access to this object.

   View. The user has read-only access to this object.

   Modify. The user can modify and view this object.

   Delete. The user can modify, view and delete this object.

   Full Access. The user can create, modify, view and delete this object.
4. Click Save. To change the values to their previously saved state, click Reset.

Chapter 5

Current Sessions

This chapter describes the session management features of Sun Java™ System Access Manager 6 2005Q1. The Session Management module provides a solution for viewing user session information and managing user sessions. It keeps track of various session times as well as allowing the administrator to terminate a session. System administrators should ignore the Load Balancer servers listed in the Platform Server list.

The Current Sessions Interface

The Current Sessions module interface allows an administrator, with the appropriate permissions, to view the session information for any user who is currently logged in to Access Manager.

Session Management Frame

The Session Management frame displays the name of the Access Manager that is currently being managed.

Session Information Window

The Session Information window displays all of the users who are currently logged into Access Manager, and displays the session time for each user. The display fields are:

User ID. Displays the user ID of the user who is currently logged in.

Time Left. Displays the amount of time (in minutes) remaining that the user has for that session before having to reauthenticate.

Max Session Time. Displays the maximum time (in minutes) that the user can be logged in before the session expires and must reauthenticate to regain access.

Idle Time. Displays the time (in minutes) that the user has been idle.

Max Idle Time. Displays the maximum time (in minutes) that a user can remain idle before having to reauthenticate.

The time limits are defined by the administrator in the Session Management Service. See “Session Service Attributes” on page 403 for more information.

You can display a specific user session, or a specific range of user sessions, by entering a string in the User ID field and clicking Filter. Wildcards are permitted.

Clicking the Refresh button will update the user session display.

Terminating a Session

Administrators with appropriate permissions can terminate a user session at any time. To do so:

1. Select the user session that you wish to terminate.
2. Click Terminate.

Chapter 6

Policy Management

This chapter describes the Policy Management feature of Sun Java™ System Access Manager 6 2005Q1. Access Manager's Policy Management feature provides a means for: the Top-level administrator or Top-level policy administrator to view, create, delete and modify policies for a specific service that can be used across all organizations. It also provides a way for an organization or suborganization administrator or policy administrator to view, create, delete and modify policies for specific use by the organization.

This chapter contains the following sections:

Overview

A *policy* defines rules that specify access privileges to an organization's protected resources. Businesses posses resources, applications and services that they need to protect, manage and monitor. Policies control the access permissions and usage of these resources by defining when and how a user can perform an action on a given resource. A policy, when applied to an object, defines the resources that a particular object can access.

NOTE An object is a principal. A *principal* can be an individual, a corporation, a role, or a group; anything that can have an identity. for more information, see the Java™ 2 Platform Standard Edition Javadocs.

A single policy can define either binary or non-binary decisions. A binary decision is *yes* / *no*, *true* / *false* or *allow* / *deny*. A non-binary decision represents the value of an attribute. For example, a mail service might include a mailboxQuota attribute with a maximum storage value set for each user. In general, a policy is configured to define what an object can do to which resource and under what conditions.

Policy Management Feature

The Policy Management feature provides a *policy service* for creating and managing policies. The policy service allows administrators to define, modify, grant, revoke and delete permissions to protect resources within the Access Manager deployment. Typically, a policy service includes a data store, a library of interfaces that allows for the creation, administration and evaluation of policies, and a policy enforcer or *policy agent*. Access Manager uses Sun Java System Directory Server for data storage, and provides Java and C APIs for policy evaluation and policy service customizaton. (see the *Access Manager Developer's Guide* for more information) It also allows administrator to use the Access Manager console for policy management. Access Manager provides one policy service, the URL Policy Agent service, which uses downloadable policy agents to enforce the policies.

URL Policy Agent Service

Out of the box, Access Manager provides the URL Policy Agent service for policy enforcement. This service allows administrators to create and manage policies through a policy enforcer or *policy agent*.

Policy Agents

The Policy Agent is the Policy Enforcement Point (PEP) for a server on which an enterprise's resources are stored. The policy agent is installed separately from Access Manager onto a web server and serves as an additional authorization step when a user sends a request for a web resource that exists on the protected web server. This authorization is in addition to any user authorization request which the resource performs. The agent protects the web server, and in turn, the resource is protected by the authorization plug-in.

For example, a Human Resources web server protected by a remotely-installed Access Manager might have an agent installed on it. This agent would prevent personnel without the proper policy from viewing confidential salary information or other sensitive data. The policies are defined by the Access Manager administrator, stored within the Access Manager deployment and used by the policy agent to allow or deny users access to the remote web server's content.

The most current Sun Java System Access Manager Policy Agents can be downloaded from the Sun Microsystems Download Center.

More information on installing and administrating the policy agents can be found in the *Sun Java System Access Manager J2EE Policy Agents Guide* or *Web Policy Agents Guide*.

NOTE Policy is evaluated in no particular order although as they are evaluated, if one action value evaluates to *deny*, subsequent policies are not evaluated, unless the Continue Evaluation On Deny Decision attribute is enabled in the Policy Configuration service. For more information, see "Policy Configuration Service Attributes" on page 385.

Policy agents enforce decisions only on web URLs (http://...). However, agents can be written using the Java and C Policy Evaluation APIs to enforce policy on other resources.

In addition, the Resource Comparator attribute in the Policy Configuration Service would also need to be changed from its default configuration to:

serviceType=*Name_of_LDAPService*|class=com.sun.identity.policy.plugins.SuffixResourceName|wildcard=*|delimiter=,|caseSensitive=false

Alternately, providing an implementation such as LDAPResourceName to implement com.sun.identity.policy.interfaces.ResourceName and configuring the Resource Comparator appropriately would also work.

NOTE The fields for the Resource Comparator attribute are explained in “Policy Configuration Service Attributes” on page 385.

The Policy Agent Process

The process for protected web resources begins when a web browser requests a URL that resides on a server protected by the policy agent. The server’s installed policy agent intercepts the request and checks for existing authentication credentials (a session token).

If the agent has intercepted a request and validated the existing session token, the following process is followed.

1. If the session token is valid, the user is allowed or denied access. If the token is invalid, the user is redirected to the Authentication Service, as outlined in the following steps.
2. The Authentication Service verifies that the credentials are also valid and issues a token.
3. Once the user’s credentials are properly authenticated, the agent issues a request to the Naming Service which defines the URLs used to access Access Manager’s internal services.
4. The Naming Service returns locators for the policy service, and the agent sends a request to the Policy Service to get policy decisions applicable to the user.
5. Based on the policy decisions for the resource being accessed, the user is either allowed or denied access. If advice on the policy decision indicates a different authentication level or authentication mechanism, the agent redirects the request to the Authentication Service until all criteria is validated.

Assuming the agent has intercepted a request for which there is no existing session token, the agent redirects the user to their default login page even if the resource is protected using a different authentication method.

NOTE Policy-based resource authorization and user authentication are separate types of authentication. More information on this can be found in “Policy-Based Resource Management” on page 143.

Policy Types

There are two types of policies that can be configured using Access Manager: a *normal* policy or a *referral* policy. A normal policy consists of *rules*, *subjects* and *conditions*. A referral policy consists of *rules* and *referrals* to organizations.

Normal Policy

In Access Manager, a policy that defines access permissions is referred to as a *normal* policy. A normal policy consists of *rules*, *subjects* and *conditions*.

Rules

A *rule* contains a resource, one or more actions, and a value. The rule, basically, defines the policy.

- A *resource* defines the specific object that is being protected; for instance, an HTML page or a user's salary information accessed using a human resources service.
- An *action* is the name of an operation that can be performed on the resource; examples of web server actions are POST or GET. An allowable action for a human resources service might be to be able to change a home telephone number.
- A *value* defines the permission for the action, for example, allow or deny.

NOTE It is acceptable to define an action without resources.

Subjects

A *subject* defines the user or collection of users (for instance, a group or those who possess a specific role) that the policy affects. Subjects are assigned to policies. The general rule for subjects is that the policy would apply only if the user is a member of at least one subject in the policy. The default subjects are:

- Authenticated Users
- Access Manager Roles
- LDAP Groups
- LDAP Roles

- LDAP Users
- Organization
- Web Services Client

Access Manager Roles Versus LDAP Roles

An Access Manager role is created using Access Manager. These roles have object classes mandated by Access Manager. An LDAP role is any role definition that uses the Directory Server role capability. These roles have object classes mandated by Directory Server role definition. All Access Manager roles can be used as Directory Server roles. However, all Directory Server roles are not necessarily Access Manager roles. LDAP roles can be leveraged from an existing directory by configuring the Policy Configuration Service. Access Manager roles can only be accessed through the hosting Access Manager Policy Service. Evaluating membership in Access Manager roles will be faster as it accesses the Access Manager SDK and cache. The LDAP Role Search filter can be modified in the Policy Configuration Service to narrow the scope and improve performance.

Nested Roles

Nested roles can be evaluated correctly as LDAP Roles in the subject of a policy definition.

Conditions

A condition allows you to define constraints on the policy. For example, if you are defining policy for a paycheck application, you can define a condition on this action limiting access to the application only during specific hours. Or, you may wish to define a condition that only grants this action if the request originates from a given set of IP addresses or from a company intranet.

The condition might additionally be used to configure different policies on different URIs on the same domain. For example, `http://org.example.com/hr/*jsp` can only be accessed by `org.example.net` from 9am to 5 p.m., yet `http://org.example.com/finance/*.jsp` can be accessed by `org.example2.net` from 5 a.m. to 11 p.m. This can be achieved by using an IP Condition along with a Time Condition. And specifying the rule resource as `http://org.example.com/hr/*.jsp`, the policy would apply to all the JSPs under `http://org.example.com/hr` including those in the sub directories.

NOTE The terms referral, rule, resource, subject, condition, action and value correspond to the elements *Referral*, *Rule*, *ResourceName*, *Subject*, *Condition*, *Attribute* and *Value* in the `policy.dtd`.

Policy Advices

If a policy is not applicable as determined by the condition, the condition can produce advice messages that indicates why the policy was not applicable to the request. These advice messages are propagated in the policy decision to the Policy Enforcement Point. The Policy Enforcement Point can retrieve this advice and try to take the appropriate action, such as redirecting the user back to the authentication mechanism to authenticate to a higher level. The user may then be prompted for higher level authentication and may be able to access to the resource, if the policy becomes applicable, after proper action for the advice is taken.

More information can be found in the following class:

`com.sun.identity.policy.ConditionDecision.getAdvices()`

Only `AuthLevelCondiiton` and `AuthSchemeCondition` provide advices if the condition is not satisfied.

`AuthLevelCondition` advice is assosciated with the following key:

`com.sun.identity.policy.plugin.AuthLevelCondition.AUTH_LEVEL_CONDITION_ADVICE`

`AuthSchemeCondition` advice is associated with the following key:

`com.sun.identity.policy.plugin.AuthLevelCondition.AUTH_SCHEME_CONDITION_ADVICE`

Custom conditions can also produce advices. However, the Access Manager Policy Agents respond only for Auth Level Advice and Auth Scheme Advice. Custom agents could be written to understand and respond to more advices and existing Access Manager agents can be extended to understand and respond to more advices. For more information, see the Policy Agents documentation at the following location:

`http://docs.sun.com/app/docs/coll/S1_IdServPolicyAgent_21`

Referral Policy

An administrator may need to delegate one organization's policy definitions and decisions to another organization. (Alternatively, policy decisions for a resource can be delegated to other policy products.) A *referral* policy controls this policy delegation for both policy creation and evaluation. It consists of one or more *rules* and one or more *referrals*.

Rules

A rule defines the resource whose policy definition and evaluation is being referred.

Referrals

The referral defines the organization to which the policy evaluation is being referred. By default, there are two types of referrals: peer organization and suborganization. They delegate to an organization on the same level and an organization on a sub-level, respectively. See "Creating Policies for Peer Organizations and Suborganizations" on page 131 for more information.

NOTE The referred-to organization can define or evaluate policies only for those resources (or sub-resources) that have been referred to it. This restriction, however, does not apply to the root organization.

Policy Definition Type Document

Once a policy is created and configured, it is stored in Directory Server in XML. In Directory Server, the XML-encoded data is stored in one place. Although policy is defined and configured using the amAdmin.dtd (or the console), it is actually stored in Directory Server as XML that is based on the policy.dtd. The policy.dtd contains the policy element tags extracted from the amAdmin.dtd (without the policy creation tags). So, when the Policy Service loads policies from Directory Server, it parses the XML based on the policy.dtd. The amAdmin.dtd is only used when creating policy with the command line. This section describes the structure of `policy.dtd`. The policy.dtd exists in the following location:

AccessManager-base`/SUNWam/dtd (Solairs)`

AccessManager-base`/identity/dtd (Linux)`

NOTE Throughout the rest of this chapter, only the Solaris directory information will be given. Please note that the directory structure for Linux is different.For more information, please see "About This Guide" on page 21.

Policy Element

Policy is the root element that defines the permissions or *rules* of a policy and to whom / what the rule applies or the *subject*. It also defines whether or not the policy is a *referral* (delegated) policy and whether there are any restrictions (or *conditions*) to the policy. It may contain one or more of the following sub-elements: *Rule*, *Conditions*, *Subjects*, or *Referrals*. The required XML attribute is name which specifies the name of the policy. The referralPolicy attribute identifies whether or not the policy is a referral policy; it defaults to a normal policy if not defined. Optional XML attributes include *name* and *description*.

NOTE When tagging a policy as *referral*, subjects and conditions are ignored during policy evaluation. Conversely, when tagging a policy as *normal*, any Referrals are ignored during policy evaluation.

Rule Element

The *Rule* element defines the specifics of the policy and can take three sub-elements: *ServiceName*, *ResourceName*, or *AttributeValuePair*. It defines the type of service or application for which the policy has been created as well as the resource name and the actions which are performed on it. A rule can be defined without any actions; for example, a referral policy rule doesn't have any actions.

NOTE It is acceptable to have a defined policy that does not include a defined *ResourceName* element.

ServiceName Element

The *ServiceName* element defines the name of the service to which the policy applies. This element represents the service type. It contains no other elements. The value is exactly as that defined in the service's XML file (based on the sms.dtd). The XML service attribute for the *ServiceName* element is the name of the service (which takes a string value).

ResourceName Element

The *ResourceName* element defines the object that will be acted upon. The policy has been specifically configured to protect this object. It contains no other elements. The XML service attribute for the *ResourceName* element is the name of the object. Examples of a *ResourceName* might be http://www.sunone.com:8080/images on a web server or ldap://sunone.com:389/dc=example,dc=com on a directory server. A more specific resource might be salary://uid=jsmith,ou=people,dc=example,dc=com where the object being acted upon is the salary information of John Smith.

AttributeValuePair Element

The *AttributeValuePair* element defines an action and its values. It is used as a sub-element to *Subject Element*, *Referral Element* and *Condition Element*. It contains both the *Attribute* and *Value* elements and no XML service attributes.

Attribute Element

The *Attribute* element defines the name of the action. An action is an operation or event that is performed on a resource. POST or GET are actions performed on web server resources, READ or SEARCH are actions performed on directory server resources. The *Attribute* element must be paired with a *Value* element. The *Attribute* element itself contains no other elements. The XML service attribute for the *Attribute* element is the name of the action.

Value Element

The *Value* element defines the action values. Allow / deny or yes / no are examples of action values. Other action values can be either boolean, numeric, or strings. The values are defined in the service's XML file (based on the sms.dtd). The *Value* element contains no other elements and it contains no XML service attributes.

CAUTION Deny rules always take precedence over allow rules. For example, if one policy denies access and another allows it, the result is a deny (provided all other conditions for both policies are met). It is recommended that deny policies be used with extreme caution as they can lead to potential conflicts. If explicit deny rules are used, policies assigned to a user through different subjects (such as role and/or group membership) may result in denied access. Typically, the policy definition process should only use allow rules. The default deny may be used when no other policies apply.

Subjects Element

The *Subjects* sub-element identifies a collection of objects to which the policy applies; this overview collection is chosen based on membership in a group, ownership of a role or individual users. It takes the *Subject* sub-element. The XML attributes that can be defined are:

name. This defines a name for the collection.

description. This defines a description of the subject

includeType. This is not currently used.

Subject Element

The *Subject* sub-element identifies a collection of objects to which the policy applies; this collection pinpoints more specific objects from the collection defined by the Subjects element. Membership can be based on roles, group membership or simply a listing of individual users. It contains a sub-element, the AttributeValuePair Element. The required XML attribute is type,which identifies a generic collection of objects from which the specifically defined subjects are taken. Other XML attributes include name which defines a name for the collection and includeType which defines whether the collection is as defined, for whether the policy applies to users who are NOT members of the subject.

NOTE When multiple subjects are defined, at least one of the subjects should apply to the user for the policy to apply. When a subject is defined using includeType set to false, the user should not be a member of that subject.

Referrals Element

The *Referrals* sub-element identifies a collection of policy referrals. It takes the *Referral* sub-element. The XML attributes it can be defined with are name which defines a name for the collection and description which takes a description.

Referral Element

The *Referral* sub-element identifies a specific policy referral. It takes as a sub-element the AttributeValuePair Element. It's required XML attribute is type which identifies a generic collection of assignments from which the specifically defined referrals are taken. It can also include the name attribute which defines a name for the collection.

Conditions Element

The *Conditions* sub-element identifies a collection of policy restrictions (time range, authentication level, et.al.). It must contain one or more of the *Condition* sub-element. The XML attributes it can be defined with are name which defines a name for the collection and description which takes a description.

NOTE The conditions element is an optional element in a policy.

Condition Element

The *Condition* sub-element identifies a specific policy restriction (time range, authentication level, et.al.). It takes as a sub-element the AttributeValuePair Element. Its required XML attribute is type which identifies a generic collection of restrictions from which the specifically defined conditions are taken. It can also include the name attribute which defines a name for the collection.

Adding a Policy Service

By default, Access Manager provides the URL Policy Agent service (`iPlanetAMWebAgentService`). This service is defined in an XML file located in the following directory:

```
etc/opt/SUNWam/config/xml/
```

You can, however add additional policy services to Access Manager. Once the policy service is created, you add it to Access Manager through the `amadmin` command line utility.

To Add a New Policy Service

1. Develop the new policy service in an XML file based on the `sms.dtd`. Access Manager provides two policy service XML files that you may wish to use as the basis for the new policy service file:

   `amWebAgent.xml` - This the the XML file for the default URL Policly Agent service. It is located in `etc/opt/SUNWam/config/xml/`.

   `SampleWebService.xml`. - This is the sample policy service file located in `/etc/opt/SUNWam/samples/policy`.

2. Save the XML file to the directory from which you will load the new policy service. For example:

   `/etc/opt/SUNWam/config/xml/newPolicyService.xml`

3. Load the new policy service with the `amadmin` command line utility. For example:

   ```
   AccessManager-base/SUNWam/bin/amadmin
       --runasdn "uid=amAdmin,ou=People,default_org,root_suffix
       --password password
       --schema /etc/opt/SUNWam/config/xml/newPolicyService.xml
   ```

4. After you load the new policy service, you can define rules for the policy definitions through the Access Manager console or by loading a new policy through `amadmin`.

Creating Policies

You can create, modify and delete policies through the Policy API and the Access Manager console, and create and delete policies through the `amadmin` command line tool. This section focuses on creating policies through the `amadmin` command line utility and through the Access Manager console. For more information on the Policy APIs, see the *Access Manager Developer's Guide*.

Policies are generally created through and XML file and added to Access Manager through the `amadmin` command line utility and then managed through the Access Manager console (although policies can be created through the console). This is because policies cannot be modified using `amadmin` directly. To modify a policy, you must first delete the policy from Access Manager and the add the modified policy using `amadmin`.

In general, policy is created at the organization (or suborganization) level to be used throughout the organization's tree.

Creating Policies With amadmin

1. Create the policy's XML file based on the `policy.dtd`. This file is located in the following directory:

   AccessManager-base`/SUNWam/dtd`

2. Once the policy's XML file is developed, you can use the following command to load it:

AccessManager-base`/SUNWam/bin/amadmin`

`--runasdn "uid=amAdmin,ou=People,`*default_org*`,`*root_suffix*`"`

`--password` *password*

`--data` *policy.xml*

To add multiple policies simultaneously, place the policies in one XML file, as opposed to having one policy in each XML file. If you load policies with multiple XML files in quick succession, the internal policy index may become corrupted, and some policies may not participate in policy evaluation.

When creating policies through `amadmin`, ensure that the authentication module is registered with the organization while creating authentication scheme condition; that the corresponding LDAP objects (organizations, groups, roles and users) exist while creating organizations', LDAP groups', LDAP roles' and LDAP users' subjects; that Access Manager roles exist while creating `IdentityServerRoles` subjects; and that the relevant organizations exist while creating sub organization or peer organization referrals.

Please note that in the text of Value elements in `SubOrgReferral`, `PeerOrgReferral`, `Organization` subject, `IdentityServerRoles` subject, `LDAPGroups` subject, `LDAPRoles` subject and `LDAPUsers` subject need to be the full DN.

To Create Policies With the Access Manager Console

1. Navigate to the Identity Management interface.

2. Choose the organization for which you would like to create a policy.

   Ensure that the location of the Policy Management window is correct for your organization.

3. Choose Policies from the View menu.

   By default, the Organizations view is visible in the View menu. All suborganizations configured, if any, will be visible below it. If creating policies for a suborganization, choose the suborganization and then choose Policies from the View menu.

4. Click New in the Navigation frame. The New Policy window opens.

5. Select the type of policy, normal or referral, that you wish to create.

   If a referral policy that refers to a suborganization does not exist, you will not be able to create any polices for that suborganization.

   It is not necessary to define all of the fields for normal or referral policies at this time. You may create the policy, then add rules, subjects, referrals, and so forth, later.

6. Type a name for the policy and click OK.

7. By default, the General view is displayed.

   The General view displays the name of the policy and allows you to enter a description of the policy that is to be created.

8. Click Save to complete the policy's configuration.

Creating Policies for Peer Organizations and Suborganizations

In order to create policies for peer or suborganizations, you must first create a referral policy in the parent (or another peer) organization. Also, the Policy Configuration service should be registered and the template created in the suborganizations. The referral policy must contain, in its rule definition, the resource prefix that is being managed by the suborganization. Once the referral policy is created in the parent organization (or another peer organization), normal policies can be created at the suborganization (or peer organization).

In this example, `o=isp` is the parent organization, `o=example.com` is the suborganization and manages resources and sub-resources of `http://www.example.com`.

To Create a Policy for a Suborganization

1. Create a referral policy at `o=isp`. For information on referral policies, see the procedure "Modifying a Referral Policy" on page 139.

   The referral policy must define `http://www.example.com` as the resource in the rule, and must contain a SubOrgReferral with `example.com` as the value in the referral.

2. Go to the Organization view and navigate to the suborganization `example.com`.

3. Ensure that the policy configuration service is registered at the suborganization level, `example.com`. For information, see "Adding Policy Configuration Services" on page 142.

4. Now that the resource is referred to `sun.com` by `isp`, normal policies can be created for the resource `http://www.example.com`, or for any resource starting with `http://www.example.com`.

   See the procedure "Modifying a Normal Policy" on page 132 for information on creating normal policies.

   To define policies for other resources managed by `example.com`, additional referral policies must be created at `o=isp`.

Managing Policies

Once a normal or referral policy is created and added to Access Manager, you can manage the policy through the Access Manager console by modifying the rules, subjects, conditions and referrals.

Modifying a Normal Policy

Through the Identity Management interface, you can create a policy that defines access permissions. Such a policy is referred to as a *normal* policy. A normal policy can consist of multiple rules, subjects, and conditions. This section lists and defines the default fields that you can specify when creating a normal policy.

To Modify Rules

1. From the Identity Management interface, select Policies from the View menu.

   The policies that were created for that organization are displayed.

2. Choose the policy you wish to modify and click the Properties arrow. The Edit Policy window is opened in the Data frame.

   By default, the General view is displayed. The attributes contained in the General view are described in "Creating Policies" on page 129.

3. Select Rules from the View menu and click New.

   If more than one service exists, they will be listed in the Data pane. Choose the service for which you wish to create a policy and click Next. The New Rule window is displayed.

4. Define the resource, actions and action values in the Rules fields. The fields are:

   Type. Displays the service for the policy to be created. The default is URL Policy Agent.

   Rule Name. Enter the name of the rule.

   Resource Name. Enter the name of a resource. For example:

   `http://www.example.com`

   Currently, Policy Agents only support `http://` and `https://` resources and do not support IP addresses in place of the hostname.

   Wildcards are supported for resource names, port number and protocol. For example:

   `http*://*:*/*.html`

   For the URL Policy Agent service, if a port number is not entered, the default port number is 80 for `http://`, and 443 for `https://`.

   To allow the management of resource for all servers installed on a specific machine, you can define the resource as `http://host*:*`. Additionally, you can define the following resource to grant an administrator to a specific organization authority for all of the services in that organization:

   `http://*.subdomain.domain.topleveldomain`

   Select Actions. For the URL Policy Agent Service, you can select either or both of the following default actions:

   - `GET`
   - `POST`

   Select Action Values. For the URL Policy Agent Service, you can choose one of the following action values:

- `Allow` lets you access the resource matching the resource defined in the rule.
- `Deny` denies access to the resource matching the resource defined in the rule.

Denial rules always take precedence over allow rules in a policy. For example, if you have two policies for a given resource, one denying access and the other allowing access, the result is a deny access (provided that the conditions for both policies are met). It is recommended that deny policies be used with extreme caution as they may lead to potential conflicts between the policies. Typically, the policy definition process should only use allow rules, and use the default deny when no policies apply to accomplish the deny case.

If explicit deny rules are used, policies that are assigned to a given user through different subjects (such as role and / or group membership) may result in denied access to a resource even if one or more of the policies allow access. For example, if there is a deny policy for a resource applicable to an Employee role and there is another allow policy for the same resource applicable to Manager role, policy decisions for users assigned both Employee and Manager roles would be denied.

One way to resolve such problems is to design policies using Condition plug-ins. In the case above, a “role condition” that applies the deny policy to users authenticated to the Employee role and applies the allow policy to users authenticated to the Manager role helps differentiate the two policies. Another way could be to use the `authentication level` condition, where the Manager role authenticates at a higher authentication level. See “To Add or Modify Conditions” on page 137 for more information.

NOTE If the service is defined so that an action does not need resource definitions, the resource field will not be displayed. If the service contains both types of actions (some requiring resources, some without resources), an option is displayed to select rules with actions requiring no resources, or rules with actions requiring resources.

5. Click Finish to save the rule. This only saves the configuration in memory. Follow step 7 to complete the process.
6. Repeat steps 1 through 5 to create additional rules.

7. All of the rules created for that policy are displayed in the table in the Rules view. Click Save to add the rules to the policy.

   To remove a rule from a policy, select the rule and click Remove.

   You can edit any rule definition by clicking on the Edit link next to the rule name.

To Modify Subjects

1. To define the subject for the policy, select Subject from the View menu and click New.
2. Select one of the default subject types:

   Authenticated Users. This subject type implies that any user with a valid SSOToken is a member of this subject.

   All authenticated users would be member of this Subject, even if they have authenticated to an organization that is different from the organization in which the policy is defined. This is useful if the resource owner would like to give access to resources that is managed for users from other organizations. If you want to restrict access to resources being protected to members of a specific organization, please use the Organization subject.

   Access Manager Roles. This subject type implies that any member of an Access Manager role is a member of this subject. An Access Manager role is created using Access Manager. These roles have object classes mandated by Access Manager. Access Manager roles can only be accessed through the hosting Access Manager Policy Service.

   LDAP Groups. This subject type implies that any member of an LDAP group is member of this subject.

   LDAP Roles. This subject type implies that any member of an LDAP role is a member of this subject. An LDAP Role is any role definition that uses the Directory Server role capability. These roles have object classes mandated by Directory Server role definition. The LDAP Role Search filter can be modified in the Policy Configuration Service to narrow the scope and improve performance.

   LDAP Users. This subject type implies that any LDAP user is a member of this subject.

   Organization. This subject type implies that any member of an organization is a member of this subject.

   Web Services Client. This subject type implies that a web service client (WSC) identified by the SSOToken is a member of this subject, if the DN of any principal contained in the SSOToken matches any selected value of this subject. Valid values are the DNs of trusted certificates in the local JKS keystore, which correspond to the certificates of trusted WSCs. This subject has dependency on the Liberty Web Services Framework and should be used only by Liberty Service Providers to authorize WSCs.

   Make sure that you have created the keystore before you add this Subject to a

policy. Information on setting up the keystore can be found in the following location:

AcessManager-base/SUNWam/samples/saml/xmlsig/keytool.html

Click Next to continue.

3. Enter a name for the subject.
4. Select or deselect the Exclusive field.

   If this field is not selected (default), the policy applies to the identity that is a member of the subject. If the field is selected, the policy applies the identity that is not a member of the subject.

   If multiple subjects exist in the policy, the policy applies to the identity when at least one of the subjects implies that the policy applies to the given identity.
5. Perform a search in order to display the identities to add to the subject. This step is not applicable for the Authenticated Users subject or Web Services Client subjects.

   The default (*) search pattern will display all qualified entries.
6. Select the individual identities you wish to add for the subject, or click Add All to add all of the identities at once. Click Add to move the identities to the Select List Box. This step is not applicable for the Authenticated Users subject or Web Services Client subjects.
7. Click Finish.
8. The subject's names, type and exclusive status are displayed in the table in the Subjects view. Click Save.

   To remove a subject from a policy, select the subject and click Delete, then Save.

   You can edit any subject definition by clicking on the Edit link next to the subject name.

To Add or Modify Conditions

1. Select Conditions from the View menu. Click New to add a new condition, or click the Edit link to edit an existing condition.
2. Select one of the following default conditions:
   - Authentication Level
   - Authentication Scheme

- IP Address
- LE Authentication Level
- Session
- Time

For Authentication Level, the policy applies if the user's authentication level is greater than or equal to the Authentication level set in the condition. For LE Authentication Level, the policy applies if the user's authentication level is less than or equal to the Authentication level set in the condition

3. Click Next.
4. Define the values for a given condition. The fields are:

   Name. Enter the name of the condition.

   Authentication Level

   Authentication level. Indicate the level of trust for authentication. The available authentication levels are displayed in the authentication level and authentication module table.

   The authentication level condition can be used to specify levels other than those from the registered auth modules levels for that organization. This is useful when a policy applies to user authenticated from another organization.

   Authentication Scheme

   Authentication scheme. Choose the authentication scheme for the condition from the pull-down menu. These authentication schemes are taken from the Core service template in the organization authentication modules.

   IP Address

   IP Address From/To. Specifies the range of the IP address.

   DNS Name. Specifies the DNS name. This field can be a fully qualified hostname or a string in one of the following formats:

   domainname

   **.domainname*

   Time

   Date From/To. Specifies the range of the date.

   Time. Specifies the range of time within a day.

Day. Specifies a range of days.

Timezone. Specifies a timezone, either standard or custom. Custom timezones can only be a timezone ID recognized by Java (for example, PST). If no value is specified, the default value is the Timezone set in the Access Manager JVM.

Session

Max Session Time. Specifies the maximum user session time during which a policy applies.

Terminate Session. If selected, the user session will be terminated if the session time exceeds the maximum allowed as defined in the Max Session Time field.

5. Once you have defined the condition, click Finish.

   All of the conditions created for that policy are displayed in the table in the Conditions view.

6. Click Save.

   To remove a condition from a policy, select the condition and click Delete.

   You can edit any condition definition by clicking on the Edit link next to the condition name.

Modifying a Referral Policy

Through the Identity Management interface you can delegate an organization's policy definitions and decisions to another organization. (You can also delegate policy decisions for a resource to other policy products.) A *referral* policy controls this policy delegation for both policy creation and evaluation. It consists of a *rule* and the *referral* itself.

To Modify Rules

1. Select Rules from the View menu. Click New to add a new rule, or click the Edit link to edit an existing rule.
2. Select the Service Type. Click Next if you are creating a new rule.
3. Define the resource in the Rules fields. The fields are:

   Type. Displays the policy service for the policy to be created.

   Rule Name. Enter the name of the rule.

   Resource Name. Enter the name of a resource. For example:

`http://www.sunone.com`

Currently, Policy Agents only support `http://` and `https://` resources and do not support IP addresses in place of the hostname.

Wildcards are supported for resource names, port number and protocol.

For the URL Policy Agent service, if a port number is not entered, the default port number is 80 for `http://`, and 443 for `https://`.

To allow the management of resource for all servers installed on a specific machine, you can define the resource as `http://host*:*`. Additionally, you can define the following resource to grant an administrator to a specific organization authority for all of the services in that organization:

`http://*.subdomain.domain.topleveldomain`

4. Click Finish.
5. Repeat steps 1 through 4to create additional rules.

   All of the rules created for that policy are displayed in the table in the Rules view.
6. Click Save.

   To remove a rule from a policy, select the rule and click Delete.

   You can edit any rule definition by clicking on the Edit link next to the rule name.

To Add Referrals

1. Select Referrals from the View menu. Click New to add a new referral, or click the Edit link to edit an existing referral.
2. Define the resource in the Rules fields. The fields are:

   Referral. Displays the current referral type.

   Name. Enter the name of the referral.

   Containing. Specifies a filter for the organization names that will be displayed in the Value field. By default, it will display all organization names.

   Value. Select the organization name of the referral.

3. Click OK and Save.

   To remove a referral from a policy, select the referral and click Delete.

   You can edit any referral definition by clicking on the Edit link next to the referral name.

Policy Configuration Service

The Policy Configuration service is used to configure policy-related attributes for each organization through the Access Manager console. You can also define resource name implementations and Directory Server data stores for use with the Access Manager Authentication service.

Caching Subject Evaluations

To improve policy evaluation performance, subject evaluations are cached for a period of minutes as defined by the Subjects Result Time To Live attribute in the Policy Configuration service. These cached policy decisions are referred to until the time defined in the Subjects Result Time To Live attribute has elapsed. Once this time has been reached, the next time the policy is evaluated its decision would reflect the user's changed state, if applicable (for example, if the user has been removed from a group).

amldapuser Definition

amldapuser is a user created during installation that is used to bind and search Directory Server during LDAP and Membership authentication. It is also used in the Policy Configuration service. Once the LDAP, Membership or Policy Configuration services are registered to an organization, the password for this user (configured during installation) must be entered. For more information, see the *Sun Java System Access Manager Migration Guide* .

Adding Policy Configuration Services

Adding a policy configuration service is the same as adding any type of service; it is done within the Identity Management interface. By default, the Policy Configuration service is automatically added to the top-level organization. Any policy service you create must be added to all organizations. Whenever you add the policy configuration service, you must enter the LDAP bind password in the template.

To Add the Policy Configuration Service

1. Navigate to the Identity Management interface.

   When the console opens, the default interface is Identity Management.

2. Choose the organization for which you would like to create policy.

   If logged in as the Top-Level Administrator, make sure that the location of the Identity Management module is the top-level organization where all configured organizations are visible. The default top-level organization is defined during installation.

3. Choose Services from the View menu.

   If the organization already has registered services, they will be displayed in the Navigation frame.

4. Click Add in the Navigation frame.

   A listing of services not yet registered to this organization is displayed in the Data frame.

5. From the Add Services window, opened in the Data frame, choose Policy Configuration and click OK.

   The Policy Configuration Service is added to the list of services in the Navigation frame.

6. Click the Properties arrow to configure the policy service.

   a. If the policy template has not yet been configured, you will need to create a service template for the newly registered policy service.

   b. To configure the policy service, click Create.

   c. Modify the Policy Configuration attributes. See “Policy Configuration Service Attributes” on page 385 for a description of these attributes.

7. Click Save.

   The policy configuration service is now added to the chosen organization.

NOTE Suborganizations must register their policy services independently of their parent organization. In other words, the suborganization `o=suborg,dc=sun,dc=com` will not inherit the policy configuration service from its parent `dc=sun,dc=com`.

Policy-Based Resource Management

Some organizations require an advanced authentication scenario where a user authenticates against a particular module based on the resource that they are attempting to access. Policy-based resource management is a function of Access Manager in which the user does not need to pass their default authentication module in order to access a web resource.

Limitations

Policy-based resource management contains the following limitations:

1. All policies applicable to the resource require the same authentication scheme or level of authentication. For example, if abc.html is defined in a policy for the LDAP authentication module, it can not be defined in a policy for the Certificate-based authentication module.
2. Level and scheme are the only conditions that can be defined for this policy.
3. This feature does not work across different DNS domains.

To Configure Policy-based Resource Management

Once both Access Manager and a policy agent have been installed, policy-based resource management can be configured. To do this, it is necessary to point Access Manager to the Gateway servlet.

1. Open AMAgent.properties.

   AMAgent.properties can be found (in a Solaris environment) in /etc/opt//SUNWam/agents/config/.

2. Comment out the following line:

   #com.sun.am.policy.am.loginURL = http://identity_server_host.domain_name:port/amserver/UI/Login.

3. Add the following line to the file:

   com.sun.am.policy.am.loginURL = http://identity_server_host.domain_name:port/amserver/gateway

4. Restart the agent.

Chapter 7

Managing Authentication

The Authentication Service provides a web-based user interface for all out-of-the-box authentication modules installed in the Access Manager deployment. This interface provides a dynamic and customizable means for gathering authentication credentials by displaying the login requirement screens (based on the invoked authentication module) to a user requesting access. The interface is built using Sun Java System™ Application Framework (sometimes referred to as *JATO*), a Java 2 Enterprise Edition (J2EE) presentation framework used to help developers build functional web applications.

The User Interface Login URL

The Authentication Service user interface is accessed by entering a login URL into the Location Bar of a web browser. This URL is:

http://*identity_server_host.domain_name:port*/*service_deploy_uri*/UI/Login

NOTE During installation, the *service_deploy_uri* is configured as amserver. This default service deployment URI will be used throughout this document.

The user interface login URL can also be appended with Login URL Parameters to define specific authentication methods or successful/failed authentication redirection URLs. Additional information on redirection URLs can be found in "Authentication Types" on page 152.

Login URL Parameters

A URL parameter is a name/value pair appended to the end of a URL. The parameter starts with a question mark (?) and takes the form name=value. A number of parameters can be combined in one login URL, for example:

```
http://server_name.domain_name:port/amserver/UI/Login?module=LDAP&locale=ja&goto=http://www.sun.com
```

If more than one parameter exists, they are separated by an ampersand (&). The combinations though must adhere to the following guidelines:

- Each parameter can occur only once in one URL. For example, `module=LDAP&module=NT` is not computable.
- Both the org parameter and the domain parameter determine the login organization. In this case, only one of the two parameters should be used in the login URL. If both are used and no precedence is specified, only one will take effect.
- The parameters `user`, `role`, `service`, `module` and `authlevel` are for defining authentication modules based on their respective criteria. Due to this, only one of them should be used in the login URL. If more than one is used and no precedence is specified, only one will take effect.

The following sections describe parameters that, when appended to the User Interface Login URL and typed in the Location bar of a web browser, achieve various authentication functionality.

TIP To simplify an authentication URL and parameters for distribution throughout an organization, an administrator might configure an HTML page with a simple URL that possesses links to the more complicated login URLs for all configured authentication methods.

goto Parameter

A goto=*successful_authentication_URL* parameter overrides the value defined in the Login Success URL of the Authentication Configuration service. It will link to the specified URL when a successful authentication has been achieved. A `goto=logout_URL` parameter can also be used to link to a specified URL when the user is logging out. For an example of a successful authentication URL:

```
http://server_name.domain_name:port/amserver/UI/Login?goto=http://www.sun.
com/homepage.html
```

An example goto logout URL:

```
http://server_name.domain_name:port/amserver/UI/Logout?goto=http://www.sun
.com/logout.html.
```

NOTE There is an order of precedence in which Access Manager looks for successful authentication redirection URLs. Because these redirection URLs and their order are based on the method of authentication, this order (and related information) is detailed in “Authentication Types” on page 152.

gotoOnFail Parameter

A `gotoOnFail=failed_authentication_URL` parameter overrides the value defined in the Login Failed URL of the Authentication Configuration service. It will link to the specified URL if a user has failed authentication. An example gotoOnFail URL might be http://*server_name.domain_name*:*port*/amserver/UI/Login?gotoOnFail=http://www.sun.com/auth_fail.html.

NOTE There is an order of precedence in which Access Manager looks for failed authentication redirection URLs. Because these redirection URLs and their order are based on the method of authentication, this order (and related information) is detailed in “Authentication Types” on page 152.

org Parameter

The `org=orgName` parameter allows a user to authenticate as a user in the specified organization.

TIP A user who is not already a member of the specified organization will receive an error message when they attempt to authenticate with the `org` parameter. A user profile, though, can be dynamically created in the Directory Server if all of the following are TRUE:

- The User Profile attribute in the Core Authentication Service must be set to Dynamic or Dynamic with User Alias.
- The user must successfully authenticate to the required module.
- The user does not already have a profile in Directory Server.

From this parameter, the correct login page (based on the organization and its locale setting) will be displayed. If this parameter is not set, the default is the top-level organization. For example, an `org` URL might be:

```
http://server_name.domain_name:port/amserver/UI/Login?org=sun
```

user Parameter

The `user=userName` parameter forces authentication based on the module configured in User Authentication Configuration attribute of the user's profile. For example, one user's profile can be configured to authenticate using the Certification module while another user might be configured to authenticate using the LDAP module. Adding this parameter sends the user to their configured authentication process rather than the method configured for their organization. For example:

```
http://server_name.domain_name:port/amserver/UI/Login?user=jsmith
```

role Parameter

A `role=roleName` parameter sends the user to the authentication process configured for the specified role. A user who is not already a member of the specified role will receive an error message when they attempt to authenticate with this parameter. For example:

```
http://server_name.domain_name:port/amserver/UI/Login?role=manager.
```

locale Parameter

Access Manager has the capability to display localized screens (translated into languages other than English) for the authentication process as well as for the console itself. The `locale=`*`localeName`* parameter allows the specified locale to take precedence over any other defined locales. The login locale is displayed by the client after searching for the configuration in the following places, order-specific:

1. Value of locale parameter in Login URL

   The value of the `locale=`*`localeName`* parameter takes precedence over all other defined locales.

2. Locale defined in user's profile

   If there is no URL parameter, the locale is displayed based on the value set in the User Preferred Language attribute of the user profile.

3. Locale defined in the HTTP header

   This locale is set by the web browser.

4. Locale defined in Core Authentication Service

   This is the value of the Default Auth Locale attribute in the Core Authentication module.

5. Locale defined in Platform Service

   This is the value of the Platform Locale attribute in the Platform service.

6. Operating system locale

The locale derived from this pecking order is stored in the user's session token and Access Manager uses it for loading the localized authentication module only. After successful authentication, the locale defined in the User Preferred Language attribute of the user's profile is used. If none is set, the locale used for authentication will be carried over. For example:

`http://server_name.domain_name:port/amserver/UI/Login?locale=ja.`

NOTE Information on how to localize the screen text and error messages can be found in the Access Manager Developer's Guide

module Parameter

The `module=moduleName` parameter allows authentication via the specified authentication module. Any of the modules can be specified although they must first be registered under the organization to which the user belongs and selected as one of that organization's authentication modules in the Core Authentication module. For example:

`http://server_name.domain_name:port/amserver/UI/Login?module=Unix.`

NOTE The authentication module names are case-sensitive when used in a URL parameter.

service Parameter

The `service=serviceName` parameter allows a user to authenticate via a service's configured authentication scheme. Different authentication schemes can be configured for different services using the Authentication Configuration service. For example, an online paycheck application might require authentication using the more secure Certificate Authentication module while an organization's employee directory application might require only the LDAP Authentication module. An authentication scheme can be configured, and named, for each of these services. For example:

`http://server_name.domain_name:port/amserver/UI/Login?service=sv1.`

NOTE The Authentication Configuration service is used to define a scheme for service-based authentication.

arg Parameter

The `arg=newsession` parameter is used to end a user's current session and begin a new one. The Authentication Service will destroy a user's existing session token and perform a new login in one request. This option is typically used in the Anonymous Authentication module. The user first authenticates with an anonymous session, and then hits the register or login link. For example:

`http://server_name.domain_name:port/amserver/UI/Login?arg=newsession.`

authlevel Parameter

An `authlevel=value` parameter tells the Authentication Service to call a module with an authentication level equal to or greater than the specified authentication level value. Each authentication module is defined with a fixed integer authentication level. For example:

`http://server_name.domain_name:port/amserver/UI/Login?authlevel=1.`

NOTE The Authentication Level is set in each module's specific profile. More information on this module can be found in the *Sun Java System Access Manager Administration Guide*.

domain Parameter

This parameter allows a user to login to an organization identified as the specified domain. The specified domain must match the value defined in the Domain Name attribute of the organization's profile. For example:

`http://server_name.domain_name:port/amserver/UI/Login?domain=sun.com.`

TIP A user who is not already a member of the specified domain/organization will receive an error message when they attempt to authenticate with the org parameter. A user profile, though, can be dynamically created in the Directory Server if all of the following points are TRUE:

- The User Profile attribute in the Core Authentication Service must be set to `Dynamic` or `Dynamic With User Alias`.
- The user must successfully authenticate to the required module.
- The user does not already have a profile in Directory Server.

iPSPCookie Parameter

The `iPSPCookie=yes` parameter allows a user to login with a persistent cookie. A persistent cookie is one that continues to exist after the browser window is closed. In order to use this parameter, the organization to which the user is logging in must have Persistent Cookies enabled in their Core Authentication module. Once the user authenticates and the browser is closed, the user can login with a new browser session and will be directed to console without having to reauthenticate. This will work until the value of the Persistent Cookie Max Time attribute specified in the Core Service elapses. For example:

`http://server_name.domain_name:port/amserver/UI/Login?org=example&iPSPCookie=yes`

IDTokenN Parameters

This parameter option to enables a user to pass authentication credentials using a URL or HTML forms. With the `IDTokenN=value` parameters, a user can be authenticated without accessing the Authentication Service User Interface. This process is called *Zero Page Login*. Zero page login works only for authentication modules that use one login page. The values of `IDToken0, IDToken1, ..., IDTokenN` map to the fields on the authentication module's login page. For example, the LDAP authentication module might use `IDToken1` for the `userID` information, and `IDToken2` for password information. In this case, the LDAP module IDTokenN URL would be:

`http://server_name.domain_name:port/amserver/UI/Login?module=LDAP&IDToken1=userID&IDToken2=password`

(`module=LDAP` can be omitted if LDAP is the default authentication module.)

For Anonymous authentication, the login URL parameter would be:

`http://server_name.domain_name:port/amserver/UI/Login?module=Anonymous&IDToken1=anonymousUserID.`

NOTE The token names `Login.Token0, Login.Token1, ..., Login.TokenN` (from previous releases) are still supported but will be deprecated in a future release. It is recommended to use the new `IDTokenN` parameters.

Authentication Types

The Authentication Service provides different ways in which authentication can be applied. These different authentication methods can be accessed by specifying Login URL parameters, or through the Authentication Programming Interfaces. Before an authentication module can be configured, the Core authentication service attribute Organization Authentication Modules must be modified to include the specific authentication module name.

The Authentication Configuration service is used to define authentication modules for any of the following authentication types:

- "Organization-based Authentication" on page 155
- "Role-based Authentication" on page 157
- "Service-based Authentication" on page 161
- "User-based Authentication" on page 164

- “Authentication Level-based Authentication” on page 167
- “Module Based Authentication” on page 170

Once an authentication module is defined for one of these authentication types, the module can be configured to supply redirect URLs, as well as a post-processing Java class specification, based on a successful or failed authentication process.

How Authentication Types Determine Access

For each of these methods, the user can either pass or fail the authentication. Once the determination has been made, each method follows this procedure. Step 1 through Step 3 follows a successful authentication; Step 4 follows both successful and failed authentication.

1. Access Manager confirms whether the authenticated user(s) is defined in the Directory Server data store and whether the profile is active.

   The User Profile attribute in the Core Authentication module can be defined as `Required`, `Dynamic`, `Dynamic with User Alias`, or `Ignored`. Following a successful authentication, Access Manager confirms whether the authenticated user(s) is defined in the Directory Server data store and, if the User Profile value is *Required*, confirms that the profile is active. (This is the default case.) If the User Profile is `Dynamically Configured`, the Authentication Service will create the user profile in the Directory Server data store. If the User Profile is set to `Ignore`, the user validation will not be done.

2. Execution of the Authentication Post Processing SPI is accomplished.

   The Core Authentication module contains an Authentication PostProcessing Class attribute which may contain the authentication post-processing class name as its value. `AMPostAuthProcessInterface` is the post-processing interface. It can be executed on either successful or failed authentication or on logout.

3. The following properties are added to, or updated in, the session token and the user’s session is activated.

   Organization. This is the DN of the organization to which the user belongs.

   Principal. This is the DN of the user.

   Principals. This is a list of names to which the user has authenticated. (This property may have more then one value defined as a pipe separated list.)

   UserId. This is the user’s DN as returned by the module, or in the case of modules other than LDAP or Membership, the user name. (All Principals must map to the same user. The UserID is the user DN to which they map.)

NOTE This property may be a non-DN value.

UserToken. This is a user name. (All Principals must map to the same user. The UserToken is the user name to which they map.)

Host. This is the host name or IP address for the client.

authLevel. This is the highest level to which the user has authenticated.

AuthType. This is a pipe separated list of authentication modules to which the user has authenticated (for example, `module1|module2|module3`).

clientType. This is the device type of the client browser.

Locale. This is the locale of the client.

CharSet. This is the determined character set for the client.

Role. Applicable for role-based authentication only, this is the role to which the user belongs.

Service. Applicable for service-based authentication only, this is the service to which the user belongs.

loginURL. This is the client's login URL.

4. Access Manager looks for information on where to redirect the user after either a successful or failed authentication.

   URL redirection can be to either an Access Manager page or a URL. The redirection is based on an order of precedence in which Access Manager looks for redirection based on the authentication method and whether the authentication has been successful or has failed. This order is detailed in the URL redirection portions of the following authentication methods sections.

URL Redirection

In the Authentication Configuration service, you can assign URL redirection for successful or unsuccessful authentication. The URLs, themselves, are defined in the Login Success URL and Login Failure URL attributes in this service. In order to enable URL redirection, you must add the Authentication Configuration service to your organization to make it available to configure for a role, organization, or user. Make sure that you add an authentication module, such as LDAP - REQUIRED, when adding the Authentication Configuration service. For more information, see "Authentication Configuration" on page 172.

Organization-based Authentication

This method of authentication allows a user to authenticate to an organization or sub-organization. It is the default method of authentication for Access Manager. The authentication method for an organization is set by registering the Core Authentication module to the organization and defining the Organization Authentication Configuration attribute.

Organization-based Authentication Login URLs

The organization for authentication can be specified in the User Interface Login URL by defining the `org` Parameter or the `domain` Parameter. The organization of a request for authentication is determined from the following, in order of precedence:

1. The `domain` parameter.
2. The `org` parameter.
3. The value of the `DNS Alias Names` (Organization alias names) attribute in the Administration Service.

After calling the correct organization, the authentication module(s) to which the user will authenticate are retrieved from the Organization Authentication Configuration attribute in the Core Authentication Service. The login URLs used to specify and initiate organization-based authentication are:

```
http://server_name.domain_name:port/amserver/UI/Login
```

```
http://server_name.domain_name:port/amserver/UI/Login?domain=domain_name
```

```
http://server_name.domain_name:port/amserver/UI/Login?org=org_name
```

If there is no defined parameter, the organization will be determined from the server host and domain specified in the login URL.

Organization-based Authentication Redirection URLs

Upon a successful or failed organization-based authentication, Access Manager looks for information on where to redirect the user. Following is the order of precedence in which the application will look for this information.

Successful Organization-based Authentication Redirection URLs

The redirection URL for successful organization-based authentication is determined by checking the following places in order of precedence:

1. A URL set by the authentication module.

2. A URL set by a goto Login URL parameter.
3. A URL set in the clientType custom files for the iplanet-am-user-success-url attribute of the user's profile (amUser.xml).
4. A URL set in the clientType custom files for the iplanet-am-auth-login-success-url attribute of the user's role entry.
5. A URL set in the clientType custom files for the iplanet-am-auth-login-success-url attribute of the user's organization entry.
6. A URL set in the clientType custom files for the iplanet-am-auth-login-success-url attribute as a global default.
7. A URL set in the iplanet-am-user-success-url attribute of the user's profile (amUser.xml).
8. A URL set in the iplanet-am-auth-login-success-url attribute of the user's role entry.
9. A URL set in the iplanet-am-auth-login-success-url attribute of the user's organization entry.
10. A URL set in the iplanet-am-auth-login-success-url attribute as a global default.

Failed Organization-based Authentication Redirection URLs

The redirection URL for failed organization-based authentication is determined by checking the following places in the following order:

1. A URL set by the authentication module.
2. A URL set by a gotoOnFail Login URL parameter.
3. A URL set in the clientType custom files for the iplanet-am-user-failure-url attribute of the user's entry (amUser.xml).
4. A URL set in the clientType custom files for the iplanet-am-auth-login-failure-url attribute of the user's role entry.
5. A URL set in the clientType custom files for the iplanet-am-auth-login-failure-url attribute of the user's organization entry.
6. A URL set in the clientType custom files for the iplanet-am-auth-login-failure-url attribute as a global default.
7. A URL set for the iplanet-am-user-failure-url attribute in the user's entry (amUser.xml).

8. A URL set for the `iplanet-am-auth-login-failure-url` attribute of the user's role entry.
9. A URL set for the `iplanet-am-auth-login-failure-url` attribute of the user's organization entry.
10. A URL set for the `iplanet-am-auth-login-failure-url` attribute as the global default.

To Configure Organization-Based Authentication

Authentication modules are set for an organization by first adding the Core Authentication service to the organization.

To configure the organization's authentication attributes:

1. Navigate to the organization for which you will configure the authentication attributes.
2. Select Services from the View menu.
3. Click the Core Properties arrow in the service listing.

   The Core authentication attributes are displayed in the Data pane.
4. Click Edit next to the Admin Authenticator attribute. This allows you to define the authentication services for administrators only. This attribute can be used if the authentication module for administrators needs to be different from the module for end users. The default authentication module is LDAP.

   Once you have defined the authentication services, click Save to save the changes, and click Close to return to the Core Authentication attributes for organizations.
5. Click the Edit link next to the Organization Authentication Configuration attribute. This allows you to define authentication modules for all users within the organization. The default authentication module is LDAP.
6. Once you have defined the authentication services, click Save to save the changes, and click Close to return to the Core Authentication attributes for organizations.

Role-based Authentication

This method of authentication allows a user to authenticate to a role (either static or filtered) within an organization or sub-organization.

NOTE The Authentication Configuration Service must first be registered to the organization before it can be registered as an instance to the role.

For authentication to be successful, the user must belong to the role and they must authenticate to each module defined in the Authentication Configuration Service instance configured for that role. For each instance of role-based authentication, the following attributes can be specified:

Conflict Resolution Level. This sets a priority level for the Authentication Configuration Service instance defined for different roles that both may contain the same user. For example, if `User1` is assigned to both `Role1` and `Role2`, a higher conflict resolution level can be set for `Role1` so when the user attempts authentication, `Role1` will have the higher priority for success or failure redirects and post-authentication processes.

Authentication Configuration. This defines the authentication modules configured for the role's authentication process.

Login Success URL. This defines the URL to which a user is redirected on successful authentication.

Login Failed URL. This defines the URL to which a user is redirected on failed authentication.

Authentication Post Processing Classes. This defines the post-authentication interface.

Role-based Authentication Login URLs

Role-based authentication can be specified in the The User Interface Login URL by defining a role Parameter. After calling the correct role, the authentication module(s) to which the user will authenticate are retrieved from the Authentication Configuration Service instance defined for the role.

The login URLs used to specify and initiate this role-based authentication are:

```
http://server_name.domain_name:port/amserver/UI/Login?role=role_name
```

```
http://server_name.domain_name:port/amserver/UI/Login?org=org_name&role=role_name
```

If the org Parameter is not configured, the organization to which the role belongs is determined from the server host and domain specified in the login URL itself.

Role-based Authentication Redirection URLs

Upon a successful or failed role-based authentication, Access Manager looks for information on where to redirect the user. Following is the order of precedence in which the application will look for this information.

Successful Role-based Authentication Redirection URLs

The redirection URL for successful role-based authentication is determined by checking the following places in the following order:

1. A URL set by the authentication module.
2. A URL set by a `goto` Login URL parameter.
3. A URL set in the `clientType` custom files for the `iplanet-am-user-success-url` attribute of the user's profile (`amUser.xml`).
4. A URL set in the `clientType` custom files for the `iplanet-am-auth-login-success-url` attribute of the role to which the user has authenticated.
5. A URL set in the `clientType` custom files for the `iplanet-am-auth-login-success-url` attribute of another role entry of the authenticated user. (This option is a fallback if the previous redirection URL fails.)
6. A URL set in the `clientType` custom files for the `iplanet-am-auth-login-success-url` attribute of the user's organization entry.
7. A URL set in the `clientType` custom files for the `iplanet-am-auth-login-success-url` attribute as a global default.
8. A URL set in the `iplanet-am-user-success-url` attribute of the user's profile (`amUser.xml`).
9. A URL set in the `iplanet-am-auth-login-success-url` attribute of the role to which the user has authenticated.
10. A URL set in the `iplanet-am-auth-login-success-url` attribute of another role entry of the authenticated user. (This option is a fallback if the previous redirection URL fails.)
11. A URL set in the `iplanet-am-auth-login-success-url` attribute of the user's organization entry.
12. A URL set in the `iplanet-am-auth-login-success-url` attribute as a global default.

Failed Role-based Authentication Redirection URLs

The redirection URL for failed role-based authentication is determined by checking the following places in the following order:

1. A URL set by the authentication module.
2. A URL set by a goto Login URL parameter.
3. A URL set in the clientType custom files for the iplanet-am-user-failure-url attribute of the user's profile (amUser.xml).
4. A URL set in the clientType custom files for the iplanet-am-auth-login-failure-url attribute of the role to which the user has authenticated.
5. A URL set in the clientType custom files for the iplanet-am-auth-login-failure-url attribute of another role entry of the authenticated user. (This option is a fallback if the previous redirection URL fails.)
6. A URL set in the clientType custom files for the iplanet-am-auth-login-failure-url attribute of the user's organization entry.
7. A URL set in the clientType custom files for the iplanet-am-auth-login-failure-url attribute as a global default.
8. A URL set in the iplanet-am-user-failure-url attribute of the user's profile (amUser.xml).
9. A URL set in the iplanet-am-auth-login-failure-url attribute of the role to which the user has authenticated.
10. A URL set in the iplanet-am-auth-login-failure-url attribute of another role entry of the authenticated user. (This option is a fallback if the previous redirection URL fails.)
11. A URL set in the iplanet-am-auth-login-failure-url attribute of the user's organization entry.
12. A URL set in the iplanet-am-auth-login-failure-url attribute as a global default.

To Configure Role-Based Authentication

Authentication modules are set for roles after adding the Authentication Configuration service at the role level.

1. Navigate to the organization for which you will configure the authentication attributes.

2. Choose Roles from the View menu.
3. Select the role for which to set the authentication configuration and click on the Properties arrow.

   The role's properties are displayed in the Data pane.
4. Select Services from the View menu in the Data pane.
5. Modify the Authentication Configuration attributes as necessary. An explanation of these attributes can be found in Chapter 34, "Authentication Configuration Service Attributes", or by clicking the Help link in the upper right corner of the console.
6. Click Save.

.

NOTE If you are creating a new role, the Authentication Configuration service is not automatically assigned to it. Make sure that you select the Authentication Configuration service option at the top of the role profile page before you create it.

When role-based auth is enabled, the LDAP authentication module can be left as the default, as there is no need to configure Membership.

Service-based Authentication

This method of authentication allows a user to authenticate to a specific service or application registered to an organization or sub-organization. The service is configured as a Service Instance within the Authentication Configuration Service and is associated with an Instance Name. For authentication to be successful, the user must authenticate to each module defined in the Authentication Configuration service instance configured for the service. For each instance of service-based authentication, the following attributes can be specified:

Authentication Configuration. This defines the authentication modules configured for the service's authentication process.

Login Success URL. This defines the URL to which a user is redirected on successful authentication.

Login Failed URL. This defines the URL to which a user is redirected on failed authentication.

Authentication Post Processing Classes. This defines the post-authentication interface.

Service-based Authentication Login URLs

Service-based authentication can be specified in the User Interface Login URL by defining a service Parameter. After calling the service, the authentication module(s) to which the user will authenticate are retrieved from the Authentication Configuration service instance defined for the service.

The login URLs used to specify and initiate this service-based authentication are:

`http://server_name.domain_name:port/amserver/UI/Login?service=service_name`

and

`http://server_name.domain_name:port/amserver/UI/Login?org=org_name&service=service_name`

If there is no configured org parameter, the organization will be determined from the server host and domain specified in the login URL itself.

Service-based Authentication Redirection URLs

Upon a successful or failed service-based authentication, Access Manager looks for information on where to redirect the user. Following is the order of precedence in which the application will look for this information.

Successful Service-based Authentication Redirection URLs

The redirection URL for successful service-based authentication is determined by checking the following places in the following order:

1. A URL set by the authentication module.
2. A URL set by a goto Login URL parameter.
3. A URL set in the `clientType` custom files for the `iplanet-am-user-success-url` attribute of the user's profile (`amUser.xml`).
4. A URL set in the `clientType` custom files for the `iplanet-am-auth-login-success-url` attribute of the service to which the user has authenticated.
5. A URL set in the `clientType` custom files for the `iplanet-am-auth-login-success-url` attribute of the user's role entry.
6. A URL set in the `clientType` custom files for the `iplanet-am-auth-login-success-url` attribute of the user's organization entry.
7. A URL set in the `clientType` custom files for the `iplanet-am-auth-login-success-url` attribute as a global default.

8. A URL set in the `iplanet-am-user-success-url` attribute of the user's profile (`amUser.xml`).
9. A URL set in the `iplanet-am-auth-login-success-url` attribute of the service to which the user has authenticated.
10. A URL set in the `iplanet-am-auth-login-success-url` attribute of the user's role entry.
11. A URL set in the `iplanet-am-auth-login-success-url` attribute of the user's organization entry.
12. A URL set in the `iplanet-am-auth-login-success-url` attribute as a global default.

Failed Service-based Authentication Redirection URLs

The redirection URL for failed service-based authentication is determined by checking the following places in the following order:

1. A URL set by the authentication module.
2. A URL set by a goto Login URL parameter.
3. A URL set in the `clientType` custom files for the iplanet-am-user-failure-url attribute of the user's profile (`amUser.xml`).
4. A URL set in the `clientType` custom files for the `iplanet-am-auth-login-failure-url` attribute of the service to which the user has authenticated.
5. A URL set in the `clientType` custom files for the `iplanet-am-auth-login-failure-url` attribute of the user's role entry.
6. A URL set in the `clientType` custom files for the `iplanet-am-auth-login-failure-url` attribute of the user's organization entry.
7. A URL set in the `clientType` custom files for the `iplanet-am-auth-login-failure-url` attribute as a global default.
8. A URL set in the `iplanet-am-user-failure-url` attribute of the user's profile (`amUser.xml`).
9. A URL set in the `iplanet-am-auth-login-failure-url` attribute of the service to which the user has authenticated.
10. A URL set in the `iplanet-am-auth-login-failure-url` attribute of the user's role entry.

11. A URL set in the `iplanet-am-auth-login-failure-url` attribute of the user's organization entry.
12. A URL set in the `iplanet-am-auth-login-failure-url` attribute as a global default.

To Configure Service-Based Authentication

Authentication modules are set for services after adding the Authentication Configuration service. To do so:

1. Choose Services from the View menu in the Identity Management module.

   The list of added services are displayed. If the Authentication Configuration service is not added, continue with the steps below. If the service is added, skip to step 4.

2. Click Add in the Navigation pane.

   A list of available services is displayed in the Data pane.

3. Select the checkbox for Authentication Configuration and click Add.

   The Authentication Configuration service will appear in the Navigation pane assuring the administrator that it has been added.

4. Click the Authentication Configuration Properties arrow.

   The Service Instance List is displayed in the in the Data pane.

5. Click on the service instance for which to configure the authentication modules.
6. Modify the authentication configuration attributes and click Save. An explanation of these attributes can be found in Chapter 34, "Authentication Configuration Service Attributes", or by clicking the Help link in the upper right corner of the console.

User-based Authentication

This method of authentication allows a user to authenticate to an authentication process configured specifically for them. The process is configured as a value of the User Authentication Configuration attribute in the user's profile. For authentication to be successful, the user must authenticate to each module defined.

User-based Authentication Login URLs

User-based authentication can be specified in the User Interface Login URL by defining a user Parameter. After calling the correct user, the authentication module(s) to which the user will authenticate are retrieved from the User Authentication Configuration instance defined for them.

The login URLs used to specify and initiate this role-based authentication are:

`http://server_name.domain_name:port/amserver/UI/Login?user=user_name`

`http://server_name.domain_name:port/amserver/UI/Login?org=org_name&user=user_name`

If there is no configured org Parameter, the organization to which the role belongs will be determined from the server host and domain specified in the login URL itself.

User Alias List Attribute

On receiving a request for user-based authentication, the Authentication service first verifies that the user is a valid user and then retrieves the Authentication Configuration data for them. In the case where there is more then one valid user profile associated with the value of the user Login URL parameter, all profiles must map to the specified user. The User Alias Attribute (`iplanet-am-user-alias-list`) in the User profile is where other profiles belonging to the user can be defined. If mapping fails, the user is denied a valid session. The exception would be if one of the users is a top-level admin whereby the user mapping validation is not done and the user is given Super Admin rights.

User-based Authentication Redirection URLs

Upon a successful or failed user-based authentication, Access Manager looks for information on where to redirect the user. Following is the order of precedence in which the application will look for this information.

Successful User-based Authentication Redirection URLs

The redirection URL for successful user-based authentication is determined by checking the following places in order of precedence:

1. A URL set by the authentication module.
2. A URL set by a goto Login URL parameter.
3. A URL set in the `clientType` custom files for the `iplanet-am-user-success-url` attribute of the user's profile (`amUser.xml`).

4. A URL set in the clientType custom files for the iplanet-am-auth-login-success-url attribute of the user's role entry.
5. A URL set in the clientType custom files for the iplanet-am-auth-login-success-url attribute of the user's organization entry.
6. A URL set in the clientType custom files for the iplanet-am-auth-login-success-url attribute as a global default.
7. A URL set in the iplanet-am-user-success-url attribute of the user's profile (amUser.xml).
8. A URL set in the iplanet-am-auth-login-success-url attribute of the user's role entry.
9. A URL set in the iplanet-am-auth-login-success-url attribute of the user's organization entry.
10. A URL set in the iplanet-am-auth-login-success-url attribute as a global default.

Failed User-based Authentication Redirection URLs

The redirection URL for failed user-based authentication is determined by checking the following places in the following order:

1. A URL set by the authentication module.
2. A URL set by a gotoOnFail Login URL parameter.
3. A URL set in the clientType custom files for the iplanet-am-user-failure-url attribute of the user's entry (amUser.xml).
4. A URL set in the clientType custom files for the iplanet-am-auth-login-failure-url attribute of the user's role entry.
5. A URL set in the clientType custom files for the iplanet-am-auth-login-failure-url attribute of the user's organization entry.
6. A URL set in the clientType custom files for the iplanet-am-auth-login-failure-url attribute as a global default.
7. A URL set for the iplanet-am-user-failure-url attribute in the user's entry (amUser.xml).
8. A URL set for the iplanet-am-auth-login-failure-url attribute of the user's role entry.

9. A URL set for the `iplanet-am-auth-login-failure-url` attribute of the user's organization entry.
10. A URL set for the `iplanet-am-auth-login-failure-url` attribute as the global default.

To Configure User-Based Authentication

1. Choose Users from the View menu in the Identity Management module.

   The list of users is displayed in the Navigation pane.

2. Select the user you wish to modify and click the Properties arrow.

   The User Profile is displayed in the data pane.

NOTE If you are creating a new user, the Authentication Configuration service is not automatically assigned to the user. Make sure that you select the Authentication Configuration service option at the top of the User Profile page before you create the user. If this option is not selected, the user will not inherit the authentication configuration defined at for the role.

3. To ensure that the Authentication Configuration service is assigned to the user, Select Services from the View menu. If assigned, the Authentication Configuration service will be listed as an assigned service.
4. Select User from the View menu in the Data pane.
5. Click on the Edit link next to the User Authentication Configuration attribute to define the authentication modules for the user.
6. Click Save.

Authentication Level-based Authentication

Each authentication module can be associated with an integer value for its *authentication level.* Authentication levels can be assigned by clicking the authentication module's Properties arrow in Service Configuration, and changing the corresponding value for the module's Authentication Level attribute. Higher authentication levels define a higher level of trust for the user once that user has authenticated to one or more authentication modules.

The authentication level will be set on a user's SSO token after the user has successfully authenticated to the module. If the user is required to authenticate to multiple authentication modules, and does so successfully, the highest authentication level value will be set in user's SSO token.

If a user attempts to access a service, the service can determine if the user is allowed access by checking the authentication level in user's SSO token. It then redirects the user to go through the authentication modules with a set authentication level.

Users can also access authentication modules with specific authentication level. For example, a user performs a login with the following syntax:

`http://hostname:port/deploy_URI/UI/Login?authlevel=auth_level_value`

All modules whose authentication level is larger or equal to *auth_level_value* will be displayed as an authentication menu for the user to choose. If only one matching module is found, then the login page for that authentication module will be directly displayed.

This method of authentication allows an administrator to specify the security level of the modules to which identities can authenticate. Each authentication module has a separate Authentication Level attribute and the value of this attribute can be defined as any valid integer. With Authentication Level-based authentication, the Authentication Service displays a module login page with a menu containing the authentication modules that have authentication levels equal to or greater then the value specified in the Login URL parameter. Users can select a module from the presented list. Once the user selects a module, the remaining process is based on Module-based Authentication.

Authentication Level-based Authentication Login URLs

Authentication level-based authentication can be specified in the User Interface Login URL by defining the authlevel Parameter. After calling the login screen with the relevant list of modules, the user must choose one with which to authenticate. The login URLs used to specify and initiate authentication level-based authentication are:

```
http://server_name.domain_name:port/amserver/UI/Login?authlevel=authentication_level
```

and

```
http://server_name.domain_name:port/amserver/UI/Login?org=org_name&authlevel=authentication_level
```

If there is no configured org parameter, the organization to which the user belongs will be determined from the server host and domain specified in the login URL itself.

Authentication Level-based Authentication Redirection URLs

Upon a successful or failed authentication level-based authentication, Access Manager looks for information on where to redirect the user. Following is the order of precedence in which the application will look for this information.

Successful Authentication Level-based Authentication Redirection URLs

The redirection URL for successful authentication level-based authentication is determined by checking the following places in order of precedence:

1. A URL set by the authentication module.
2. A URL set by a `goto` Login URL parameter.
3. A URL set in the `clientType` custom files for the `iplanet-am-user-success-url` attribute of the user's profile (`amUser.xml`).
4. A URL set in the `clientType` custom files for the `iplanet-am-auth-login-success-url` attribute of the user's role entry.
5. A URL set in the `clientType` custom files for the `iplanet-am-auth-login-success-url` attribute of the user's organization entry.
6. A URL set in the clientType custom files for the `iplanet-am-auth-login-success-url` attribute as a global default.
7. A URL set in the `iplanet-am-user-success-url` attribute of the user's profile (`amUser.xml`).
8. A URL set in the `iplanet-am-auth-login-success-url` attribute of the user's role entry.
9. A URL set in the `iplanet-am-auth-login-success-url` attribute of the user's organization entry.
10. A URL set in the `iplanet-am-auth-login-success-url` attribute as a global default.

Failed Authentication Level-based Authentication Redirection URLs

The redirection URL for failed authentication level-based authentication is determined by checking the following places in the following order:

1. A URL set by the authentication module.

2. A URL set by a `gotoOnFail` Login URL parameter.
3. A URL set in the `clientType` custom files for the `iplanet-am-user-failure-url` attribute of the user's entry (`amUser.xml`).
4. A URL set in the `clientType` custom files for the `iplanet-am-auth-login-failure-url` attribute of the user's role entry.
5. A URL set in the `clientType` custom files for the `iplanet-am-auth-login-failure-url` attribute of the user's organization entry.
6. A URL set in the `clientType` custom files for the `iplanet-am-auth-login-failure-url` attribute as a global default.
7. A URL set for the `iplanet-am-user-failure-url` attribute in the user's entry (`amUser.xml`).
8. A URL set for the `iplanet-am-auth-login-failure-url` attribute of the user's role entry.
9. A URL set for the `iplanet-am-auth-login-failure-url` attribute of the user's organization entry.
10. A URL set for the `iplanet-am-auth-login-failure-url` attribute as the global default.

Module Based Authentication

Users can access a specific authentication module using the following syntax:

`http://`*`hostname:port/deploy_URI`*`/UI/Login?module=`*`module_name`*

Before the authentication module can be accessed, the Core authentication service attribute Organization Authentication Modules must be modified to include the authentication module name. If the authentication module name is not included in this attribute, the "authentication module denied" page will be displayed when the user attempts to authenticate.

This method of authentication allows a user to specify the module to which they will authenticate. The specified module must be registered to the organization or sub-organization that the user is accessing. This is configured in the Organization Authentication Modules attribute of the organization's Core Authentication Service. On receiving this request for module-based authentication, the Authentication Service verifies that the module is correctly configured as noted, and if the module is not defined, the user is denied access.

NOTE See Chapter 8, "Authentication Options" for more information on how to register the authentication modules using the Access Manager console.

Module-based Authentication Login URLs

Module-based authentication can be specified in the User Interface Login URL by defining a module Parameter. The login URLs used to specify and initiate module-based authentication are:

```
http://server_name.domain_name:port/amserver/UI/Login?module=authentication_module_name
```

```
http://server_name.domain_name:port/amserver/UI/Login?org=org_name&module=authentication_module_name
```

If there is no configured org parameter, the organization to which the user belongs will be determined from the server host and domain specified in the login URL itself.

Module-based Authentication Redirection URLs

Upon a successful or failed module-based authentication, Access Manager looks for information on where to redirect the user. Following is the order of precedence in which the application will look for this information.

Successful Module-based Authentication Redirection URLs

The redirection URL for successful module-based authentication is determined by checking the following places in order of precedence:

1. A URL set by the authentication module.
2. A URL set by a goto Login URL parameter.
3. A URL set in the `clientType` custom files for the iplanet-am-user-success-url attribute of the user's profile (`amUser.xml`).
4. A URL set in the `clientType` custom files for the `iplanet-am-auth-login-success-url` attribute of the user's role entry.
5. A URL set in the `clientType` custom files for the `iplanet-am-auth-login-success-url` attribute of the user's organization entry.
6. A URL set in the `clientType` custom files for the `iplanet-am-auth-login-success-url` attribute as a global default.
7. A URL set in the `iplanet-am-user-success-url` attribute of the user's profile (`amUser.xml`).

8. A URL set in the iplanet-am-auth-login-success-url attribute of the user's role entry.
9. A URL set in the iplanet-am-auth-login-success-url attribute of the user's organization entry.
10. A URL set in the iplanet-am-auth-login-success-url attribute as a global default.

Failed Module-based Authentication Redirection URLs

The redirection URL for failed module-based authentication is determined by checking the following places in the following order:

1. A URL set by the authentication module.
2. A URL set by a gotoOnFail Login URL parameter.
3. A URL set in the clientType custom files for the iplanet-am-user-failure-url attribute of the user's entry (amUser.xml).
4. A URL set in the clientType custom files for the iplanet-am-auth-login-failure-url attribute of the user's role entry.
5. A URL set in the clientType custom files for the iplanet-am-auth-login-failure-url attribute of the user's organization entry.
6. A URL set in the clientType custom files for the iplanet-am-auth-login-failure-url attribute as a global default.
7. A URL set for the iplanet-am-user-failure-url attribute in the user's entry (amUser.xml).
8. A URL set for the iplanet-am-auth-login-failure-url attribute of the user's role entry.
9. A URL set for the iplanet-am-auth-login-failure-url attribute of the user's organization entry.
10. A URL set for the iplanet-am-auth-login-failure-url attribute as the global default.

Authentication Configuration

The Authentication Configuration service is used to define authentication modules for any of the following authentication types:

- organization

- role
- service
- user

Once an authentication module is defined for one of these authentication types, the module can be configured to supply redirect URLs, as well as a post-processing Java class specification, based on a successful or failed authentication process.

Before an authentication module can be configured, the Core authentication service attribute Organization Authentication Modules must be modified to include the specific authentication module name.

Authentication Configuration User Interface

The Authentication Configuration services allows you to define one or more authentication services (or *modules*) that a user must pass before being allowed access to the console or any secured resource within Access Manager. Organization, role, service, and user-based authentication use a common user interface to define the authentication modules. (Instructions for access the Authentication Configuration interface for specific object types are described in subsequent sections).

1. Click on the Edit link next to the object's Authentication Configuration attribute to display the Module List window.
2. This window lists the authentication modules that have been assigned to the object. If no modules exist, click Add to display the Add Module window.

   The Add Module Window contains three files to define:

   Module Name. This pull-down list allows you to select the authentication modules (including custom modules that may be added) that have been enabled in the Organization Authentication Modules attribute of the Core Authentication module.

   Flag. This pull-down menu allows you specify the authentication module requirements. It can be one of:

   - REQUIRED - The authentication module is required to succeed. If it succeeds or fails, authentication continues to proceed down the authentication module list.

- REQUISITE - The authentication module is required to succeed. If it succeeds, authentication continues down the authentication module list. If it fails, control returns to the application (authentication does not proceed down the authentication module list.)
- SUFFICIENT - The authentication module is not required to succeed. If it does succeed, control immediately returns to the application (authentication does not proceed down the authentication module list.). If it fails, authentication continues down the list.
- OPTIONAL - The authentication module is not required to succeed. If it succeeds or fails, authentication still continues to proceed down the list.

These flags establish an enforcement criteria for the authentication module for which they are defined. There is hierarchy for enforcement, with REQUIRED being the highest, and OPTION being the lowest.

For example, if an administrator defines an LDAP module with the REQUIRED flag, then the user's credential must pass the LDAP authentication requirements to access a given resource.

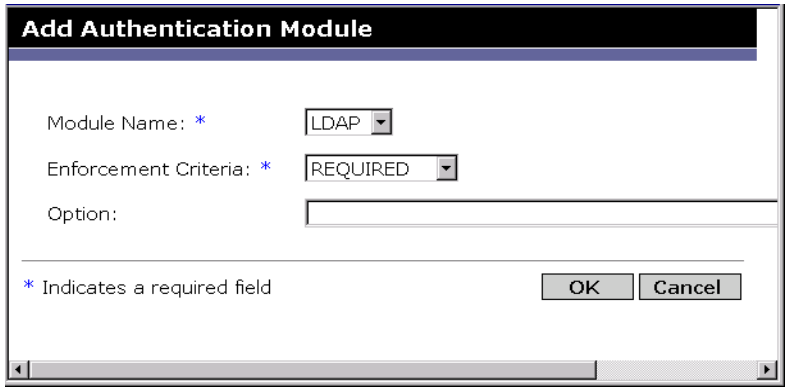
If you add multiple authentication modules and for each module the Flag is set to REQUIRED, the user must pass all authentication requirements before being granted access.

For more information on the flag definitions, refer to the JAAS (Java Authentication and Authorization Service) located at:

`http://java.sun.com/security/jaas/doc/module.html`

Option. Allows for additional options for the for the module as a key=value pair. Multiple options are separated by a space.

Figure 7-1 Add Module List Window For A User



3. Once the fields are selected, click OK to return to the Module List window. The authentication modules you have defined are listed in this window. Click Save.

   You can add as many authentication modules to this list as you wish. Adding multiple authentication modules is called *authentication chaining*. If you are chaining authentication modules, note that the order in which they are listed defines the order of hierarchy of enforcement. For more information on authentication chaining, see "Authentication Module Chaining" on page 176.

   To change the order of the authentication modules:

   a. Click the Reorder button.

   b. Select the module you wish to reorder.

   c. Use the Up and Down buttons to place it in the desired position.

4. To remove any authentication module from the list, select the checkbox next to the authentication module and click Delete.

NOTE If you enter `amadmin` credentials in any of the modules in a chain, you will receive the amadmin profile. Authentication does not check for alias mapping in this case, nor does it check for modules in the chain.

Authentication Module Chaining

One or more authentication modules can be configured so a user must pass authentication credentials to all of them. This is referred to as *authentication chaining*. Authentication chaining in Access Manager is achieved using the JAAS framework integrated in the Authentication Service. Module chaining is configured under the Authentication Configuration service. Each registered module is assigned one of the following four values:

- Required
- Requisite
- Sufficient
- Optional

Once authentication to the modules, as defined by the flags, in the chain is successful, control is returned to the Authentication Service (from the JAAS framework) which validates all the user IDs used to authenticate and maps them to one user. The mapping is achieved by configuring the User Alias List attribute in the user's profile. A valid session token is issued to the user if all the maps are correct; if not, the user is denied a valid session token. The following properties would represent the single authenticated user to which the other users are aliased:

- Principal (would contain the DN of the user in the case that the user has one)
- UserToken
- UserId

With Dynamic Profile creation enabled if all user IDs do not map to the same user and if one of the user IDs exists in the local directory server then other user IDs will be added to the user alias list attribute of the existing user.

NOTE

- In authentication chaining, if all user IDs do not map to one single user, the failure redirection URL will be picked up from the last failed authentication module or none if all individual modules succeed (with different user ID). If case of user-based authentication, no matter what user ID is given in the authentication page, the failure redirection URL will always be picked up from the user parameter in the login URL.
- With Dynamic Profile creation enabled, if all user ids do not map to the same use, and if one of the user ids exists in the local directory server, then additional user ids will be added to the existing user's user alias list attribute.

Authentication Configuration for Organizations

Authentication modules are set for an organization by first adding the Core Authentication service to the organization.

To configure the organization's authentication attributes:

1. Navigate to the organization for which you will configure the authentication attributes.
2. Select Services from the View menu.
3. Click the Core Properties arrow in the service listing.

   The Core authentication attributes are displayed in the Data pane.
4. Click the edit link next to the Admin Authenticator attribute. This allows you to define the authentication services for administrators only. This attribute can be used if the authentication module for administrators needs to be different from the module for end users. The default authentication module is LDAP.

   Once you have defined the authentication services, click Save to save the changes, and click Close to return to the Core Authentication attributes for organizations.
5. Click the Edit link next to the Organization Authentication Configuration attribute. This allows you to define authentication modules for all users within the organization. The default authentication module is LDAP.
6. Once you have defined the authentication services, click Save to save the changes, and click Close to return to the Core Authentication attributes for organizations.

Authentication Configuration for Roles

Authentication modules are set for roles after adding the Authentication Configuration service at the role level.

1. Navigate to the organization for which you will configure the authentication attributes.
2. Choose Roles from the View menu.
3. Select the role for which to set the authentication configuration and click on the Properties arrow.

   The role's properties are displayed in the Data pane.

4. Select Services from the View menu in the Data pane.
5. Modify the Authentication Configuration attributes as necessary. An explanation of these attributes can be found inChapter 34, “Authentication Configuration Service Attributes” or by clicking the Help link in the upper right corner of the console.
6. Click Save.

.

NOTE If you are creating a new role, the Authentication Configuration service is not automatically assigned to it. Make sure that you select the Authentication Configuration service option at the top of the role profile page before you create it.

When role-based auth is enabled, the LDAP authentication module can be left as the default, as there is no need to configure Membership.

Authentication Configuration for Services

Authentication modules are set for services after adding the Authentication Configuration service. To do so:

1. Choose Services from the View menu in the Identity Management module.

   The list of added services are displayed. If the Authentication Configuration service is not added, continue with the steps below. If the service is added, skip to Step 4.
2. Click Add in the Navigation pane.

   A list of available services is displayed in the Data pane.
3. Select the checkbox for Authentication Configuration and click Add.

   The Authentication Configuration service will appear in the Navigation pane assuring the administrator that it has been added.
4. Click the Authentication Configuration Properties arrow.

   The Service Instance List is displayed in the in the Data pane.
5. Click on the service instance for which to configure the authentication modules.
6. Modify the authentication configuration attributes and click Save. An explanation of these attributes can be found in Chapter 34, “Authentication Configuration Service Attributes” or by clicking the Help link in the upper right corner of the console.

Authentication Configuration for Users

1. Choose Users from the View menu in the Identity Management module.

   The list of users is displayed in the Navigation pane.

2. Select the user you wish to modify and click the Properties arrow.

   The User Profile is displayed in the data pane.

.

NOTE If you are creating a new user, the Authentication Configuration service is not automatically assigned to the user. Make sure that you select the Authentication Configuration service option at the top of the User Profile page before you create the user. If this option is not selected, the user will not inherit the authentication configuration defined at for the role.

3. To ensure that the Authentication Configuration service is assigned to the user, Select Services from the View menu. If assigned, the Authentication Configuration service will be listed as an assigned service.
4. Select General from the View menu in the Data pane.
5. Click on the Edit link next to the User Authentication Configuration attribute to define the authentication modules for the user.
6. Click Save.

Account Locking

The Authentication Service provides a feature where a user will be *locked out* from authenticating after *n* failures. This feature is turned off by default, but can be enabled using the Access Manager console.

NOTE Only modules that throw an Invalid Password Exception can leverage the Account Locking feature.

The Core Authentication service contains attributes for enabling and customizing this feature including (but not limited to):

- **Login Failure Lockout Mode** which enables account locking.

- **Login Failure Lockout Count** which defines the number of tries that a user may attempt to authenticate before being locked out. This count is valid per user ID only; the same user ID needs to fail for the given count after which that user ID would be locked out.
- **Login Failure Lockout Interval** defines (in minutes) the amount of time in which the Login Failure Lockout Count value must be completed before a user is locked out.
- **Email Address to Send Lockout Notification** specifies an email address to which user lockout notifications will be sent.
- **Warn User After N Failure** specifies the number of authentication failures that can occur before a warning message will be displayed to the user. This allows an administrator to set additional login attempts after the user is warned about an impending lockout.
- **Login Failure Lockout Duration** defines (in minutes) how long the user will have to wait before attempting to authenticate again after lockout.
- **Lockout Attribute Name** defines which LDAP attribute in the user's profile will be set to inactive for Physical Locking.
- **Lockout Attribute Value** defines to what the LDAP attribute specified in **Lockout Attribute Name** will be set: inactive or active.

Email notifications are sent to administrators regarding any account lockouts. (Account locking activities are also logged.) For more information on the account locking attributes, see Chapter 21, "Core Authentication Attributes".

NOTE For special instructions when using this feature on a Microsoft® Windows 2000 operating system, see "Simple Mail Transfer Protocol (SMTP)" in Appendix A, "AMConfig.properties File," of the *Access Manager Developer's Guide*

Access Manager supports two types of account locking are supported: Physical Locking and Memory Locking, defined in the following sections.

Physical Locking

This is the default locking behavior for Access Manager. The locking is initiated by changing the status of a LDAP attribute in the user's profile to inactive. The Lockout Attribute Name attribute defines the LDAP attribute used for locking purposes. See the *Sun Java System Access Manager Administration Guide* for more information on configuring physical locking.

NOTE An aliased user is one that is mapped to an existing LDAP user profile by configuring the User Alias List Attribute (`iplanet-am-user-alias-list` in `amUser.xml`) in the LDAP profile. Aliased users can be verified by adding iplanet-am-user-alias-list to the Alias Search Attribute Name field in the Core Authentication Service. That said, if an aliased user is locked out, the actual LDAP profile to which the user is aliased will be locked. This pertains only to physical lockout with authentication modules other than LDAP and Membership.

Memory Locking

Memory locking is enabled by changing the Login Failure Lockout Duration attribute to a value greater then 0. The user's account is then locked in memory for the number of minutes specified. The account will be unlocked after the time period has passed. Following are some special considerations when using the memory locking feature:

- If Access Manager is restarted, all accounts locked in memory are unlocked.
- If a user's account is locked in memory and the administrator changes the account locking mechanism to physical locking (by setting the lockout duration back to 0), the user's account will be unlocked in memory and the lock count reset.
- After memory lockout, when using authentication modules other than LDAP and Membership, if the user attempts to login with the correct password, a *User does not have profile in this organization error.* is returned rather than a *User is not active.* error.

NOTE If the Failure URL attribute is set in the user's profile, neither the lockout warning message nor the message indicating that their account has been locked will not be displayed; the user will be redirected to the defined URL.

Authentication Service Failover

Authentication service failover automatically redirects an authentication request request to a secondary server if the primary server fails because of a hardware or software problem or if the server is temporarily shut down.

An authentication context must first be created on an instance of Access Manager where the authentication service is available. If this instance of Access Manager is not available, an authentication context can then be created on a different instance of Access Manager through the authentication failover mechanism. The authentication context will check for server availability in the following order:

1. The authentication service URL is passed to the AuthContext API. For example:

   ```
   AuthContext(orgName, url)
   ```

   If this API is used, it will only use the server referenced by the URL. No failover will occur even if the authentication service is available on that server.

2. The authentication context will check the server defined in the `com.iplanet.am.server*` attribute of the `AMConfig.properties` file.

3. If step 2 fails, then the authentication context queries the platform list from a server where the Naming service is available This platform list is automatically created when multiple instances of Access Manager are installed (generally, for failover purposes) sharing a one instance of Directory Server.

   For example, if the platform list contains URLs for `Server1`, `Server2` and `Server3`, then the authentication context will loop through `Server1`, `Server2` and `Server3` until authentication succeeds on one of them.

The platform list may not always be obtained from the same server, as it depends on the availability of the Naming service. Furthermore, Naming service failover may occur first. Multiple Naming service URLs are specified in the `com.iplanet.am.naming.url property` (in `AMConfing.properties`). The first available Naming service URL will be used to identify the server, which will contain the list of servers (in its platform server list) on which authentication failover will occur.

Fully Qualified Domain Name Mapping

Fully Qualified Domain Name (FQDN) mapping enables the Authentication Service to take corrective action in the case where a user may have typed in an incorrect URL (such as specifying a partial host name or IP address to access protected resources). FQDN mapping is enabled by modifying the com.sun.identity.server.fqdnMap attribute in the AMConfig.properties file. The format for specifying this property is:

com.sun.identity.server.fqdnMap[*invalid-name*]=*valid-name*

The value *invalid-name* would be a possible invalid FQDN host name that may be typed by the user, and *valid-name* would be the actual host name to which the filter will redirect the user. Any number of mappings can be specified (as illustrated in Code Example 1-1) as long as they conform to the stated requirements. If this property is not set, the user would be sent to the default server name configured in the com.iplanet.am.server.host=*server_name* property also found in the AMConfig.properties file.

:

Code Example 7-1 FQDN Mapping Attribute In AMConfig.properties

```
com.sun.identity.server.fqdnMap[isserver]=isserver.mydomain.com
com.sun.identity.server.fqdnMap[isserver.mydomain]=isserver.mydomain.com
com.sun.identity.server.fqdnMap[IP address]=isserver.mydomain.com
```

Possible Uses For FQDN Mapping

This property can be used for creating a mapping for more than one host name which may be the case if applications hosted on a server are accessible by more than one host name. This property can also be used to configure Access Manager to not take corrective action for certain URLs. For example, if no redirect is required for users who access applications by using an IP address, this feature can be implemented by specifying a map entry such as:

com.sun.identity.server.fqdnMap[*IP address*]=*IP address*.

CAUTION If more than one mapping is defined, ensure that there are no overlapping values in the invalid FQDN name. Failing to do so may result in the application becoming inaccessible.

Persistent Cookie

A persistent cookie is one that continues to exist after the web browser is closed, allowing a user to login with a new browser session without having to reauthenticate. The name of the cookie is defined by the com.iplanet.am.pcookie.name property in AMConfig.properties; the default value is DProPCookie. The cookie value is a 3DES-encrypted string containing the userDN, organization name, authentication module name, maximum session time, idle time, and cache time. To enable persistent cookies:

1. Turn on the Persistent Cookie Mode in the Core Authentication module.

2. Configure a time value for the Persistent Cookie Maximum Time attribute in the Core Authentication module.
3. Append the iPSPCookie Parameter with a value of yes to the User Interface Login URL.

   Once the user authenticates using this URL, if the browser is closed, they can open a new browser window and will be redirected to the console without reauthentication. This will work until the time defined in Step 2 elapses.

Persistent Cookie Mode can be turned on using the Authentication SPI method:

AMLoginModule.setPersistentCookieOn().

Multi-LDAP Authentication Module Configuration

As a form of failover or to configure multiple values for an attribute when the Access Manager console only provides one value field, an administrator can define multiple LDAP authentication module configurations under one organization. Although these additional configurations are not visible from the console, they work in conjunction with the primary configuration if an initial search for the requesting user's authorization is not found. For example, one organization can define a search through LDAP servers for authentication in two different domains or it can configure multiple user naming attributes in one domain. For the latter, which has only one text field in the console, if a user is not found using the primary search criteria, the LDAP module will then search using the second scope. Following are the steps to configure additional LDAP configurations.

To Add An Additional LDAP Configuration

1. Write an XML file including the complete set of attributes and new values needed for second (or third) LDAP authentication configuration.

   The available attributes can be referenced by viewing the amAuthLDAP.xml located in `etc/opt/SUNWam/config/xml`. This XML file created in this step though, unlike the `amAuthLDAP`.xml, is based on the structure of the `amadmin`.dtd. Any or all attributes can be defined for this file. Code Example 1-2 is an example of a subconfiguration file that includes values for all attributes available to the LDAP authentication configuration.

Code Example 7-2 Sample XML File To Add An LDAP SubConfiguration

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
  Copyright (c) 2002 Sun Microsystems, Inc. All rights reserved.
  Use is subject to license terms.
-->
<!DOCTYPE Requests
    PUBLIC "-//iPlanet//Sun ONE Identity Server 6.0 Admin CLI DTD//EN"
    "jar://com/iplanet/am/admin/cli/amAdmin.dtd"
>
<!--
  Before adding subConfiguration load the schema with
GlobalConfiguration defined and replace corresponding
 serviceName and subConfigID in this sample file OR load
 serviceConfigurationRequests.xml before loading this sample
-->
<Requests>
<OrganizationRequests DN="dc=iplanet,dc=com">
   <AddSubConfiguration subConfigName = "ssc"
      subConfigId = "serverconfig"
      priority = "0" serviceName="iPlanetAMAuthLDAPService">

      <AttributeValuePair>
         <Attribute name="iplanet-am-auth-ldap-server"/>
         <Value>newvalue</Value>
      </AttributeValuePair>
      <AttributeValuePair>
         <Attribute name="iplanet-am-auth-ldap-server"/>
         <Value>vbrao.red.iplanet.com:389</Value>
      </AttributeValuePair>
      <AttributeValuePair>
         <Attribute name="iplanet-am-auth-ldap-base-dn"/>
         <Value>dc=iplanet,dc=com</Value>
      </AttributeValuePair>
      <AttributeValuePair>
         <Attribute name="planet-am-auth-ldap-bind-dn"/>
         <Value>cn=amldapuser,ou=DSAME Users,dc=iplanet,dc=com</Value>
      </AttributeValuePair>
      <AttributeValuePair>
         <Attribute name="iplanet-am-auth-ldap-bind-passwd"/>
         <Value>plain text password</Value>
      </AttributeValuePair>
      <AttributeValuePair>
         <Attribute name="iplanet-am-auth-ldap-user-naming-attribute"/>
         <Value>uid</Value>
      </AttributeValuePair>
      <AttributeValuePair>
         <Attribute name="iplanet-am-auth-ldap-user-search-attributes"/>
         <Value>uid</Value>
      </AttributeValuePair>
      <AttributeValuePair>
         <Attribute name="iplanet-am-auth-ldap-search-scope"/>
         <Value>SUBTREE</Value>
      </AttributeValuePair>
      <AttributeValuePair>
```

Code Example 7-2 Sample XML File To Add An LDAP SubConfiguration *(Continued)*

```
        <Attribute name="iplanet-am-auth-ldap-ssl-enabled"/>
        <Value>false</Value>
      </AttributeValuePair>
      <AttributeValuePair>
        <Attribute name="iplanet-am-auth-ldap-return-user-dn"/>
        <Value>true</Value>
      </AttributeValuePair>
      <AttributeValuePair>
        <Attribute name="iplanet-am-auth-ldap-auth-level"/>
        <Value>0</Value>
      </AttributeValuePair>
      <AttributeValuePair>
        <Attribute name="iplanet-am-auth-ldap-server-check"/>
        <Value>15</Value>
      </AttributeValuePair>

    </AddSubConfiguration>

  </OrganizationRequests>
  </Requests>
```

2. Copy the plain text password as the value for the iplanet-am-auth-ldap-bind-passwd in the XML file created in Step 1.

   The value of this attribute is formatted in bold in Code Example 1-2 on page 41.

3. Load the XML file using the amadmin command line tool.

   ```
   ./amadmin -u amadmin -w administrator_password -v -t name_of_XML_file.
   ```

Note that this second LDAP configuration can not be seen or modified using the Access Manager console.

TIP There is a sample available for multi-LDAP configuration. See the serviceAddMultipleLDAPConfigurationRequests.xml command line template in */AcessManager-base*/SUNWam/samples/admin/cli/bulk-ops/. Instructions can be found in Readme.html at */AcessManager-base*/SUNWam/samples/admin/cli/.

Session Upgrade

The Authentication Service allows for the upgrading of a valid session token based on a second, successful authentication performed by the same user to one organization. If a user with a valid session token attempts to authenticate to a resource secured by his current organization and this second authentication request is successful, the session is updated with the new properties based on the new authentication. If the authentication fails, the user's current session is returned without an upgrade. If the user with a valid session attempts to authenticate to a resource secured by a different organization, the user will receive a message asking whether they would like to authenticate to the new organization. The user can, at this point, maintain the current session or attempt to authenticate to the new organization. Successful authentication will result in the old session being destroyed and a new one being created.

During session upgrade, if a login page times out, redirection to the original success URL will occur. Timeout values are determined based on:

- The page timeoutvalue set for each module (default is 1 minute)
- com.iplanet.am.invalidMaxSessionTime property in AMConfig.properties (default is 10 minutes)
- iplanet-am-max-session-time (default is 120 minutes)

The values of com.iplanet.am.invalidMaxSessionTimeout and iplanet-am-max-session-time should be greater than the page timeout value, or the valid session information during session upgrade will be lost and URL redirection to the previous successful URL will fail.

Validation Plug-in Interface

An Administrator can write username or password validation logic suitable to their organization, and plug this into the Authentication Service. (This functionality is supported only by the LDAP and Membership authentication modules.) Before authenticating the user or changing the password, Access Manager will invoke this plugin. If the validation is successful, authentication continues; if it fails, an authentication failed page will be thrown. The plugin extends the com.iplanet.am.sdk.AMUserPasswordValidation class which is part of the Service Management SDK. Information on this SDK can be found in the com.iplanet.am.sdk package in the Access Manager Javadocs. The steps below document how to write and configure a validation plugin for Access Manager.

1. The new plugin class will extend the com.iplanet.am.sdk.`AMUserPasswordValidation` class and implement the `validateUserID()` and `validatePassword()` methods. `AMException` should be thrown if validation fails.
2. Compile the plugin class and place the .class file in the desired location. Update the classpath so that it is accessible by the Access Manager during runtime.
3. Login to the Access Manager console as top-level administrator. Click on the Service Management tab, and get to the attributes for the Administration Service. Type the name of the plugin class (including the package name) in the `UserID & Password Validation Plugin Class` field.
4. Logout and login.

JAAS Shared State

The JAAS shared state provides sharing of both user ID and password between authentication modules. Options are defined for each authentication module for:

- Organization
- User
- Service
- Role

Upon failure, the module prompts for its required credentials. After failed authentication, the module stops running, or the logout shared state clears.

Enabling JAAS Shared State

To configure the JAAS shared state:

- Use the `iplanet-am-auth-sharedstate-enabled` option.
- The usage for the shared state option is: `iplanet-am-auth-shared-state-enabled=true`
- The default for this option is true.

Upon failure, the authentication module will prompt for the required credentials as per the `tryFirstPass` option behavior suggested in the JAAS specification.

JAAS Shared State Store Option

To configure the JAAS shared state store option:

- Use the iplanet-amauth-store-shared-state-enabled option.
- The usage for the store shared state option is:
  `iplanet-am-auth-shared-state-enabled=true`
- The default for this option is false.

After a commit, an abort or a logout, the shared state will be cleared.

Chapter 8

Authentication Options

Sun Java™ System Access Manager 6 2005Q1 provides a framework for authentication, which is a process that verifies the identities of users accessing applications within an enterprise. A user must pass an authentication process before accessing the Access Manager console, or any other Access Manager-protected resource. Authentication is implemented through plug-ins that validate the user's identity. (This plug-in architecture is described more fully in the *Access Manager Developer's Guide*.)

The Access Manager console is used to set the default values, to add authentication modules, to create an authentication template and to enable the associated authentication module. This chapter provides an overview of the authentication modules and instructions for adding them. It contains the following sections:

Core Authentication

Access Manager provides, by default, fifteen different authentication modules, as well as a Core authentication module. The Core authentication module provides overall configuration for the authentication module. Before adding and enabling Active Directory, Anonymous, Certificate-based, HTTP Basic, JDBC, LDAP, any authentication module, the Core authentication must be added and enabled. Both the Core and LDAP Authentication modules are automatically enabled for the default organization. Chapter 21, “Core Authentication Attributes” contains a detailed listing of the Core attributes.

Adding and Enabling the Core Service

1. Go to the organization for which the Core module is to be added.
2. Choose Services from the View menu.
3. Click Add in the Navigation pane.

   A list of available modules displays in the Data pane.
4. Select the checkbox for Core Authentication and click Add.

   The Core Authentication module will appear in the Navigation pane assuring the administrator that it has been added.
5. Click the Core Authentication Properties arrow.

   The message *A template does not currently exist for this service. Do you want to create one now?* appears in the Data pane.
6. Click Create.

   The Core attributes appear in the Data pane. Modify the attributes as necessary. An explanation of the Core attributes can be found in Chapter 21, “Core Authentication Attributes” or by clicking the Help link in the upper right corner of the console.

Active Directory Authentication

The Active Directory authentication module performs authentication in a similar manner to the LDAP Directory Authentication module, but uses Microsoft's Active Directory™ server (as opposed to Directory Server in LDAP authentication module). Although the LDAP authentication module can be configured for an Active Directory server, this module allows you have both LDAP and Active Directory authentication exist under the same organization.

NOTE For this release, the Active Directory authentication module only supports user authentication. Password policy is only supported in the LDAP authentication module.

Adding and Enabling Active Directory Authentication

You must log in to Access Manager as the Organization Administrator or Top-Level Administrator.

1. Go to the organization for which Membership Authentication is to be added.
2. Choose Services from the View menu.

   The Core module, if already added, displays in the Navigation pane. If it is not already added, it can be done concurrently with the Active Directory authentication module.
3. Click Add in the Navigation pane.

   A list of available modules displays in the Data pane.
4. Select the checkbox for Active Directory authentication and click Add.

   The Active Directory authentication module will appear in the Navigation pane assuring the administrator that it has been added.
5. Click the Active Directory Authentication Properties arrow.

   The message *A template does not currently exist for this module. Do you want to create one now?* appears in the Data pane.
6. Click Create.

   The Active Directory authentication attributes appear in the Data pane. Modify the attributes as necessary.

7. Click Save.

   The Active Directory authentication module has been enabled.

Logging In Using Active Directory Authentication

In order to log in using Active Directory authentication, the Core authentication module attribute "Organization Authentication Modules" on page 298 must be modified to enable and select Active Directory authentication. This ensures that when the user logs in using `http://hostname:port/deploy_URI/UI/Login?module=AD`, (note case sensitivity) the user will see the Active Directory authentication login window. Based on the authentication type that is being used (such as service, role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

Anonymous Authentication

By default, when this module is enabled, a user can log in to Access Manager as an *anonymous* user. A list of anonymous users can also be defined for this module by configuring the Valid Anonymous User List attribute. Granting anonymous access means that it can be accessed without providing a password. Anonymous access can be limited to specific types of access (for example, access for read or access for search) or to specific subtrees or individual entries within the directory.

Adding and Enabling Anonymous Authentication

You must log in to Access Manager as the Organization Administrator or Top-Level Administrator.

1. Go to the organization for which Anonymous Authentication is to be added.
2. Choose Services from the View menu.

   The Core module, if already added, displays in the Navigation pane. If it is not already added, it can be done concurrently with the Anonymous Authentication module.
3. Click Add in the Navigation pane.

   A list of available modules displays in the Data pane.

4. Select the checkbox for Anonymous Authentication and click Add.

   The Anonymous Authentication module will appear in the Navigation pane assuring the administrator that it has been added.

5. Click the Anonymous Authentication Properties arrow.

   The message *A template does not currently exist for this service. Do you want to create one now?* appears in the Data pane.

6. Click Create.

   The Anonymous Authentication attributes appear in the Data pane. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 19, "Anonymous Authentication Attributes" or by clicking the Help link in the upper right corner of the console.

7. Click Save.

   The Anonymous Authentication module has been enabled.

Logging In Using Anonymous Authentication

In order to log in using Anonymous Authentication, the Core Authentication module attribute "Organization Authentication Modules" on page 298 must be modified to enable and select Anonymous Authentication. This ensures that when the user logs in using `http(s)://hostname:port/SERVER_DEPLOY_URI//UI/Login?module=Anonymous&org=org_name`. To login without the Anonymous Authentication login window, use the following syntax:

`http(s)://hostname:port/SERVER_DEPLOY_URI//UI/Login?module=Anonymous&org=org_name&Login.Token1=user_id`

Based on the authentication type that is being used (such as service, role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

NOTE The Default Anonymous User Name attribute value in the Anonymous Authentication module is `anonymous`. This is the name users use to log in. A default Anonymous User must be created within the organization. The user id should be identical to the user name specified in the Anonymous Authentication attributes. This can optionally be case sensitive.

Certificate-based Authentication

Certificate-based Authentication involves using a personal digital certificate (PDC) to identify and authenticate a user. A PDC can be configured to require a match against a PDC stored in Directory Server, and verification against a Certificate Revocation List.

There are a number of things that need to be accomplished before adding the Certificate-based Authentication module to an organization. First, the web container that is installed with the Access Manager needs to be secured and configured for Certificate-based Authentication. Before enabling the Certificate-based module, see Chapter 6, "Using Certificates and Keys" in the *Sun ONE Web Server 6.1 Administrator's Guide for* these initial Web Server configuration steps. This document can be found at the following location:

```
http://docs.sun.com/db/prod/s1websrv#hic
```

Or, see the *Sun ONE Application Sever Administrator's Guide to Security* at the following location:

```
http://docs.sun.com/db/prod/s1appsrv#hic
```

NOTE Each user that will authenticate using the certificate-based module must request a PDC for the user's browser. Instructions are different depending upon the browser used. See your browser's documentation for more information.

Adding and Enabling Certificate-based Authentication

You must log in to Access Manager as the Organization Administrator.

1. Go to the organization for which Certificate-based Authentication is to be added.
2. Choose Services from the View menu.

   The Core module, if already added, displays in the Navigation pane. If it is not already added, it can be done concurrently with the Certificate-based Authentication module.
3. Click Add in the Navigation pane.

   A list of available modules displays in the Data pane.

4. Select the checkbox for Certificate-based Authentication and click Add.

   The Certificate-based Authentication module will appear in the Navigation pane assuring the administrator that it has been added.

5. Click the Certificate-based Authentication Properties arrow.

   The message *A template does not currently exist for this service. Do you want to create one now?* appears in the Data pane.

6. Click Create.

   The Certificate-based Authentication attributes appear in the Data pane. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 20, "Certificate Authentication Attributes" or by clicking the Help link in the upper right corner of the console.

7. Click Save.

Adding a Server URL in Platform Server List for Certificate-based Authentication

In order to add this module, you must log in to Access Manager as the Organization Administrator and have Access Manager and the web container configured for SSL and with client authentication enabled. For more information, see "Configuring Access Manager in SSL Mode" on page 63.

Logging In Using Certificate-based Authentication

In order to make certificate-based authentication the default authentication method, the Core Authentication module attribute Organization Authentication Modules (see page 298) must be modified. This ensures that when the user logs in using `https://hostname:port/deploy_URI/UI/Login?module=Cert`, the user will see the Certificate-based Authentication login window. Based on the authentication type that is being used (such as role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

HTTP Basic Authentication

This module uses basic authentication, which is the HTTP protocol's built-in authentication support. The web server issues a client request for username and password, and sends that information back to the server as part of the authorized request. Access Manager retrieves the username and password and then internally authenticates the user to the LDAP authentication module. In order for HTTP Basic to function correctly, the LDAP authentication module must be added (adding the HTTP Basic module alone will not work). For more information, see "Adding and Enabling LDAP Authentication" on page 201. Once the user successfully authenticates, he/she will be able to re-authenticate without being prompted for username and password.

Adding and Enabling HTTP Basic Authentication

You must log in to Access Manager as the Organization Administrator or Top-Level Administrator and have the LDAP authentication module already registered.

1. Go to the organization for which HTTP Basic Authentication is to be added.
2. Choose Services from the View menu.

   The Core module, if already added, displays in the Navigation pane. If it is not already added, it can be done concurrently with the HTTP Basic Authentication module.
3. Click Add in the Navigation pane.

   A list of available modules displays in the Data pane.
4. Select the checkbox for HTTP Basic Authentication and click Add.

   The HTTP Basic Authentication module will appear in the Navigation pane assuring the administrator that it has been added.
5. Click the HTTP Basic Authentication Properties arrow.

   The message *A template does not currently exist for this service. Do you want to create one now?* appears in the Data pane.

6. Click Create.

   The HTTP Basic Authentication attributes appear in the Data pane. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 22, "HTTP Basic Authentication Attributes" or by clicking the Help link in the upper right corner of the console.

7. Click Save.

   The HTPP Basic Authentication module has been enabled.

Logging In Using HTTP Basic Authentication

In order to log in using HTTP Authentication, the Core Authentication module attribute "Organization Authentication Modules" on page 298 must be modified to enable and select HTTP Basic authentication. This ensures that when the user logs in using `http://hostname:port/server_deploy_URI/UI/Login?module=HTTPBasic`, the user will see the authentication login window. Based on the authentication type that is being used (such as service, role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL. If authentication fails, a new instance should be opened and the user should login again. To logout completely after using HTTP Basic authentication, all of the existing browser instances must be closed, and a new browser instance must be started.

JDBC Authentication

The Java Database Connectivity (JDBC) Authentication module provides a mechanism to allow Access Manager to authenticate users through any SQL databases that provide JDBC technology-enabled drivers. The connection to the SQL database can be either directly through a JDBC driver, or a JNDI connection pool.

NOTE This module has been tested on MySQL4.0 and Oracle 8i

Adding and Enabling JDBC Authentication

You must log in to Identity Server as the Organization Administrator or Top-Level Administrator.

1. Go to the organization for which JDBC Authentication is to be added.
2. Choose Services from the View menu

   The Core module, if already added, displays in the Navigation pane. If it is not already added, it can be done concurrently with the JDBC Authentication module.
3. Click Add in the Navigation pane.

   A list of available modules displays in the Data pane.
4. Select the checkbox for JDBC Authentication and click Add.

   The JDBC Authentication module will appear in the Navigation pane assuring the administrator that it has been added.
5. Click the JDBC Authentication Properties arrow.

   The message *A template does not currently exist for this service. Do you want to create one now?* appears in the Data pane.
6. Click Create.

   The JDBC Authentication attributes appear in the Data pane. Modify the attributes as necessary.
7. Click Save.

   The JDBC Authentication module has been enabled.

Logging In Using JDBC Authentication

In order to log in using JDBC Authentication, the Core Authentication module attribute "Organization Authentication Modules" on page 298 must be modified to enable and select JDBC Authentication. This ensures that when the user logs in using `http://hostname:port/deploy_URI/UI/Login?module=JDBC`, (note case sensitivity) the user will see the JDBC Authentication login window. Based on the authentication type that is being used (such as service, role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

LDAP Directory Authentication

With the LDAP Authentication module, when a user logs in, he or she is required to bind to the LDAP Directory Server with a specific user DN and password. This is the default authenticating module for all organization-based authentication. If the user provides a user id and password that are in the Directory Server, the user is allowed access to, and is set up with, a valid Access Manager session. Both the Core and LDAP Authentication modules are automatically enabled for the default organization.The following instructions are provided in the event that the module is disabled.

Adding and Enabling LDAP Authentication

You must log in to Access Manager as the Organization Administrator or Top-Level Administrator.

1. Go to the organization for which LDAP Authentication is to be added.
2. Choose Services from the View menu.

   The Core module, if already added, displays in the Navigation pane. If it is not already added, it can be done concurrently with the LDAP Authentication module.
3. Click Add in the Navigation pane.

   A list of available modules displays in the Data pane.
4. Select the checkbox for LDAP Authentication and click Add.

   The LDAP Authentication module will appear in the Navigation pane assuring the administrator that it has been added.
5. Click the LDAP Authentication Properties arrow.

   The message *A template does not currently exist for this service. Do you want to create one now?* appears in the Data pane.
6. Click Create.

   The LDAP Authentication attributes appear in the Data pane. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 24, “LDAP Authentication Attributes” or by clicking the Help link in the upper right corner of the console.

7. Enter the password in the Password for Root User Bind attribute. By default, the `amldapuser` password that was entered during installation is used as the bind user. If your Directory Server allows anonymous access to read user entries, you can skip this step.

   To use a different bind user, change the DN of the user in the DN For Root User Bind attribute, and enter the password for that user in the Password for Root User Bind attribute.

8. Click Save.

   The LDAP Authentication module has been enabled.

Logging In Using LDAP Authentication

In order to log in using LDAP Authentication, the Core Authentication module attribute "Organization Authentication Modules" on page 298 must be modified to enable and select LDAP Authentication. This ensures that when the user logs in using `http://hostname:port/server_deploy_URI/UI/Login?module=LDAP`, the user will see the LDAP Authentication login window. Based on the authentication type that is being used (such as service, role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

Enabling LDAP Authentication Failover

The LDAP authentication attributes include a value field for both a primary and a secondary Directory Server. Access Manager will look to the second server for authentication if the primary server becomes unavailable. For more information, see the LDAP attributes "Primary LDAP Server" on page 316 and "Secondary LDAP Server" on page 316.

Multiple LDAP Configuration

As a form of failover or to configure multiple values for an attribute when the Access Manager console only provides one value field, an administrator can define multiple LDAP configurations under one organization. Although these additional configurations are not visible from the console, they work in conjunction with the primary configuration if an initial search for the requesting user's authorization is not found. For information on multiple LDAP configuration, see "Multi LDAP Configuration" in the *Access Manager Developer's Guide.*

Membership Authentication

Membership authentication is implemented similarly to personalized sites such as `my.site.com`, or `mysun.sun.com`. When this module is enabled, a user creates an account and personalizes it without the aid of an administrator. With this new account, the user can access it as a added user. The user can also access the viewer interface, saved on the user profile database as authorization data and user preferences.

Adding and Enabling Membership Authentication

You must log in to Access Manager as the Organization Administrator or Top-Level Administrator.

1. Go to the organization for which Membership Authentication is to be added.
2. Choose Services from the View menu.

   The Core module, if already added, displays in the Navigation pane. If it is not already added, it can be done concurrently with the Membership Authentication module.
3. Click Add in the Navigation pane.

   A list of available modules displays in the Data pane.
4. Select the checkbox for Membership Authentication and click Add.

   The Membership Authentication module will appear in the Navigation pane assuring the administrator that it has been added.
5. Click the Membership Authentication Properties arrow.

   The message *A template does not currently exist for this service. Do you want to create one now?* appears in the Data pane.
6. Click Create.

   The Membership Authentication attributes appear in the Data pane. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 25, “Membership Authentication Attributes” or by selecting the Help link in the upper right corner of the console.

7. Enter the password in the Password for Root User Bind attribute. By default, the `amldapuser` password that was entered during installation is used as the bind user.

   To use a different bind user, change the DN of the user in the DN For Root User Bind attribute, and enter the password for that user in the Password for Root User Bind attribute.

8. Click Save.

   The Membership Authentication module has been enabled.

Logging In Using Membership Authentication

In order to log in using Membership Authentication, the Core Authentication module attribute "Organization Authentication Modules" on page 298 must be modified to enable and select Membership Authentication. This ensures that when the user logs in using `http://`*`hostname:port/deploy_URI`*`/UI/Login?module=Membership`, (note case sensitivity) the user will see the Membership Authentication login (Self Registration) window. Based on the authentication type that is being used (such as module, role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

MSISDN Authentication

The Mobile Station Integrated Services Digital Network (MSISDN) authentication module enables authentication using a mobile subscriber ISDN associated with a device such as a cellular telephone. It is a non-interactive module. The module retrieves the subscriber ISDN and validates it against the Directory Server to find a user that matches the number.

Adding and Enabling MSISDN Authentication

You must log in to Identity Server as the Organization Administrator or Top-Level Administrator.

1. Go to the organization for which MSISDN Authentication is to be added.

2. Choose Services from the View menu

   The Core module, if already added, displays in the Navigation pane. If it is not already added, it can be done concurrently with the MSISDN Authentication module.

3. Click Add in the Navigation pane.

   A list of available modules displays in the Data pane.

4. Select the checkbox for MSISDN Authentication and click Add.

   The MSISDN Authentication module will appear in the Navigation pane assuring the administrator that it has been added.

5. Click the MSISDN Authentication Properties arrow.

   The message *A template does not currently exist for this service. Do you want to create one now?* appears in the Data pane.

6. Click Create.

   The MSISDN Authentication attributes appear in the Data pane. Modify the attributes as necessary.

7. Click Save.

   The MSISDN Authentication module has been enabled.

Logging In Using MSISDN Authentication

In order to log in using MSISDN Authentication, the Core Authentication module attribute "Organization Authentication Modules" on page 298 must be modified to enable and select MSISDN Authentication. This ensures that when the user logs in using `http://hostname:port/deploy_URI/UI/Login?module=MSISDN`, (note case sensitivity) the user will see the MSISDN Authentication login window. Based on the authentication type that is being used (such as service, role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

Windows NT Authentication

Access Manager can be configured to work with an Windows NT / Windows 2000 server that is already installed. Access Manager provides the client portion of NT authentication.

1. Configure the NT server. For detailed instructions, see the Windows NT server documentation.
2. Before you can add and enable the Windows NT authentication module, you must obtain and install a Samba client to communicate with Access Manager on your Solaris system. For more information, see "NT Authentication Attributes" on page 331.
3. Add and enable the Windows NT authentication module.

Installing the Samba Client

In order to activate the Windows NT Authentication module, Samba Client 2.2.2 must be downloaded and installed to the following directory:

```
AccessManager-base/SUNWam/bin
```

Samba Client is a file and print server for blending Windows and UNIX machines together without requiring a separate Windows NT/2000 Server. More information, and the download itself, can be accessed at http://wwws.sun.com/software/download/products/3e3af224.html.

Red Hat Linux ships with a Samba client, located in the following directory:

```
/usr/bin
```

In order to authenticate using the Windows NT Authentication module for Linux, copy the client binary to the following Access Manager directory:

AccessManager-base`/sun/identity/bin`

NOTE If you have multiple interfaces, extra configuration is required. Multiple interfaces can be set by configuration in the `smb.conf` file so it passes to the `mbclient`.

Adding and Enabling Windows NT Authentication

You must log in to Access Manager as the Organization Administrator or Top-Level Administrator.

1. Go to the organization for which Windows NT Authentication is to be added.
2. Choose Services from the View menu.

   The Core module, if already added, displays in the Navigation pane. If it is not already added, it can be done concurrently with the Windows NT Authentication module.
3. Click Add in the Navigation pane.

   A list of available modules displays in the Data pane.
4. Select the checkbox for Windows NT Authentication and click Add.

   The Windows NT Authentication module will appear in the Navigation pane assuring the administrator that it has been added.
5. Click the Windows NT Authentication Properties arrow.

   The message *A template does not currently exist for this service. Do you want to create one now?* appears in the Data pane.
6. Click Create.

   The Windows NT Authentication attributes appear in the Data pane. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 27, “NT Authentication Attributes” or by selecting the Help link in the upper right corner of the console.
7. Click Save.

   The Windows NT Authentication module has been enabled.

Logging In Using Windows NT Authentication

In order to log in using Windows NT Authentication, the Core Authentication module attribute “Organization Authentication Modules” on page 298 must be modified to enable and select NT Authentication. This ensures that when the user logs in using `http://hostname:port/deploy_URI/UI/Login?module=NT`, the

user will see the Windows NT Authentication login window. Based on the authentication type that is being used (such as service, role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

RADIUS Server Authentication

Access Manager can be configured to work with a RADIUS server that is already installed. This is useful if there is a legacy RADIUS server being used for authentication in your enterprise. Enabling the RADIUS authentication module is a two-step process:

1. Configure the RADIUS server.

   For detailed instructions, see the RADIUS server documentation.

2. Register and enable the RADIUS authentication module.

Adding and Enabling RADIUS Authentication

You must log in to Access Manager as the Organization Administrator.

1. Go to the organization for which RADIUS Authentication is to be added.
2. Choose Services from the View menu.

   The Core module, if already added, displays in the Navigation pane. If it is not already added, it can be done concurrently with the RADIUS Authentication module.

3. Click Add in the Navigation pane.

   A list of available modules displays in the Data pane.

4. Select the checkbox for RADIUS Authentication and click Add.

   The RADIUS Authentication module will appear in the Navigation pane assuring the administrator that it has been added.

5. Click the RADIUS Authentication Properties arrow.

   The message *A template does not currently exist for this service. Do you want to create one now?* appears in the Data pane.

6. Click Create.

   The RADIUS Authentication attributes appear in the Data pane. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 28, "RADIUS Authentication Attributes" or by selecting the Help link in the upper right corner of the console.

7. Click Save.

   The RADIUS Authentication module has been enabled.

Logging In Using RADIUS Authentication

In order to log in using RADIUS Authentication, the Core Authentication module attribute "Organization Authentication Modules" on page 298 must be modified to enable and select RADIUS Authentication. This ensures that when the user logs in using `http://hostname:port/deploy_URI/UI/Login?module=RADIUS`, the user will see the RADIUS Authentication login window. Based on the authentication type that is being used (such as service, role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

Configuring RADUIS with Sun ONE Application Server

When the RADUIS client forms a socket connection to its server, by default, only the connect permission of the SocketPermissions is allowed in the Application Server's `server.policy` file. In order for RADUIS authentication to work correctly, permissions need to be granted for the following actions:

- accept
- connect
- listen
- resolve

To grant a permission for a socket connection, you must add an entry into Application Server's `server.policy` file. A SocketPermission consists of a host specification and a set of actions specifying ways to connect to that host. The host is specified as the following:

`host = hostname | IPaddress:portrange:portrange = portnumber | -portnumberportnumber-portnumber`

The host is expressed as a DNS name, as a numerical IP address, or as localhost (for the local machine). The wildcard "*" may be included once in a DNS name host specification. If it is included, it must be in the left-most position, as in `*.example.com`.

The port (or portrange) is optional. A port specification of the form `N-`, where `N` is a port number, signifies all ports numbered `N` and above. A specification of the form `-N` indicates all ports numbered `N` and below.

The `listen` action is only meaningful when used with a localhost. The `resolve` (resolve host/IP name service lookups) action is implied when any of the other actions are present.

For example, when creating SocketPermissions, note that if the following permission is granted to some code, it allows that code to connect to `port 1645` on `machine1.example.com`, and to accept connections on that port:

```
permission java.net.SocketPermission machine1.example.com:1645,
"connect,accept";
```

Similarly, if the following permission is granted to some code, it allows that code to accept connections on, connect to, or listen to any port between 1024 and 65535 on the local host:

```
permission java.net.SocketPermission "machine1.example.com:1645",
"connect,accept";

permission java.net.SocketPermission "localhost:1024-",
"accept,connect,listen";
```

NOTE Granting code permission to accept or make connections to remote hosts may cause problems, because malevolent code can then more easily transfer and share confidential data among parties who may not otherwise have access to the data. Make sure to give only appropriate permissions by specifying exact port number instead of allowing a range of port numbers

SafeWord Authentication

Access Manager can be configured to handle SafeWord Authentication requests to Secure Computing's SafeWord™ or SafeWord PremierAccess™ authentication servers. Access Manager provides the client portion of SafeWord authentication. The SafeWord server may exist on the system on which Access Manager is installed, or on a separate system.

Adding and Enabling SafeWord Authentication

You must log in to Access Manager as the Organization Administrator or Top-Level Administrator.

1. Go to the organization for which SafeWord Authentication is to be added.
2. Choose Services from the View menu.

   The Core module, if already added, displays in the Navigation pane. If it is not already added, it can be done concurrently with the SafeWord Authentication module.
3. Click Add in the Navigation pane.

   A list of available modules displays in the Data pane.
4. Select the checkbox for SafeWord Authentication and click Add.

   The SafeWord Authentication module will appear in the Navigation pane, assuring the administrator that it has been added.
5. Click the SafeWord Authentication Properties arrow.

   The message *A template does not currently exist for this service. Do you want to create one now?* appears in the Data pane.
6. Click Create.

   The SafeWord Authentication attributes appear in the Data pane. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 29, “SafeWord Authentication Attributes”, or by clicking the Help link on the upper right corner of the console.
7. Click Save.

   The SafeWord Authentication module has been enabled.

Logging In Using SafeWord Authentication

In order to log in using SafeWord Authentication, the Core Authentication module attribute “Organization Authentication Modules” on page 298 must be modified to enable and select SafeWord Authentication. This ensures that when the user logs in using `http://hostname:port/deploy_URI/UI/Login?module=SafeWord`, the

user will see the SafeWord Authentication login window. Based on the authentication type that is being used (such as role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

Configuring SafeWord with Sun ONE Application Server

When the SafeWord client forms a socket connection to its server, by default, only the `connect` permission of the SocketPermissions is allowed in the Application Server's `server.policy` file. In order for SafeWord authentication to work correctly, permissions need to be granted for the following actions:

- accept
- connect
- listen
- resolve

To grant a permission for a socket connection, you must add an entry into Application Server's `server.policy` file. A SocketPermission consists of a host specification and a set of actions specifying ways to connect to that host. The host is specified as the following:

```
host = (hostname | IPaddress)[:portrange] portrange = portnumber |
-portnumberportnumber-[portnumber]
```

The host is expressed as a DNS name, as a numerical IP address, or as localhost (for the local machine). The wildcard "*" may be included once in a DNS name host specification. If it is included, it must be in the left-most position, as in `*.example.com`.

The port (or portrange) is optional. A port specification of the form `N-`, where `N` is a port number, signifies all ports numbered `N` and above. A specification of the form `-N` indicates all ports numbered `N` and below.

The `listen` action is only meaningful when used with a localhost. The `resolve` (resolve host/IP name service lookups) action is implied when any of the other actions are present.

For example, when creating SocketPermissions, note that if the following permission is granted to some code, it allows that code to connect to `port 1645` on `machine1.example.com`, and to accept connections on that port:

```
permission java.net.SocketPermission machine1.example.com:5030,
"connect,accept";
```

Similarly, if the following permission is granted to some code, it allows that code to accept connections on, connect to, or listen to any port between 1024 and 65535 on the local host:

```
permission java.net.SocketPermission "machine1.example.com:5030",
"connect,accept";
```

```
permission java.net.SocketPermission "localhost:1024-",
"accept,connect,listen";
```

NOTE Granting code permission to accept or make connections to remote hosts may cause problems, because malevolent code can then more easily transfer and share confidential data among parties who may not otherwise have access to the data. Make sure to give only appropriate permissions by specifying exact port number instead of allowing a range of port numbers

SAML Authentication

The Security Assertion Markup Language (SAML) authentication module receives and validates SAML Assertions on a target server. SAML SSO will only work if this module is configured on the target machine, including after an upgrade (for example, Access Manager 2004Q2 to Access Manager 2005Q1).

Adding and Enabling SAML Authentication

You must log in to Access Manager as the Organization Administrator or Top-Level Administrator and have the LDAP authentication module already registered.

1. Go to the organization for which SAML Authentication is to be added.
2. Choose Services from the View menu.

   The Core module, if already added, displays in the Navigation pane. If it is not already added, it can be done concurrently with the SAML Authentication module.
3. Click Add in the Navigation pane.

   A list of available modules displays in the Data pane.

4. Select the checkbox for SAML Authentication and click Add.

   The SAML Authentication module will appear in the Navigation pane assuring the administrator that it has been added.

5. Click the SAML Authentication Properties arrow.

   The message *A template does not currently exist for this service. Do you want to create one now?* appears in the Data pane.

6. Click Create.

   The SAML Authentication attributes appear in the Data pane. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 22, "HTTP Basic Authentication Attributes" or by clicking the Help link in the upper right corner of the console.

7. Click Save.

   The SAML Authentication module has been enabled.

Logging In Using SAML Authentication

In order to log in using SAML Authentication, the Core Authentication module attribute "Organization Authentication Modules" on page 298 must be modified to enable and select HTTP Basic authentication. This ensures that when the user logs in using `http://hostname:port/server_deploy_URI/UI/Login?module=SAML`, the user will see the authentication login window. Based on the authentication type that is being used (such as service, role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

SecurID Authentication

Access Manager can be configured to handle SecureID Authentication requests to RSA's ACE/Server authentication servers. Access Manager provides the client portion of SecurID authentication. The ACE/Server may exist on the system on which Access Manager is installed, or on a separate system. In order to authenticate locally-administered userids (see `admintool` (1M)), root access is required.

SecurID Authentication makes use of an authentication *helper*, `amsecuridd`, which is a separate process from the main Access Manager process. Upon startup, this helper listens on a port for configuration information. If Access Manager is installed to run as `nobody`, or a userid other than root, then the *`AccessManager-base`*`/SUNWam/share/bin/amsecuridd` process must still execute as root. For more information on the `amsecuridd` helper, see "The amsecuridd Helper" on page 255.

NOTE For this release of Access Manager, the SecurID Authentication module is not available for the Linux or Solaris x86 platforms and this should not be registered, configured, or enabled on these two platforms. It is only available for SPARC systems.

Adding and Enabling SecurID Authentication

You must log in to Access Manager as the Organization Administrator or Top-Level Administrator.

1. Go to the organization for which SecurID Authentication is to be added.
2. Choose Services from the View menu.

   The Core module, if already added, displays in the Navigation pane. If it is not already added, it can be done concurrently with the SecurID Authentication module.
3. Click Add in the Navigation pane.

   A list of available modules displays in the Data pane.
4. Select the checkbox for SecurID Authentication and click Add.

   The SecurID Authentication module will appear in the Navigation pane, assuring the administrator that it has been added.
5. Click the SecurID Authentication Properties arrow.

   The message *A template does not currently exist for this service. Do you want to create one now?* appears in the Data pane.

6. Click Create.

   The SecurID Authentication attributes appear in the Data pane. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 31, "SecurID Authentication Attributes", or by clicking the Help link on the upper right corner of the console.

7. Click Save.

   The SecurID Authentication module has been enabled.

Logging In Using SecurID Authentication

In order to log in using SecurID Authentication, the Core Authentication module attribute "Organization Authentication Modules" on page 298 must be modified to enable and select SecurID Authentication. This ensures that when the user logs in using `http://hostname:port/deploy_URI/UI/Login?module=SecurID,` the user will see the SecurID Authentication login window. Based on the authentication type that is being used (such as role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

Unix Authentication

Access Manager can be configured to process authentication requests against Unix userids and passwords known to the Solaris or Linux system on which Access Manager is installed. While there is only one organizational attribute, and a few global attributes for Unix authentication, there are some system-oriented considerations. In order to authenticate locally-administered userids (see `admintool` (1M)), root access is required.

Unix Authentication makes use of an authentication *helper*, `amunixd`, which is a separate process from the main Access Manager process. Upon startup, this helper listens on a port for configuration information. There is only one Unix helper per Access Manager to serve all of its organizations.

If Access Manager is installed to run as `nobody`, or a userid other than root, then the *`AccessManager-base`*`/SUNWam/share/bin/amunixd` process must still execute as root. The Unix authentication module invokes the `amunixd` daemon by opening a socket to `localhost:58946` to listen for Unix authentication requests. To run the `amunixd` helper process on the default port, enter the following command:

```
./amunixd
```

To run amunixd on a non-default port, enter the following command:

```
./amunixd [-c portnm] [ipaddress]
```

The ipaddress and portnumber is located in the UnixHelper.ipadrs (in IPV4 format) and UnixHelper.port attributes in AMConfig.properties. You can run amunixd through the amserver command line utility (amserver runs the process automatically, retrieving the portnumber and ipaddress from AMConfig.properties).

The passwd entry in the /etc/nsswitch.conf file determines whether the /etc/passwd and /etc/shadow files, or NIS are consulted for authentication.

Adding and Enabling Unix Authentication

You must log in to the Access Manager as Top-Level Administrator for the following steps.

1. Select the Service Configuration module.
2. Click on the Unix Authentication Properties arrow in the Service Name list.

   Several Global and one Organization attributes are displayed. Because one Unix helper serves all of the Access Manager server's organizations, most of the Unix attributes are global. An explanation of these attributes can be found in Chapter 32, “Unix Authentication Attributes”, or by clicking the Help link in the upper right corner of the console.
3. Click Save to save the new values for the attributes.

   You may log in to Access Manager as the Organization Administrator to enable Unix Authentication for an organization.
4. Go to the organization for which Unix Authentication is to be added.
5. Choose Services from the View menu.

   The Core module, if already added, displays in the Navigation pane. If it is not already added, it can be done concurrently with the Unix Authentication module.
6. Click Add in the Navigation pane.

   A list of available modules displays in the Data pane.

7. Select the checkbox for Unix Authentication and click Add.

   The Unix Authentication module will appear in the Navigation pane, assuring the administrator that it has been added.

8. Click the Unix Authentication Properties arrow.

   The message *A template does not currently exist for this service. Do you want to create one now?* appears in the date pane.

9. Click Create.

   The Unix Authentication organization attribute appears in the Data pane. Modify the Authentication Level attribute as necessary. An explanation of this attribute can be found in Chapter 32, "Unix Authentication Attributes", or by clicking the Help link in the upper right corner of the console.

10. Click Save. The Unix Authentication module is enabled.

Logging In Using Unix Authentication

In order to log in using Unix Authentication, the Core Authentication module attribute "Organization Authentication Modules" on page 298 must be modified to enable and select Unix Authentication. This ensures that when the user logs in using `http://hostname:port/deploy_URI/UI/Login?module=Unix`, the user will see the Unix Authentication login window. Based on the authentication type that is being used (such as service, role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

Windows Desktop SSO Authentication

The Windows Desktop SSO Authentication module is a Kerberos-based authentication plug-in module used for Windows 2000™. It allows a user who has already authenticated to a Kerberos Distribution Center (KDC) to authenticate to Access Manager without re-submitting the login criteria (Single Sign-on).

The user presents the Kerberos token to the Access Manager through the SPNEGO (Simple and Protected GSS-API Negotiation Mechanism) protocol. In order to perform Kerberos-based Single Sign-on to Access Manager through this authentication module, the user must, on the client side, support the SPNEGO protocol to authenticate itself. In general, any user that supports this protocol should be able to use this module to authenticate to Access Manager. Depending

on the availability of the token on the client side, this module provides a SPENGO token or a Kerberos token (in both cases, the protocols are the same).Microsoft Internet Explorer (5.01 or later) running on Windows 2000 (or later) currently supports this protocol. In addition, Mozilla 1.4 on Solaris (9 and 10) has SPNEGO support, but the token returned is only a KERBEROS token, because SPNEGO is not supported on Solaris.

NOTE You must use JDK 1.4 or above to utilize the new features of Kerberos V5 authentication module and Java GSS API to perform Kerberos based SSO in this SPNEGO module.

Known Restriction with Internet Explorer

If you are using Microsoft Internet Explorer 6.x when for WindowsDesktopSSO authentication and the browser does not have access to the user's kerberos / SPNEGO token that matches the (KDC) realm configured in the WindowsDesktopSSO module, the browser will behave incorrectly to other modules after it fails authenticating to the WindowsDesktopSSO module. The direct cause of the problem is that after Internet Explorer fails the WindowsDesktopSSO module, the browser becomes incapable of passing callbacks (of other modules) to Access Manager, even if the callbacks are prompted, until the browser is restarted. Therefore all the modules coming after WindowsDesktopSSO will fail due to null user credentials.

See the following documentation for related information:

http://support.microsoft.com/default.aspx?scid=kb;en-us;308074

http://www.wedgetail.com/jcsi/sso/doc/guide/troubleshooting.html#ieNTLM

Adding and Enabling Windows Desktop SSO Authentication

Enabling Windows Desktop SSO Authentication is a three-step process:

1. Create a User in the Windows 2000 Domain Controller.
2. Setup Internet Explorer.
3. Add and Configure the Windows Desktop SSO Authentication module.

To Create a User in the Windows 2000 Domain Controller

1. In the domain controller, create a user account for the Access Manager authentication module.
   a. From the Start menu, go to Programs>Administration Tools.
   b. Select Active Directory Users and Computers.
   c. Create a new user with the Access Manager host name as the User ID (login name). The Access Manager host name should not include the domain name.
2. Associate the user account with a service provider name and export the keytab files to the system in which Access Manager is installed. To do so, run the following commands:

   `ktpass -princ host/`*`hostname.domainname`*`@DCDOMAIN -pass `*`password`*` -mapuser `*`userName`*`-out `*`hostname`*`.host.keytab`

   `ktpass -princ HTTP/`*`hostname.domainname`*`@DCDOMAIN -pass `*`password`*` -mapuser `*`userName`*`-out `*`hostname`*`.HTTP.keytab`

   The `ktpass` command accepts the following parameters:

   hostname. The host name (without the domain name) on which Access Manager runs.

   domainname. The Access Manager domain name.

   DCDOMAIN. The domain name of the domain controller. This may be different from the Access Manager domain name.

   password. The password of the user account. Make sure that password is correct, as `ktpass` does not verify passwords.

   userName. The user account ID. This should be the same as hostname.

NOTE Make sure that both keytab files are kept secure.

The service template values should be similar to the following example:

Service Principal: `HTTP/machine1.EXAMPLE.COM@ISQA.EXAMPLE.COM`

Keytab File Name: `/tmp/machine1.HTTP.keytab`

Kerberos Realm: `ISQA.EXAMPLE.COM`

Kerberos Server Name: `machine2.EXAMPLE.com`

Return Principal with Domain Name: `false`

Authentication Level: `22`

3. Restart the server.

To Set Up Internet Explorer

These steps apply to Microsoft Internet Explorer™ 6 and later. If you are using an earlier version, make sure that Access Manager is in the browser's internet zone and enable Native Windows Authentication.

1. In the Tool menu, go to Internet Options>Advanced / Security>Security.
2. Select the Integrated Windows Authentication option.
3. Go to Security>Local Internet.
   a. Select Custom Level. In the User Authentication / Logon panel, select the Automatic Logon Only in Intranet Zone option.
   b. Go to Sites and select all of the options.
   c. Click Advanced and add the Access Manager to the local zone (if it is not added already).

Known Restriction with Internet Explorer

If you are using Microsoft Internet Explorer 6.x when for WindowsDesktopSSO authentication and the browser does not have access to the user's kerberos / SPNEGO token that matches the (KDC) realm configured in the WindowsDesktopSSO module, the browser will behave incorrectly to other modules after it fails authenticating to the WindowsDesktopSSO module. The direct cause of the problem is that after Internet Explorer fails the

WindowsDesktopSSO module, the browser becomes incapable of passing callbacks (of other modules) to Access Manager, even if the callbacks are prompted, until the browser is restarted. Therefore all the modules coming after WindowsDesktopSSO will fail due to null user credentials.

See the following documentation for related information:

http://support.microsoft.com/default.aspx?scid=kb;en-us;308074

http://www.wedgetail.com/jcsi/sso/doc/guide/troubleshooting.html#ieNTLM

To Add and Configure Windows Desktop SSO Authentication

You must log in to Access Manager as the Organization Administrator or Top-Level Administrator.

1. Go to the organization for which Windows Desktop SSO Authentication is to be added.
2. Choose Services from the View menu.

   The Core module, if already added, displays in the Navigation pane. If it is not already added, it can be done concurrently with the Windows Desktop SSO Authentication module.
3. Click Add in the Navigation pane.

   A list of available modules displays in the Data pane.
4. Select the checkbox for Windows Desktop SSO Authentication and click Add.

   The Windows Desktop SSO Authentication module will appear in the Navigation pane assuring the administrator that it has been added.
5. Click the Windows Desktop SSO Authentication Properties arrow.

   The message *A template does not currently exist for this service. Do you want to create one now?* appears in the Data pane.
6. Click Create.

   The Windows Desktop SSO Authentication attributes appear in the Data pane. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 33, “Windows Desktop SSO Authentication Attributes” or by selecting the Help link in the upper right corner of the console.
7. Click Save. The Windows Desktop SSO Authentication module is enabled.

Logging In Using Windows Desktop SSO Authentication

In order to log in using Windows Desktop SSO Authentication, the Core Authentication module attribute “Organization Authentication Modules” on page 298 must be modified to enable and select Windows Desktop SSO Authentication. This ensures that when the user logs in from a host which is part of the Windows 2000 Domain Controller and has logged in as a domain user using `http://hostname:port/deploy_URI/UI/Login?module=WindowsDesktopSSO`, the user will be authenticated. Based on the authentication type that is being used (such as service, role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

Chapter 9

Password Reset Service

Sun Java™ System Access Manager 6 2005Q1 provides a Password Reset service to allow users to reset their password for access to a given service or application protected by Access Manager.The Password Reset service attributes, defined by the top-level administrator, control user validation credentials (in the form of *secret questions*), control the mechanism for new or existing password notification, and sets possible lockout intervals for incorrect user validation.

This chapter contains the following sections:

Registering the Password Reset Service

The Password Reset service does not need to be registered for the organization in which the user resides. If the Password Reset service does not exist in the organization in which the user resides, it will inherit the values defined for the service in the Service Configuration module.

To Register Password Reset for Users in a Different Organization

1. In the Identity Management module, choose Organizations and select the organization for which you wish to register the service.

2. Click Register in the Navigation frame.

   A list of available services displays in the Data frame.

3. Select the checkbox for Password Reset and click Register.

   The Password Reset service will appear in the Navigation frame assuring the administrator that it has been registered.

Configuring the Password Reset Service

Once the Password Reset service has been registered, the service must be configured by a user with administrator privileges.

To Configure the Service

1. Select the organization for which the Password Reset service is registered.
2. Click the Password Reset Properties arrow.

   The message “No template available for this service” appears in the Data frame. Click Create.

3. The Password Reset attributes appear in the Data frame allowing you to define requirements for the Password Reset service. Make sure that the Password Reset service is enabled (it is by default). At a minimum, the following attributes must be defined:

   - User Validation
   - Secret Question
   - Bind DN
   - Bind Password

   The Bind DN attribute must contain a user with privileges for resetting the password (for example, Help Desk Administrator). Due a limitation in Directory Server, Password Reset does not work when the bind DN is `cn=Directory Manager`.

   The remaining attributes are optional. Descriptions of the Password Reset attributes can be found in “Password Reset Service Attributes” on page 375 or by clicking the Help link in the upper right corner of the console.

NOTE Access Manager automatically installs the Password Reset web application for random password generation. However, you can write your own plug-in classes for password generation and password notification. See the following Readme.html files in the following locations for samples for these plug-in classes.

PasswordGenerator:

`AccessManager-base/SUNWam/samples/console/PasswordGenerator`

NotifyPassword:

`AccessManager-base/SUNWam/samples/console/NotifyPassword`

4. Select the Personal Question Enabled attribute if the user is to define his/her unique personal questions. Once the attributes are defined, click Save.

Password Reset Lockout

The Password Reset service contains a lockout feature that will restrict users to a certain number of attempts to correctly answer their secret questions. The lockout feature is configured through the Password Reset service attributes. Descriptions of these attributes can be found in "Password Reset Service Attributes" on page 375. Password Reset supports two types of lockout, memory lockout and physical lockout.

Memory Lockout

This is a temporary lockout and is in effect only when the value in the Password Reset Failure Lockout Duration attribute is greater than zero and the Enable Password Reset Failure Lockout attribute is enabled. This lockout will prevent users from resetting their password through the Password Reset web application. The lockout lasts for the duration specified in Password Reset Failure Lockout Duration, or until the server is restarted.

Physical Lockout

This is a more permanent lockout. If the value set in the Password Reset Failure Lockout Count attribute is set to 0 and the Enable Password Reset Failure Lockout attribute is enabled, the users' account status is changed to inactive when he or she incorrectly answers the secret questions.

Password Reset for End Users

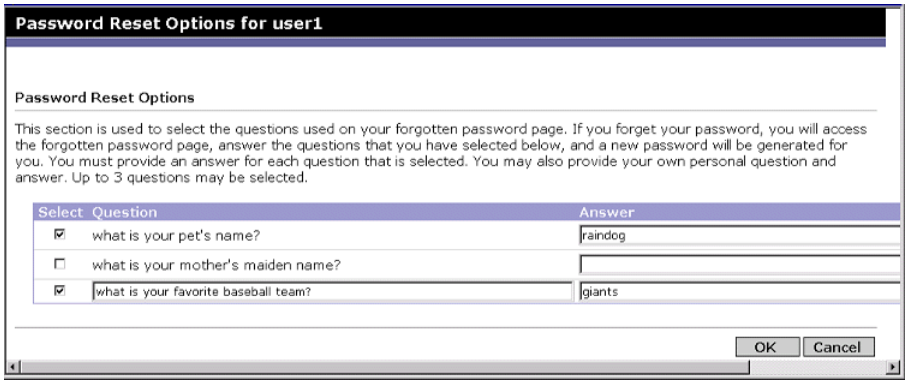
The following sections describe the user experience for the Password Reset service.

Customizing Password Reset

Once the Password Reset service has been enabled and the attributes defined by the administrator, users are able to log into the Access Manager console in order to customize their secret questions. For example:

1. The user logs into the Access Manager console, providing Username and Password and is successfully authenticated.
2. In the User Profile page, the user selects Password Reset Options. This displays the Available Questions Answer Screen.
3. The user is presented with the available questions that the administrator defined for the service, such as:
   - What is your pet's name?
   - What is your favorite TV show?
   - What is your mother's maiden name?
   - What is your favorite restaurant?
4. The user selects the secret questions, up to the maximum number of questions that the administrator defined for the organization (the maximum amount is defined the Password Reset Service). The user then provides answers to the selected questions. These questions and answers will be the basis for resetting the user's password (see the following section). If the administrator has selected the Personal Question Enabled attribute, text fields are provided, allowing the user to enter a unique secret question and provide an answer.

Figure 9-1 Available Questions Answer Screen with Personal Question Enabled



5. The user clicks Save.

Resetting Forgotten Passwords

In the case where users forget their password, Access Manager uses the Password Reset web application to randomly generate new passwords and notify the user of the new password. A typical forgotten password scenario follows:

1. The user logs into the Password Reset web application from a URL given to them by the administrator. For example:

   `http://`*`hostname:port`*`/ampassword` (for the default organization)

   or

   `http://`*`hostname: port`*`/`*`deploy_uri`*`/UI/PWResetUserValidation?org=`*`orgname`*, where *orgname* is the name of the organization.

NOTE If the Password Reset service is not enabled for a parent organization, but is enabled for a sub-organization, users must use the following syntax to access the service:

`http://`*`hostname:`*
`port``/`*`deploy_uri`*`/UI/PWResetUserValidation?org=`*`orgname`*

2. The user enters the user id.

3. The user is presented with the personal questions that were defined in the Password Reset service and select by the user during customization. If the user has not previously logged into the User Profile page and customized the personal questions, the password will not be generated.

   Once the user answers the questions correctly, the new password is generated and emailed to the user. Attempt notification is sent to the user whether the questions are answered correctly or not. Users must have their email address entered in the User Profile page in order for the new password and attempt notification to be received.

Password Policies

A secure password policy minimizes the risks associated with easily-guessed passwords by enforcing the following:

- Users must change their passwords according to a schedule.
- Users must provide non-trivial passwords.
- Accounts may be locked after a number of binds with the wrong password.

Directory Server provides several ways to set password policy at any node in a tree and there are several ways to set the policy. For details refer following Directory Server documentation:

`http://docs.sun.com/source/816-6700-10/aci.html#14773`

`http://docs.sun.com/source/816-6698-10/useracct.html#14386`

Part III

Command Line Reference Guide

This is the Command Line Reference Guide, part three of the Sun Java™ System Access Manager 6 2005Q1 Administration Guide. This section contains the following chapters:

All of the command line tools described in this section can be found in the following default locations"

```
AccessManager-base/SUNWam/bin (Solairs)

AccessManager-base/identity/bin (Linux)
```

Chapter 10

The amadmin Command Line Tool

This chapter provides information on the amadmin command line tool and contains the following sections:

The amadmin Command Line Executable

The primary purposes of the command line executable amadmin is to load XML service files into the Directory Server and to perform batch administrative tasks on the DIT. amadmin can be found in AccessManager-base/SUNWam/bin and is used to:

- Load XML service files - Administrators load services into Access Manager that use the XML service file format defined in the sms.dtd. All services must be loaded using amadmin; they cannot be imported through the Access Manager console.

NOTE XML service files are stored in the Directory Server as static *blobs* of XML data that is referenced by Access Manager. This information is not used by Directory Server which only understands LDAP.

- Perform batch updates of identity objects to the DIT - Administrators can perform batch updates to the Directory Server DIT using the batch processing XML file format defined in the amadmin.dtd. For example, if an administrator wants to create 10 organizations, 1000 users, and 100 groups, it can be done in one attempt by putting the requests in one or more batch processing XML files and loading them using amadmin. More information on this can be found in the "Service Management" chapter in the *Access Manager Developer's Guide*.

NOTE amadmin only supports a subset of features that the Access Manager console supports and is not intended as a replacement. It is recommended that the console be used for small administrative tasks while amadmin is used for larger administrative tasks.

The amadmin Syntax

There are a number of structural rules that must be followed in order to use amadmin. The generic syntaxes for using the tool are:

- amadmin -u | --runasdn *dnname* -w | --password *password* [-l | --locale *localename*] [[-v | --verbose] | [-d |--debug]] -t | --data *xmlfile1* [*xmlfile2* ...]
- amadmin -u | --runasdn *dnname* -w | --password *password* [-l | --locale *localename*] [[-v | --verbose] | [-d | --debug]] -s | --schema *xmlfile1* [*xmlfile2* ...]
- amadmin -u | --runasdn *dnname* -w | --password *password* [-l | --locale *localename*] [[-v | --verbose] | [-d | --debug]] -r | --deleteService *serviceName1* [*serviceName2* ...]
- amadmin -u | --runasdn *dnname* -w | --password *password* or -f | --password file *passwordfile* [-c | --continue] [-l | --locale *localename*] [[-v | --verbose] | [-d | --debug]] -m | --session *servername pattern*
- amadmin -h | --help
- amadmin -n | --version
- amadmin -u | --runasdn *dnname* -w | --password *password* or - f |--passwordfile *passwordfile* [-l | --locale *localename*] [[-v | --verbose] | [-d] |--debug]] -a |--addAttributes *serviceName schemaType xmlfile*[*xmlfile2*] ...

NOTE Two hyphens must be entered exactly as shown in the syntax.

amadmin Options

Following are definitions of the amadmin command line parameter options:

--runasdn (-u)

`--runasdn` is used to authenticate the user to the LDAP server. The argument is a value equal to that of the Distinguished Name (DN) of the user authorized to run `amadmin`; for example

`--runasdn uid=amAdmin,ou=People,o=iplanet.com,o=isp`.

The DN can also be formatted by inserting spaces between the domain components and double quoting the entire DN such as: `--runasdn "uid=amAdmin, ou=People, o=iplanet.com, o=isp"`.

--password (-w)

`--password` is a mandatory option and takes a value equal to that of the password of the DN specified with the `--runasdn` option.

--locale (-l)

`--locale` is an option that takes a value equal to that of the name of the locale. This option can be used for the customization of the message language. If not provided, the default locale, `en_US`, is used.

--continue (-c)

--continue is an option that will continue to process the XML files even if there are errors. For example, if there are three XML files to be loaded at the same time, and the first XML file fails, `amadmin` will continue to load the remaining files. The continue option only applies to separate requests.

--session (-m)

`--session` (`-m`) is an option to manage the sessions, or to display the current sessions. When specifying `--runasdn`, it must be the same as the DN for the super user in `AMConfig.properties`, or just ID for the top-level admin user.

The following example will display all sessions for a particular service host name,:

```
amadmin -u uid=amadmin,ou=people,dc=iplanet,dc=com -v  -w 12345678 -m
http://sun.com:58080
```

The following example will display a particular user's session:

```
amadmin -u uid=amadmin,ou=people,dc=iplanet,dc=com -v  -w 12345678 -m
http://sun.com:58080 username
```

You can terminate a session by entering the corresponding index number, or enter multiple index numbers (with spaces) to terminate multiple sessions.

While using the following option:

`amadmin -m | --session servername pattern`

The *pattern* may be a wildcard (*). If this pattern is using a wildcard (*), it has to be escaped with a meta character (\) from the shell.

--debug (-d)

`--debug` is an option that will write messages to the `amAdmin` file created under the `/var/opt/SUNWam/debug` directory. These messages are technically-detailed but not i18n-compliant. To generate `amadmin` operation logs, when logging to database, the classpath for the database driver needs to be added manually. For example, add the following lines when logging to `mysql` in `amadmin`:

```
CLASSPATH=$CLASSPATH:/opt/IS61/SUNWam/lib/mysql-connector-java-3.0.6-st
able-bin.jar

export CLASSPATH
```

--verbose (-v)

`--verbose` is an option that prints to the screen the overall progress of the `amadmin` command. It does not print to a file the detailed information. Messages output to the command line are i18n- compliant.

--data (-t)

`--data` is an option that takes as its value the name of the batch processing XML file being imported. One or more XML files can be specified. This XML file can create, delete and read various directory objects as well as register and unregister services. For more information on what types of XML files can be passed to this option, see the “Servic Management” chapter in the *Access Manager Developer’s Guide*.

--schema (-s)

`--schema` is an option that loads the attributes of an Access Manager service into the Directory Server. It takes as an argument an XML service file in which the service attributes are defined. This XML service file is based on the `sms.dtd`. One or more XML files can be specified.

NOTE Either the `--data` or `--schema` option must be specified, depending on whether configuring batch updates to the DIT, or loading service schema and configuration data.

--deleteservice (-r)

`--deleteservice` is an option for deleting a service and its schema only.

--serviceName

`--serviceName` is an option that takes a value equal to the service name which is defined under the `Service name=...` tag of an XML service file. This portion is displayed in Code Example 10-1 on page 237.

Code Example 10-1 Portion of `sampleMailService.xml`

```
...
<ServicesConfiguration>
    <Service name="sampleMailService" version="1.0">
        <Schema
 serviceHierarchy="/other.configuration/sampleMailService"
            i18nFileName="sampleMailService"
            i18nKey="iplanet-am-sample-mail-service-description">
...
```

--help (-h)

`--help` is an argument that displays the syntax for the `amadmin` command.

--version (-n)

`--version` is an argument that displays the utility name, product name, product version and legal notice.

Using amadmin for Federation Management

This section lists the parameters of amadmin for use with Federation Management. For more information on Federation Management, see the *Access Manager Federation Management Guide*.

Loading the Liberty meta compliance XML into Directory Server

```
amadmin -u|--runasdn <user's DN>

    -w|--password <password> or -f|--passwordfile <passwordfile>

    -e|--entityname <entity name>

    -g|--import <xmlfile>
```

--runasdn (-u)

The user's DN

--password (-w)

The user's password.

--passwordfile (-f)

The name of file that contains user's password.

--entityname (-e)

The entity name. For example, `http://www.example.com`. An entity should belong to only one organization.

--import (-g)

The name of an XML file that contains the meta information. This file should adhere to Liberty meta specification and XSD.

Exporting an Entity to an XML File (Without XML Digital Signing)

amadmin -u|--runasdn <user's DN>

```
-w|--password <password> or -f|--passwordfile <passwordfile>
    -e|--entityname <entity name>
    -o|--export <filename>
```

--runasdn (-u)

The user's DN

--password (-w)

The user's password.

--passwordfile (-f)

The name of file that contains user's password.

--entityname (--e)

The name of Entity that resides in the Directory Server

--export (-o)

The name of the file to contain the XML of the entity. XML shall be Liberty meta XSD compliance.

Exporting an Entity to an XML File (With XML Digital Signing)

```
amadmin -u|--runasdn <user's DN>
```

```
-w|--password <password> or -f|--passwordfile <passwordfile>
-e|--entityname <entity name>
-q|--exportwithsig <filename>
```

--runasdn (-u)

The user's DN

--password (-w)

The user's password.

--passwordfile (-f)

The name of file that contains user's password.

--entityname (--e)

The name of Entity that resides in the Directory Server

--exportwithsig (-o)

The name of the file to contain the XML of the entity. This file is digitally signed. The XML must be Liberty meta XSD compliant.

Using amadmin for Resource Bundles

The following section shows the `amadmin` syntax for adding, locating and removing resource bundles.

Add resource bundle.

```
amadmin -u|--runasdn <user-dn> -w|--password <user-password>
       -b|--addresourcebundle <name-of-resource-bundle>
       -i|--resourcebundlefilename <resource-bundle-file-name>
       [-R|--resourcelocale] <locale>
```

Get resource strings.

```
amadmin -u|--runasdn <user-dn> -w|--password <user-password>
       -z|--getresourcestrings <name-of-resource-bundle>
```

```
        [-R|--resourcelocale] <locale>
```

Remove resource bundle.

```
amadmin -u|--runasdn <user-dn> -w|--password <user-password>
        -j|--deleteresourcebundle <name-of-resource-bundle>
        [-R|--resourcelocale] <locale>
```

Chapter 11

The amserver Command Line Tool

This chapter provides information on the `amserver` command line tool. This chapter contains the following section:

- "The amserver Command Line Executable" on page 241

The amserver Command Line Executable

The `amserver` command line executable starts and stops the `amunixd` and `amsercuridd` helpers, associated with Unix and SecurID authentication modules, respectively.

amserver Syntax

The generic syntax for the tools is:

```
./amserver { start | stop }
```

start

`start` is a command that starts the helper.

stop

`stop` is a command that stops the helper.

Chapter 12

The am2bak Command Line Tool

This chapter provides information on the am2bak command line tool and contains the following section:

The am2bak Command Line Executable

Access Manager contains an am2bak utility under AccessManager-base/SUNWam/bin. This utility performs a backup of either all or optional components of Access Manager. Directory Server must be running while taking the log backup.

The am2bak Syntax

The generic syntax for using the am2bak tool for the Solaris operating system is:

```
./am2bak [ -v | --verbose ] [ -k | --backup backup-name ] [ -l |
--location location ] [[-c | --config] | [-b | --debug] | [-g | --log]
| [-t | --cert] | [-d | --ds] | [-a | --all]]*
```

```
./am2bak -h | --help
```

```
./am2bak -n | --version
```

The generic syntax for using the am2bak tool for the Windows 2000 operating system is:

```
am2bak [ -v | --verbose ] [ -k | --backup backup-name ] [ -l |
--location location ] [[-c | --config] | [-b | --debug] | [-g | --log]
| [-t | --cert] | [-d | --ds] | [-a | --all]]*
```

```
am2bak -h | --help
```

```
am2bak -n | --version
```

NOTE Two hyphens must be entered exactly as shown in the syntax.

am2bak Options

--verbose (-v)

`--verbose` is used to run the backup utility in verbose mode.

--backup backup-name (-k)

`--backup` *backup-name* defines the name of the backup file. The default is `ambak`.

--location (-l)

`--location` specifies the directory location of the backup. The default location is `AccessManager-base/backup`.

--config (-c)

`--config` specifies backup only for configuration files.

--debug (-b)

`--debug` specifies backup only for debug files.

--log (-g)

`--log` specifies backup only for log files.

--cert (-t)

`--cert` specifies backup only for certificate database files.

--ds (-d)

`--ds` specifies backup only for the Directory Server.

--all (-a)

`--all` specifies a complete backup of the entire Access Manager.

--help (-h)

`--help` is an argument that displays the syntax for the `am2bak` command.

--version (-n)

`--version` is an argument that displays the utility name, product name, product version and legal notice.

Backup Procedure

1. Login as root.

   The user running this script must have root access.

2. Run the script ensuring that the correct path is used, if necessary.

   The script will backup the following Solaris™ Operating Environment files:

   - Configuration and Customization Files:
     - *AcessManager-base*`/SUNWam/config/`
     - *AcessManager-base*`/SUNWam/locale/`
     - *AcessManager-base*`/SUNWam/servers/httpacl`
     - *AcessManager-base*`/SUNWam/lib/*.properties` (Java property files)
     - *AcessManager-base*`/SUNWam/bin/amserver.`*instance-name*
     - *AcessManager-base*`/SUNWam/servers/https-`*all_instances*
     - *AcessManager-base*`/SUNWam/servers/web-apps-`*all_instances*
     - *AcessManager-base*`/SUNWam/web-apps/services/WEB-INF/config`
     - *AcessManager-base*`/SUNWam/web-apps/services/config`
     - *AcessManager-base*`/SUNWam/web-apps/applications/WEB-INF/classes`
     - *AcessManager-base*`/SUNWam/web-apps/applications/console`
     - `/etc/rc3.d/K55amserver.`*all_instances*
     - `/etc/rc3.d/S55amserver.`*all_instances*
     - *DirectoryServer_base*`/slapd-`*host*`/config/schema/`
     - *DirectoryServer_base*`/slapd-`*host*`/config/slapd-collations.conf`
     - *DirectoryServer_base*`/slapd-`*host*`/config/dse.ldif`
   - Log And Debug Files:
     - `var/opt/SUNWam/logs` (Access Manager log files)
     - `var/opt/SUNWam/install` (Access Manager installation log files)

- `var/opt/SUNWam/debug` (Access Manager debug files)

❍ Certificates:

- *AcessManager-base*`/SUNWam/servers/alias`
- *DirectoryServer_base*`/alias`

The script will also backup the following Microsoft® Windows 2000 operating system files:

❍ Configuration and Customization Files:

- *AcessManager-base*`/web-apps/services/WEB-INF/config/*`
- *AcessManager-base*`/locale/*`
- *AcessManager-base*`/web-apps/applications/WEB-INF/classes/*.properties` (java property files)
- *AcessManager-base*`/servers/https-`*host*`/config/jvm12.conf`
- *AcessManager-base*`/servers/https-`*host*`/config/magnus.conf`
- *AcessManager-base*`/servers/https-`*host*`/config/obj.conf`
- *DirectoryServer_base*`/slapd-host/config/schema/*.ldif`
- *DirectoryServer_base*`/slapd-host/config/slapd-collations.conf`
- *DirectoryServer_base*`/slapd-host/config/dse.ldif`

❍ Log And Debug Files:

- `var/opt/logs` (Access Manager log files)
- `var/opt/debug` (Access Manager debug files)

❍ Certificates:

- *AcessManager-base*`/servers/alias`
- *AcessManager-base*`/alias`

Chapter 13

The bak2am Command Line Tool

This chapter provides information on the bak2am command line tool and contains the following section:

The bak2am Command Line Executable

Access Manager contains an bak2am utility under AccessManager-base/SUNWam/bin. This utility performs a restore of the Access Manager components that were backed-up by the am2back utility.

The bak2am Syntax

The generic syntax for using the bak2am tool for the Solaris operating system is:

```
./bak2am [ -v | --verbose ] -z | --gzip tar.gz-file
./bak2am [ -v | --verbose ] -t | --tar tar-file
./bak2am -h | --help
./bak2am -n | --version
```

The generic syntax for using the bak2am tool for the Windows 2000 operating system is:

```
bak2am [ -v | --verbose ] -d | --directory directory-name
bak2am -h | --help
bak2am -n | --version
```

NOTE Two hyphens must be entered exactly as shown in the syntax.

bak2am Options

--gzip backup-name

`--gzip` specifies the full path and filename of the backup file in `tar.gz` format. By default, the path is `AccessManager-base/backup`. This option is for Solaris only.

--tar backup-name

`--tar` specifies the full path and filename of the backup file in `tar` format. By default, the path is `AccessManager-base/backup`. This option is for Solaris only.

--verbose

`--verbose` is used to run the backup utility in verbose mode.

--directory

`--directory` specifies the backup directory. By default, the path is `AccessManager-base/backup`. This option is for Windows 2000 only.

--help

`--help` is an argument that displays the syntax for the `bak2am` command.

--version

`--version` is an argument that displays the utility name, product name, product version and legal notice.

1. Login as root.

   The user running this script must have root access.

2. Untar the input tar file.

   This was generated when the backup script was run.

Chapter 14

The ampassword Command Line Tool

This chapter provides information on the amPassword command line tool and contains the following sections:

The ampassword Command Line Executable

Access Manager contains an ampassword utility under /opt/SUNWam/bin on SPARC systems and /opt/sun/Identity/bin on Linux systems.. This utility allows you change the Access Manager password for the administrator or user.

The ampassword Syntax

The generic syntax for using the ampassword tool is:

```
ampassword -a | --admin [ -o | --old oldPassword -n | --new newPassword ]
```

```
ampassword -p | --proxy [ -o | --old oldPassword -n | --new newPassword ]
```

```
ampassword -e | --encrypt [ password ]
```

NOTE Two hyphens must be entered exactly as shown in the syntax.

ampassword Options

--admin (-a)

--admin is used to change the admin password.

--proxy (-p)

`--proxy` is used to change the proxy password. It corresponds to the proxy user (user type `proxy` in `serverconfig.xml`.)

--encrypt (-e)

`--encrypt` is used to encrypt the password. It is printed to the command line. For example, to encrypt a new `dsamuser` password, use the following command:

```
ampassword -e newPassword
```

Then, place the new `dsamuser` password in `serverconfig.xml` and restart the web container (Web Server or Application Server).

Running ampassword on SSL

To run ampassword with Access Manager running in Secure-Socket Layer (SSL) mode:

1. Modify the `serverconfig.xml` file, located in the following directory:

   `AccessManager-base/SUNWam/config/`

2. Change `port` the server attribute to the SSL port which Access Manager is running.

3. Change the `type` attribute to SSL.

   For example:

```
<iPlanetDataAccessLayer>

<ServerGroup name="default" minConnPool="1" maxConnPool="10">

    <Server name="Server1" host="sun.com" port="636" type="SSL" />

    <User name="User1" type="proxy">
```

```
        <DirDN>

                 cn=puser,ou=DSAME Users,dc=iplanet,dc=com

        </DirDN>

        <DirPassword>

                  AQIC5wM2LY4Sfcy+AQBQxghVwhBE92i78cqf

         </DirPassword>

</User> ...
```

ampassword only changes the password in Directory Server. You will have to manually change passwords in the `ServerConfig.xml` and all authentication templates for Access Manager.

Chapter 15

The VerifyArchive Command Line Tool

This chapter provides information on the `VerifyArchive` command line tool and contains the following section:

The VerifyArchive Command Line Executable

The purpose of `VerifyArchive` is to verify the log archives. A log archive is a set of timestamped logs and their corresponding key stores (keystores contain the keys used to generate the MACs and the Digital Signatures which are used to detect tampering of the log files). Verification of an archive detects possible tampering and / or deletion of any file in the archive.

`VerifyArchive` extracts all of the archive sets, and all files belonging to each archive set, for a given `logName`. When executed, `VerifyArchive` searches each log record to for tampering If tampering is detected, it prints a message specifying which file and the number of the record that has been tampered with..

`VerifyArchive` also checks for any files that have been deleted from the archive set. If a deleted file is detected, it prints a message explaining that verification has failed. If no tampering or deleted files are detected, it returns a message explaining that the archive verification has been successfully completed.

NOTE An error may occur if you run `amverifyarchive` as a user without administrator privileges.

VerifyArchive Syntax

All of the parameters options are required. The syntax is as follows:

`amverifyarchive -l logName -p` *path* `-u` *uname* `-w` *password*

VerifyArchive Options

logName

`logName` refers to the name of the log which is to be verified (such as, `amConsole`, `amAuthentication` and so forth..). `VerifyArchive` verifies the both the access and error logs for the given logName. For example, if `amConsole` is specified, the verifier verifies the `amConsole.access` and `amConsole.error` files. Alternatively, the `logName` can be specified as `amConsole.access` or `amConsole.error` to restrict the verification of those logs only.

path

`path` is the full directory path where the log files are stored.

uname

`uname` is the user id of the Access Manager administrator.

password

`password` is the password of the Access Manager adminstrator.

Chapter 16

The amsecuridd Helper

This chapter provides information on the `amsecuiridd` helper and contains the following section:

The amsecuridd Helper Command Line Executable

The Access Manager SecurID authentication module is implemented using the Security Dynamic ACE/Client C API and the `amsecruidd` helper, which communicates between the Access Manager SecurID authentication module and the SecurID Server. The SecurID authentication module invokes the `amsecuridd` daemon by opening a socket to `localhost:57943` to listen for SecurID authentication requests.

NOTE 57943 is the default port number. If this port number is already used, you can specify a different port number in the SecurID Helper Authentication Port attribute in the SecurID Authentication module. This port number must be unique accross all organizations.

Because the interface to `amsecuridd` is in clear text through `stdin`, only local host connections are permitted. `amsecuridd` uses the SecurID remote API (version 5.x) on the back end for data encryption.

The amsecuridd helper listens on port number 58943 (by default) to receive its configuration information. If this port is already used, you can change it in the securidHelper.ports attribute in the AMConfig.properties file (by default, located in *AccessManager-base*/SUNWam/config/). The securidHelp.ports attribute contains a space-separated list of the ports for each amsecuridd helper instance. Restart Identity Sever once the changes to AMConfig.properties are saved.

NOTE A separate instance of amsecuridd should run for each organization that communicates with a separate ACE/Server (containing different sdconf.rec files).

amsecuridd Syntax

The syntax is as follows:

```
amsecuridd [-v] [-c portnum]
```

amsecuridd Options

verbose (-v)

Turns on verbose mode and logs to /var/opt/SUNWam/debug/securidd_client.debug.

configure portnumber (-c portnm)

Configures the listening port number. The default is 58943.

Running the amsecuridd helper

amsecuridd is located, by default, in *AccessManager-base*/SUNWam/share/bin. To run the helper on the default ports, enter the following command (without options):

```
./amsecuridd
```

To run the helper on non-default port, enter the following command:

```
./amsecuridd [-v] [-c portnm]
```

amsecuridd can also be run through the amserver command line utitility, but it will only run on the default ports.

Required Libraries

In order to run the helper, the following libraries are required (most can be found in the operating system in `/usr/lib/`):

- `libnsl.so.1`
- `libthread.so.1`
- `libc.so.1`
- `libdl.so.1`
- `libmp.so.2`
- `librt.so.1`
- `libaio.so.1`
- `libmd5.so.1`

NOTE Set `LD_LIBRARY_PATH` to *`AccessManager-base`*`/Sunwam/lib/` to find `libaceclnt.so`.

Part IV

Attribute Reference

This is the Attribute Reference, part four of the Sun Java System Acceess Manager Administration Guide. It discusses the configured attributes within Access Manager's default services. This part contains the following chapters:

Chapter 17

Administration Service Attributes

The Administration Service consists of global and organization attributes. The values applied to the global attributes are applied across the Sun Java System Acceess Manager configuration and are inherited by every configured organization. They can not be applied directly to roles or organizations as the goal of global attributes is to customize the Access Manager application. Values applied to the organization attributes are default values for each organization configured and can be changed when the service is registered to the organization. The organization attributes are not inherited by entries of the organization. The Administration Attributes are divided into:

Global Attributes

The global attributes in the Administration Service are:

Enable Federation Management

When selected, this field enables Federation Management. It is selected by default. To disable this feature, deselect the field The Federation Management Service tab will not appear in the console.

Enable User Management

When selected as True, this field enables User Management. This is enabled by default.

Show People Containers

This attribute specifies whether to display People Containers in the Access Manager console. If this option is selected, the menu choice People Containers displays in the View menu for Organizations, Containers and Group Containers. People Containers will be seen at the top-level only for a flat DIT.

People containers are organizational units containing user profiles. It is recommended that you use a single people container in your DIT and leverage the flexibility of roles to manage accounts and services. The default behavior of the Access Manager console is to hide the People Container. However, if you have multiple people containers in your DIT, select Show People Containers to display People Containers as managed objects in the Access Manager console.

Show Containers In View Menu

This attribute specifies whether to display any containers in the View menu of the Access Manager console. The default value is `false`. An administrator can optionally chose either:

- `false` (checkbox not selected) — Containers are not listed among the choices on the View menu at the top-level for organizations and other containers.
- `true` (checkbox selected) — Containers are listed among the choices on the View menu at the top-level and for organizations and other containers.

Show Group Containers

This attribute specifies whether to show Group Containers in the Access Manager console. If this option is selected, the menu choice Group Containers displays in the View menu for organizations, containers, and group containers. Group containers are organizational units for groups.

Managed Group Type

This option specifies whether subscription groups created through the console are static or dynamic. The console will either create and display subscription groups that are static or dynamic, not both. (Filtered groups are always supported regardless of the value given to this attribute.) The default value is dynamic.

- A static group explicitly lists each group member using the `groupOfNames` or `groupOfUniqueNames` object class. The group entry contains the `uniqueMember` attribute for each member of the group. Members of static groups are manually added; the user entry itself remains unchanged. Static groups are suitable for groups with few members.
- A dynamic group uses a `memberOf` attribute in the entry of each group member. Members of dynamic groups are generated through the use of an LDAP filter which searches and returns all entries which contain the `memberOf` attribute. Dynamic groups are suitable for groups that have a very large membership.

- A filtered group uses an LDAP filter to search and return members that meet the requirement of the filter. For instance, the filter can generate members with a specific uid (`uid=g*`) or email address (`mail=*@sun.com`). In these examples, the LDAP filter would return all users whose uid begins with `g` or whose email address ends with `sun.com`, respectively. Filtered groups can only be created within the User Management view by choosing Membership by Filter.

An administrator can select one of the following:

- `Dynamic` — Groups created through the Membership By Subscription option will be dynamic.
- `Static` — Groups created through the Membership By Subscription option will be static.

Default Role Permissions

This attribute defines a list of default access control instructions (ACIs) or *permissions* that are used to grant administrator privileges when creating new roles. One of these ACIs is selected depending on the level of privilege desired. Access Manager ships with four default role permissions:

No Permissions

No permissions are to be set on the role.

Organization Admin

The Organization Administrator has read and write access to all entries in the configured organization.

Organization Help Desk Admin

The Organization Help Desk Administrator has read access to all entries in the configured organization and write access to the `userPassword` attribute.

Organization Policy Admin

The Organization Policy Administrator has read and write access to all policies in the organization. The Organization Policy Administrator can not create a referral policy to a peer organization.

NOTE Roles are defined using the format `aci_name | aci_desc | dn:aci ## dn:aci ## dn:aci` where:

- *aci_name* is the name of the ACI.
- *aci_desc* is a description of the access these ACIs allow. For maximum usability, assume the reader of this description does not understand ACIs or other directory concepts.

aci_name and *aci_desc* are i18n keys contained in the `amAdminUserMsgs.properties` file. The values displayed in the console come from the `.properties` file, and the keys are used to retrieve those values.

- *dn:aci* represents pairs of DNs and ACIs separated by ##. Access Manager sets each ACI in the associated DN entry. This format also supports tags that can be substituted for values that would otherwise have to be specified literally in an ACI: ROLENAME, ORGANIZATION, GROUPNAME and PCNAME. Using these tags lets you define roles flexible enough to be used as defaults. When a role is created based on one of the default roles, tags in the ACI resolve to values taken from the DN of the new role.

Enable Domain Component Tree

The Domain Component tree (DC tree) is a specific DIT structure used by many Sun Java System components to map between DNS names and organizations' entries.

When this option is enabled, the DC tree entry for an organization is created, provided that the DNS name of the organization is entered at the time the organization is created. The DNS name field will appear in the Organization Create page. This option is only applicable to top-level organizations, and will not be displayed for suborganizations.

Any status change made to the `inetdomainstatus` attribute through the Access Manager SDK in the organization tree will update the corresponding DC tree entry status. (Updates to status that are not made through the Access Manager SDK will not be synchronized.) For example, if a new organization, `sun`, is created with the DNS name attribute sun.com, the following entry will be created in the DC tree:

`dc=sun,dc=com,o=internet,`*root suffix*

The DC tree may optionally have its own root suffix configured by setting `com.iplanet.am.domaincomponent` in `AMConfig.properties`. By default, this is set to the Access Manager root. If a different suffix is desired, this suffix must be created using LDAP commands. The ACIs for administrators that create organizations required modification so that they have unrestricted access to the new DC tree root.

Enable Administrative Groups

This option specifies whether to create the DomainAdministrators and DomainHelpDeskAdministrators groups. If selected (`true`), these groups are created and associated with the Organization Admin Role and Organization Help Desk Admin Role, respectively. Once created, adding or removing a user to one of these associated roles automatically adds or removes the user from the corresponding group. This behavior, however, does not work in reverse. Adding or removing a user to one of these groups will not add or remove the user in the user's associated roles.

The DomainAdministrators and DomainHelpDeskAdministrators groups are only created in organizations that are created after this option is enabled.

NOTE This option does not apply to suborganizations, with the exception of the `root org`. At the `root org`, the ServiceAdministrators and ServiceHelpDesk Administrators groups are created and associated with the Top-level Admin and Top-level Help Desk Admin roles, respectively. The same behavior applies.

Enable Compliance User Deletion

This option specifies whether a user's entry will be deleted, or just marked as deleted, from the directory. When a user's entry is deleted and this option is selected (`true`), the user's entry will still exist in the directory, but will be marked as deleted. User entries that are marked for deletion are not returned during Directory Server searches. If this option is not selected, the user's entry will be deleted from the directory.

Dynamic Administrative Roles ACIs

This attribute defines the access control instructions for the administrator roles that are created dynamically when a group or organization is configured using Access Manager. These roles are used for granting administrative privileges for the specific grouping of entries created. The default ACIs can be modified only under this attribute listing.

CAUTION Administrators at the Organization level have a wider scope of access than do group administrators. But, by default, when a user is added to a group administrator role, that user can change the password of anyone in the group. This would include any organization administrator who is a member of that group.

Container Help Desk Admin

The Container Help Desk Admin role has read access to all entries in an organizational unit and write access to the `userPassword` attribute in user entries only in this container unit.

Organization Help Desk Admin

The Organization Help Desk Administrator has read access to all entries in an organization and write access to the `userPassword` attribute.

NOTE When a suborganization is created, remember that the administration roles are created in the suborganization, not in the parent organization.

Container Admin

The Container Admin role has read and write access to all entries in an LDAP organizational unit. In Access Manager, the LDAP organizational unit is often referred to as a container.

Organization Policy Admin

The Organization Policy Administrator has read and write access to all policies, and can create, assign, modify, and delete all policies within that organization.

People Container Admin

By default, any user entry in an newly created organization is a member of that organization's People Container. The People Container Administrator has read and write access to all user entries in the organization's People Container. Keep in mind that this role DOES NOT have read and write access to the attributes that contain role and group DNs therefore, they cannot modify the attributes of, or remove a user from, a role or a group.

NOTE Other containers can be configured with Access Manager to hold user entries, group entries or even other containers. To apply an Administrator role to a container created after the organization has already been configured, the Container Admin Role or Container Help Desk Admin defaults would be used.

Group Admin

The Group Administrator has read and write access to all members of a specific group, and can create new users, assign users to the groups they manage, and delete the users the that they have created.

When a group is created, the Group Administrator role is automatically generated with the necessary privileges to manage the group. The role is not automatically assigned to a group member. It must be assigned by the group's creator, or anyone that has access to the Group Administrator Role.

Top-level Admin

The Top-level Administrator has read and write access to all entries in the top-level organization. In other words, this Top-level Admin role has privileges for every configuration principal within the Access Manager application.

Organization Admin

The Organization Administrator has read and write access to all entries in an organization. When an organization is created, the Organization Admin role is automatically generated with the necessary privileges to manage the organization.

User Profile Service Classes

This attribute lists the services that will have a custom display in the User Profile page. The default display generated by the console may not be sufficient for some services. This attribute creates a custom display for any service, giving full control over what and how the service information is displayed. The syntax is as follows:

Service name | *relative url*

NOTE Services that are listed in this attribute will not display in the User Create pages. Any data configuration for a custom service display must be performed the User Profile pages.

DC Node Attribute List

This field defines the set of attributes that will be set in the DC tree entry when an object is created. The default parameters are:

- maildomainwelcomemessage
- preferredmailhost
- mailclientattachmentquota
- mailroutingsmarthost

- mailroutingsmarthost
- mailroutingsmarthost
- mailaccessproxyreplay
- preferredlanguage
- domainuidseparator
- maildomainmsgquota
- maildomainallowedserviceaccess
- preferredmailmessagestore
- maildomaindiskquota
- maildomaindiskquota
- objectclass=maildomain
- mailroutinghosts

Search Filters for Deleted Objects

This field defines the search filters for objects to be removed when User Compliance Deletion mode is enabled.

Default People Container

This attribute specifies the default people container into which the user is created.

Default Groups Container

This attribute specifies the default groups container into which the group is created.

Default Agents Container

This attribute specifies the default agent container into which the agent is created.

Organization Attributes

The organization attributes in the administration service are:

Groups Default People Container

This field specifies the default People Container where users will be placed when they are created. There is no default value. A valid value is the DN of a people container. See the note under Groups People Container List attribute for the People Container fallback order.

Groups People Container List

This field specifies a list of People Containers from which a Group Administrator can choose when creating a new user. This list can be used if there are multiple People Containers in the directory tree. (If no People Containers are specified in this list or in the Groups Default People Container field, users are created in the default Access Manager people container, `ou=people`.) There is no default value for this field. The syntax for this attribute is as follows:

dn of group | *dn of people container*

NOTE When a user is created, this attribute is checked for a container in which to place the entry. If the attribute is empty, the Groups Default People Container attribute is checked for a container. If the latter attribute is empty, the entry is created under `ou=people`.

User Profile Display Class

This attribute specifies the Java class used by the Access Manager console when it displays the User Profile pages.

End User Profile Display Class

This attribute specifies the Java class used by the Access Manager console when it displays the End User Profile pages.

Show Roles on User Profile Page

This option specifies whether to display a list of roles assigned to a user as part of the user's User Profile page. If the value is `false` (not selected), the User Profile page shows the user's roles only for administrators. The default value is `false`.

Show Groups on User Profile Page

This option specifies whether to display a list of groups assigned to a user as part of the user's User Profile page. If the value is `false` (not selected), the User Profile page shows the user's groups only for administrators. The default value is `false`.

Enable User Self Subscription to Group

This option specifies whether users can add themselves to groups that are open to subscription. If the value is false, the user profile page allows the user's group membership to be modified only by an administrator. The default value is `false`.

NOTE This option applies only when the Show Groups on User Profile Page option is selected.

User Profile Display Options

This menu specifies which service attributes will be displayed in the user profile page. An administrator can select from the following:

- `UserOnly` — Display viewable User schema attributes for services assigned to the user.

  User service attribute values are viewable by the user when the attribute contains the keyword Display. See the *Access Manager Developer's Guide* for details.

- `Combined` — Display viewable User and Dynamic schema attributes for services assigned to the user.

User Creation Default Roles

This listing defines roles that will be assigned to newly created users automatically. There is no default value. An administrator can input the DN of one or more roles.

NOTE This field only takes a full Distinguished Name address, not a role name. The roles can only be Access Manager roles, not LDAP (Directory Server) roles.

Administrative Console Tabs

This field lists the Java classes of modules that will be displayed at the top of the console. The syntax is `i18N key | java class name.` (The i18N key is used for the localized name of the entry in the View menu.)

Maximum Results Returned From Search

This field defines the maximum number of results returned from a search. The default value is 100.

CAUTION Use caution when setting this attribute to large value. For sizing limits, see the *Sun Java System Directory Server Installation and Tuning Guide* at the following location:

`http://docs.sun.com/db/doc/816-6697-10`

Modifications to this attribute done through `LDAPModify` will take precedence to those made through the Access Manager Console. For more information on changing this attribute using `LDAPModify`, see the *Access Manager Developer's Guide*.

Timeout For Search

This field defines the amount of time (in number of seconds) that a search will continue before timing out. It is used to stop potentially long searches. After the maximum search time is reached, an error is returned. The default is 5 seconds.

JSP Directory Name

This field specifies the name of the directory that contains the `.jsp` files used to construct the console, to give an organization a different appearance (customization). The `.jsp` files need to be copied into the directory that is specified in this field.

Online Help Documents

This field lists the online help links that will be created on the main Access Manager help page. This allows other applications to add their online help links in the Access Manager page. The format for this attribute is as follows:

linki18nkey | *html page to load when clicked* | *i18n properties file* | *remote server*

NOTE *remote server* is optional argument that allows you to specify the remote server on which the online help document is located.

For example:

```
IdentityServer Help | /AMAdminHelp.html | amAdminModuleMsgs
```

Required Services

This field lists the services that are dynamically added to the users' entries when they are created. Administrators can choose which services are added at the time of creation.

This attribute is not used by the console, but by the Access Manager SDK. Users that are dynamically created and created by the `amadmin` command line utility will be assigned the services listed in this attribute.

User Search Key

This attribute defines the attribute name that is to be searched upon when performing a simple search in the Navigation page. The default value for this attribute is `cn`. For example, if this attribute uses the default:

If you enter `j*` in the Name field in the Navigation frame, users whose names begins with "j" or "J" will be displayed.

User Search Return Attribute

This field defines the attribute name used when displaying the users returned from a simple search. The default of this attribute is `uid cn`. This will display the user ID and the user's full name.

The attribute name that is listed first is also used as the key for sorting the set of users that will be returned. To avoid performance degradation, use an attribute whose value is set in a user's entry.

User Creation Notification List

This field defines a list of email addresses that will be sent notification when a new user is created. Multiple email addresses can be specified, as in the following syntax:

e-mail|*locale*|*charset*

e-mail|*locale*|*charset*

e-mail|*locale*|*charset*

The notification list also accepts different locales by using the |locale option. For example, to send the notification to an administrator in France:

someuser@example.com|fr|fr

See Table 21-1 on page 301 for a list of locales.

NOTE The sender email ID can be changed by modifying property 497 in amProfile.properties, which is located, by default, at AccessManager-base/SUNWam/locale.

User Deletion Notification List

This field defines a list of email addresses that will be sent notification when a user is deleted. Multiple email addresses can be specified, as in the following syntax:

e-mail|*locale*|*charset*

e-mail|*locale*|*charset*

e-mail|*locale*|*charset*

The notification list also accepts different locales by using the |locale option. For example, to send the notification to an administrator in France:

someuser@example.com|fr|fr

See Table 21-1 on page 301 for a list of locales.

NOTE The sender email ID can be changed by modifying property 497 in amProfile.properties, which is located, by default, at AccessManager-base/SUNWam/locale. The default sender ID is DSAME.

User Modification Notification List

This field defines a list of attributes and email addresses associated with the attribute. When a user modification occurs on an attribute defined in the list, the email address associated with the attribute will be sent notification. Each attribute can have a different set of addresses associated to it. Multiple email address can be specified, as in the following syntax:

```
attrName e-mail|locale|charset e-mail|locale|charset .....
```

```
attrName e-mail|locale|charset e-mail|locale|charset .....
```

The `self` keyword may be used in place of one of the addresses. This sends mail to the user whose profile was modified.

For example:

```
manager someuser@sun.com|self|admin@sun.com
```

Mail will be sent to the address specified in the `manager` attribute, `someuser@sun.com`, `admin@sun`, the person who modified the user (`self`).

The notification list also accepts different locales by using the `|locale` option. For example, to send the notification to an administrator in France:

```
manager someuser@sun.com|self|admin@sun.com|fr
```

See Table 21-1 on page 301 for a list of locales.

NOTE The attribute name is the same as it appears in the Directory Server schema, and not as the display name in the console.

Maximum Entries Displayed per Page

This attribute allows you to define the maximum rows that can be displayed per page. The default is 25. For example, if a user search returns 100 rows, there will be 4 pages with 25 rows displayed in each page.

Event Listener Classes

This attribute contains a list of listeners that receive creation, modification and deletion events from the Access Manager console.

Pre and Post Processing Classes

This field defines a list of implementation classes through plug-ins that extend the `com.iplanet.am.sdk.AMCallBack` class to receive callbacks during pre and post processing operations for users, organization, roles and groups. The operations are:

- create
- delete
- modify
- add users to roles / groups
- delete users from roles / groups

You must enter the full class name of the plug-in. For example:

```
com.iplanet.am.sdk.AMCallbacSample
```

You must then change the class path of your web container (from the Access Manager installation base) to include the full path to the location of the plug-in class.

Enable External Attributes Fetch

This option enables callbacks for plug-ins to retrieve external attributes (any external application-specific attribute). External attributes are not cached in the Access Manager SDK, so this attribute allows you enable attribute retrieval per organization level. By default, this option is not enabled.

Invalid User ID Characters

This attribute defines a list of characters that are not allowed in a user's name.

Each character must be separated by the | character. For example:

```
*|(|)|&|!
```

UserID and Password Validation Plugin Class

This class provides a userID and password validation plugin mechanism.

The methods of this class need to be overridden by the implementation plugin modules that validate the userID and/or password for the user. The implementation plugin modules will be invoked whenever a userID or password value is being added or modified using the Access Manager console, the `amadmin` command line interface, or using the SDK.

The plugins that extend this class can be configured per organization. If a plugin is not configured for an organization, then the plugin configured at the global level will be used.

If the validation of the plugin fails, the plugin module can throw an exception to notify the application to indicate the error in the userID or password supplied by the user.

Chapter 18

Active Directory Authentication Attributes

The Active Directory Authentication attributes are organization attributes. The values applied to them under Service Configuration become the default values for the Active Directory Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the organization. The Active Directory Authentication attributes are:

- “User Creation Attributes List” on page 284
- “Authentication Level” on page 284

Primary Active Directory Server

This field specifies the host name and port number of the primary Active Directory server specified during Access Manager installation. This is the first server contacted for Active Directory authentication. The format is `hostname:port`. (If there is no port number, assume 389.).

If you have Access Manager deployed with multiple domains, you can specify the communication link between specific instances of Access Manager and Directory Server in the following format (multiple entries must be prefixed by the local server name):

```
local_servername|server:port local_servername2|server2:port2 ...
```

For example, if you have two Access Manager instances deployed in different locations (`L1-machine1-IS` and `L2- machine2-IS`) communicating with different instances of Directory Server (`L1-machine1-DS` and `L2-machine2-DS`), it would look the following:

```
L1-machine1-IS.example.com|L1-machine1-DS.example.com:389
L2-machine2-IS.example.com|L2-machine2-DS.example.com:389
```

Secondary Active Directory Server

This field specifies the host name and port number of a secondary Active Directory server available to the Access Manager platform. If the primary Active Directory server does not respond to a request for authentication, this server would then be contacted. If the primary server is up, Access Manager will switch back to the primary server. The format is also `hostname:port`. Multiple entries must be prefixed by the local server name.

CAUTION When authenticating users from a Directory Server that is remote from the Access Manager enterprise, it is important that both the Primary and Secondary Active Directory Server Ports have values. The value for one Directory Server location can be used for both fields.

DN to Start User Search

This field specifies the DN of the node where the search for a user would start. (For performance reasons, this DN should be as specific as possible.) The default value is the root of the directory tree. Any valid DN will be recognized. If `OBJECT` is selected in the Search Scope attribute, the DN should specify one level above the level in which the profile exists.

Multiple entries must be prefixed by the local server name. The format is as follows:

`servername|search dn`

For multiple entries

`servername1|search dn servername2|search dn servername3|search dn...`

If multiple users are found for the same search, authentication will fail.

DN for Root User Bind

This field specifies the DN of the user that will be used to bind to the Directory Server specified in the Primary Active Directory Server and Port field as administrator. The authentication service needs to bind as this DN in order to search for a matching user DN based on the user login ID. The default value is `amLDAPuser`. Any valid DN will be recognized.

Make sure that password is correct before you logout, because if it is incorrect, you will be locked out. If this should occur, you can login with the super user DN in the `com.iplanet.authentication.super.user` property in the `AMConfig.Properties` file. By default, this the amAdmin account with which you would normally log in, although you will use the full DN. For example:

`uid_amAdmin,ou=People,`*`AccessManager-base`*

Password for Root User Bind

This field carries the password for the administrator profile specified in the DN for Root User Bind field. There is no default value. Only the administrator's valid Active Directory password will be recognized.

Password For Root User Bind (Confirm)

Confirmation of the password.

Active Directory Attribute Used to Retrieve User Profile

After successful authentication by a user, the user's profile is retrieved. The value of this attribute is used to perform the search. The field specifies the Active Directory attribute to use. By default, Access Manager assumes that user entries are identified by the `uid` attribute. If your Directory Server uses a different attribute (such as `givenname`)specify the attribute name in this field.

NOTE The user search filter will be a combination of the Search Filter attribute and the Active Directory Attribute Used to Retrieve User Profile.

Active Directory Attributes Used to Search for a User to be Authenticated

This field lists the attributes to be used to form the search filter for a user that is to be authenticated, and allows the user to authenticate with more than one attribute in the user's entry. For example, if this field is set to `uid`, `employeenumber` and `mail`, the user could authenticate with any of these names.

User Search Filter

This field specifies an attribute to be used to find the user under the DN to Start User Search field. It works with the User Entry Naming Attribute. There is no default value. Any valid user entry attribute will be recognized.

Search Scope

This menu indicates the number of levels in the Directory Server that will be searched for a matching user profile. The search begins from the node specified in the attribute "DN to Start User Search" on page 281. The default value is SUBTREE. One of the following choices can be selected from the list:

- OBJECT - Searches only the specified node
- ONELEVEL - Searches at the level of the specified node and one level down
- SUBTREE - Search all entries at and below the specified node

CAUTION Users from suborganizations may be able to login even if the sub organization's status is inactive. To avoid this, make sure that the Search Scope and the Base DN are set to the specific organization to which the user belongs.

Enable SSL Access to Active Directory Server

This option enables SSL access to the Directory Server specified in the Primary and Secondary Active Directory Server and Port field. By default, this is not enabled and the SSL protocol will not be used to access the Directory Server. However, if this attribute is enabled, you can bind to a non-SSL server.

If the LDAP Server is running with SSL enabled (LDAPS), you must make sure that Access Manager is configured with proper SSL trusted certificates so that AM could connect to Directory server over LDAPS protocol

Return User DN To Authenticate

When the Access Manager directory is the same as the directory configured for Active Directory, this option may be enabled. If enabled, this option allows the Active Directory authentication module to return the DN instead of the `userId`, and no search is necessary. Normally, an authentication module returns only the `userId`, and the authentication service searches for the user in the local Access Manager Active Directory. If an external Active Directory directory is used, this option is typically not enabled.

Active Directory Server Check Interval

This attribute is used for Active Directory Server failback. It defines the number of minutes in which a thread will "sleep" before verifying that the Active Directory primary server is running.

User Creation Attributes List

This attribute is used by the Active Directory authentication module when the Active Directory server is configured as an external Active Directory server. It contains a mapping of attributes between a local and an external Directory Server. This attribute has the following format:

```
attr1|externalattr1
attr2|externalattr2
```

When this attribute is populated, the values of the external attributes are read from the external Directory Server and are set for the internal Directory Server attributes. The values of the external attributes are set in the internal attributes only when the User Profile attribute (in the Core Authentication module) is set to “Dynamically Created” and the user does not exist in local Directory Server instance. The newly created user will contain the values for internal attributes, as specified in User Creation Attributes List, with the external attribute values to which they map.

Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is `0`.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See “Default Authentication Level” on page 306 for details.

Chapter 19

Anonymous Authentication Attributes

The Anonymous Authentication attributes are organization attributes. The values applied to them under Service Configuration become the default values for the Anonymous Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the subtrees of the organization. The Anonymous Authentication attributes are:

Valid Anonymous User List

This field contains a list of user IDs that have permission to login without providing credentials. If a user's login name matches a user ID in this list, access is granted and the session is assigned to the specified user ID.

If this list is empty, accessing the following default module login URL will be authenticated as the Default Anonymous User Name:

```
protocol://server_host.server_domain:server_port/server_deploy_uri/UI/Login?module=Anonymous&org=org_name
```

If this list is not empty, accessing Default module login URL (same as above) will prompt the user to enter any valid Anonymous user name

If this list is not empty, the user can log in without seeing the login page by accessing the following URL:

protocol://server_host.server_domain:server_port/server_deploy_uri/UI/Login?module=Anonymous&org=org_name&IDToken1=<valid Anonymous username>

Default Anonymous User Name

This field defines the user ID that a session is assigned to if Valid Anonymous User List is empty and the following Default module login URL is accessed:

protocol://server_host.server_domain:server_port/server_deploy_uri/UI/Login?module=Anonymous&org=org_name

The default value is anonymous. An Anonymous user must also be created in the organization.

NOTE If Valid Anonymous User List is not empty, you can login without accessing the login page by using the user defined in Default Anonymous User Name. This can be done by accessing the following URL:

protocol://server_host.server_domain:server_port/server_deploy_uri/UI/Login?module=Anonymous&org=org_name&IDToken1=<DefaultAnonymous User Name>

Enable Case Sensitive User IDs

If enabled, this option allows for case-sensitivity for user IDs. By default, this attribute is not enabled.

Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is 0.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See “Default Authentication Level” on page 306 for details.

Chapter 20

Certificate Authentication Attributes

The Certificate Authentication attributes are organization attributes. The values applied to them under Service Configuration become the default values for the Certificate Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the subtrees of the organization. The Certificate Authentication attributes are:

- “Authentication Level” on page 294

Match Certificate in LDAP

This option specifies whether to check if the user certificate presented at login is stored in the LDAP Server. If no match is found, the user is denied access. If a match is found and no other validation is required, the user is granted access. The default is that the Certificate Authentication service does not check for the user certificate.

NOTE A certificate stored in the Directory Server is not necessarily valid; it may be on the certificate revocation list. See “Match Certificate to CRL” on page 290. However, the web container may check the validity of the user certificate presented at login.

Subject DN Attribute Used to Search LDAP for Certificates

This field specifies the attribute of the certificate’s `SubjectDN` value that will be used to search LDAP for certificates. This attribute must uniquely identify a user entry. The actual value will be used for the search. The default is `CN`.

Match Certificate to CRL

This option specifies whether to compare the user certificate against the Certificate Revocation List (CRL) in the LDAP Server. The CRL is located by one of the attribute names in the issuer’s `SubjectDN`. If the certificate is on the CRL, the user is denied access; if not, the user is allowed to proceed. This attribute is, by default, not enabled.

NOTE Certificates should be revoked when the owner of the certificate has changed status and no longer has the right to use the certificate or when the private key of a certificate owner has been compromised.

Issuer DN Attribute Used to Search LDAP for CRLs

This field specifies the attribute of the received certificate's issuer `subjectDN` value that will be used to search LDAP for CRLs. This field is used only when the Match Certificate to CRL attribute is enabled. The actual value will be used for the search. The default is `CN`.

HTTP Parameters for CRL Update

This fields specifies the HTTP parameters f for obtaining a CRL from a servlet for a CRL update. Contact the administrator of your CA for these parameters.

Enable OCSP Validation

This parameter enables OCSP validation to be performed by contacting the corresponding OCSP responder. The OCSP responder is decided as follows during runtime:

- If `com.sun.identity.authentication.ocspCheck` is true and the OCSP responder is set in the `com.sun.identity.authentication.ocsp.repsonder.url` attribute, the value of the attribute will be used as the OCSP responder.
- If `com.sun.identity.authentication.ocspCheck` is set to true and If the value of the attribute is not set in the `AMConfig.properties` file, the OCSP responder presented in your client certificate is used as the OCSP responder.

If `com.sun.identity.authentication.ocspCheck` is set to false or if `com.sum.identity.authentication.ocspCheck` is set to true and if an OCSP responder can not be found, no OCSP validation will be performed.

NOTE Before enabling OCSP Validation, make sure that the time of the Access Manager machine and the OCSP responder machine are in sync as close as possible. Also, the time on the Access Manager machine must not be behind the time on the OCSP responder. For example:

OCSP responder machine - 12:00:00 pm

Access Manager machine - 12:00:30 pm

LDAP Server Where Certificates Are Stored

This field specifies the name and port number of the LDAP server where the certificates are stored. The default value is the host name and port specified when Access Manager was installed. The host name and port of any LDAP Server where the certificates are stored can be used. The format is *hostname:port*.

LDAP Search Start DN

This field specifies the DN of the node where the search for the user's certificate should start. There is no default value. The field will recognize any valid DN. Multiple entries must be prefixed by the local server name. The format is as follows:

```
servername|search dn
```

For multiple entries

```
servername1|search dn servername2|search dn servername3|search dn...
```

If multiple users are found for the same search, authentication will fail.

LDAP Server Principal User

This field accepts the DN of the principal user for the LDAP server where the certificates are stored. There is no default value for this field which will recognize any valid DN. The principal user must be authorized to read, and search certificate information stored in the Directory Server.

LDAP Server Principal Password

This field carries the LDAP password associated with the user specified in the LDAP Server Principal User field. There is no default value for this field which will recognize the valid LDAP password for the specified principal user.

NOTE This value is stored as readable text in the directory.

LDAP Attribute for Profile ID

This field specifies the attribute in the Directory Server entry that matches the certificate whose value should be used to identify the correct user profile. There is no default value for this field which will recognize any valid attribute in a user entry (cn, sn, and so on) that can be used as the user ID.

Use SSL for LDAP Access

This option specifies whether to use SSL to access the LDAP server. The default is that the Certificate Authentication service does not use SSL for LDAP access.

Certificate Field Used to Access User Profile

This menu specifies which field in the certificate's Subject DN should be used to search for a matching user profile. For example, if you choose `email address`, the certificate authentication service will search for the user profile that matches the attribute `emailAddr` in the user certificate. The user logging in then uses the matched profile. The default field is `subject CN`. The list contains:

- email address
- subject CN
- subject DN
- subject UID
- other

Other Certificate Field Used to Access User Profile

If the value of the Certificate Field Used to Access User Profile attribute is set to `other`, then this field specifies the attribute that will be selected from the received certificate's `subjectDN` value. The authentication service will then search the user profile that matches the value of that attribute.

Trusted Remote Hosts

This attribute defines a list of trusted hosts that can be trusted to send certificates to Access Manager. Access Manager must verify whether the certificate emanated from one of these hosts. This configuration only used with Sun Java System Portal Server.

This attribute accepts the following values:

- **none.** This attribute is disabled. This is set by default.
- **any**. Accepts Portal Server Gateway-style certificate authentication from any client IP address.
- ***IP ADDR***. Lists the IP addresses from which to accept Portal Server Gateway-style certificate authentication requests (the IP Address of the Gateway(s)). The attribute is configurable on an organization basis.

SSL Port Number

This attribute specifies the port number for the secure socket layer. Currently, this attribute is only used by the Gateway servlet. Before you add or change an SSL Port Number, see the “Policy-Based Resource Management “section in Chapter 7 of the Access Manager Developer’s Guide.

Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is `0`.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See “Default Authentication Level” on page 306 for details.

Chapter 21

Core Authentication Attributes

The Core Authentication service is the basic service for all of the default authentication services as well as any custom authentication module attributes. Core authentication must be configured as a service for each organization that wishes to use any form of authentication. The Core Authentication attributes consist of global and organization attributes.The values applied to the global attributes are applied across the Sun Java System Acceess Manager configuration and are inherited by every configured organization. (They can not be applied directly to roles or organizations as the goal of global attributes is to customize the Access Manager application.) The values applied to the organization attributes under Service Configuration become the default values for the Core Authentication template. The service template needs to be created after adding the service for the organization. The default values can be changed after adding by the organization's administrator. Organization attributes are not inherited by entries in the organization. The Core Authentication attributes are separated into:

Global Attributes

The global attributes in the Core Authentication service are:

Pluggable Authentication Module Classes

This field specifies the Java classes of the authentication modules available to any organization configured within the Access Manager platform. By default, this includes LDAP, SafeWord, SecurID, Application, Anonymous, HTTP Basic, Membership, Unix, Certificate, NT, RADIUS and Windows Desktop SSO. You can write custom authentication modules by implementing the AMLoginModule SPI or the JAAS LoginModule SPI. For more information, see the Access Manager Developer's Guide. To define new services, this field must take a text string specifying the full class name (including package name) of each new authentication service.

Supported Authentication Modules for Clients

This attribute specifies a list of supported authentication modules for a specific client. The format is as follows:

```
clientType | module1,module2,module3
```

This attribute is in effect when Client Detection is enabled.

LDAP Connection Pool Size

This attribute specifies the minimum and maximum connection pool to be used on a specific LDAP server and port. This attribute is for LDAP and Membership authentication services only. The format is as follows:

```
host:port:min:max
```

NOTE This connection pool is different than the SDK connection pool configured in `serverconfig.xml`.

Default LDAP Connection Pool Size

This attribute sets the default minimum and maximum connection pool to be used with all LDAP authentication module configurations. If an entry for the host and port exists in the LDAP Connection Pool Size attribute, the minimum and maximum settings will not be used from LDAP Connection Default Pool Size.

Organization Attributes

The organization attributes in the Core Authentication service are:

Organization Authentication Modules

This list specifies the authentication modules that have been registered and are available to the organization. Each administrator can choose the type of authentication for each specific organization. Multiple authentication modules provide flexibility, but users must be sure that their login setting is appropriate for the selected authentication module. The default authentication is LDAP. The authentication services included with Access Manager are:

NOTE The Administrator must create and notify the core and authentication module templates in a created organization for that organization to function properly.

User Profile

This option allows you to specify options for a user profile.

- Required - This specifies that on successful authentication, the user needs to have a profile in the local Directory Server installed with Access Manager for the authentication service to issue an SSOToken.
- Dynamic - This specifies that on successful authentication, the authentication service will create the user profile if one does not already exist. The SSOToken will then be issued. The user profile is created in the local Directory Server installed with Access Manager.
- Dynamic With User Alias - This specifies that on successful authentication, the authentication services will create the user profile with the User Alias List attribute.
- Ignore - This specifies that the user profile is not required by the authentication service to issue the SSOToken for a successful authentication.

Administrator Authentication Configuration

Clicking the edit link will allow you to define the authentication service for administrators only. This attribute can be used if the authentication module for administrators needs to be different from the module for end users. The modules configured in this attribute are picked up when the Access Manager console is accessed. For example:

`http://servername.port/console_deploy_uri`

User Profile Dynamic Creation Default Roles

This field specifies the roles assigned to a new user whose profiles are created if Dynamic Creation is selected through the feature "User Profile" on page 298. There is no default value. The administrator must specify the DNs of the roles that will be assigned to the new user.

NOTE The role specified must be under the organization for which authentication is being configured. This role can be either an Access Manager or LDAP role, but it cannot be a filtered role.

If you wish to automatically assign specific services to the user, you have to configure the Required Services attribute in the User Profile.

Enable Persistent Cookie Mode

This option determines whether users can restart the browser and still return to their authenticated session. User sessions can be retained by enabling Enable Persistent Cookie Mode. When Enable Persistent Cookie Mode is enabled, a user session does not expire until its persistent cookie expires, or the user explicitly logs out. The expiration time is specified in Persistent Cookie Maximum Time. The default value is that Persistent Cookie Mode is not enabled and the authentication service uses only memory cookies.

NOTE A persistent cookie must be explicitly requested by the client using the `iPSPCookie=yes` parameter in the login URL.

Persistent Cookie Maximum Time

This field specifies the interval after which a persistent cookie expires. (Enable Persistent Cookie Mode must be enabled by selecting its checkbox.) The interval begins when the user's session has been successfully authenticated. The default value is `2147483` (time in seconds). The field will take any integer value between `0` and `2147483`.

People Container For All Users

After successful authentication by a user, the user's profile is retrieved. The value in this field specifies where to search for the profile. Generally, this value will be the DN of the default People Container. All user entries added to an organization are automatically added to the organization's default People Container. The default value is `ou=People`, and generally, this is completed with the organization name(s) and root suffix. The field will take a valid DN for any organizational unit.

NOTE Authentication searches for a user profile by:

- Searching under the default People Container, then
- Searching under the default organization, then
- Searching for the user in the default organization using the Alias Search Attribute Name attribute.

The final search is for SSO cases where the user name used to authenticate may not be the naming attribute in the profile. For example, a user may authenticate using Safeword ID of `jn10191`, but the profile is `uid=jamie`.

Alias Search Attribute Name

After successful authentication by a user, the user's profile is retrieved. This field specifies a second LDAP attribute to search from if a search on the first LDAP attribute, specified in "User Naming Attribute" on page 300, fails to locate a matching user profile. Primarily, this attribute will be used when the user identification returned from an authentication module is not the same as that specified in User Naming Attribute. For example, a RADIUS server might return `abc1234` but the user name is `abc`. There is no default value for this attribute. The field will take any valid LDAP attribute (for example, `cn`).

User Naming Attribute

After successful authentication by a user, the user's profile is retrieved. The value of this attribute specifies the LDAP attribute to use for the search. By default, Access Manager assumes that user entries are identified by the `uid` attribute. If your Directory Server uses a different attribute (such as `givenname`) specify the attribute name in this field.

Default Authentication Locale

This field specifies the default language subtype to be used by the authentication service. The default value is en_US. A listing of valid language subtypes can be found in Table 21-1.

> In order to use a different locale, all authentication templates for that locale must first be created. A new directory must then be created for these templates. See "Login URL Parameters" on page 146 for more information.

Table 21-1 Supported Language Locales

Language Tag	Language
af	Afrikaans
be	Byelorussian
bg	Bulgarian
ca	Catalan
cs	Czechoslovakian
da	Danish
de	German
el	Greek
en	English
es	Spanish
eu	Basque
fi	Finnish
fo	Faroese
fr	French
ga	Irish
gl	Galician
hr	Croatian
hu	Hungarian
id	Indonesian
is	Icelandic
it	Italian

Table 21-1 Supported Language Locales *(Continued)*

Language Tag	Language
ja	Japanese
ko	Korean
nl	Dutch
no	Norwegian
pl	Polish
pt	Portuguese
ro	Romanian
ru	Russian
sk	Slovakian
sl	Slovenian
sq	Albanian
sr	Serbian
sv	Swedish
tr	Turkish
uk	Ukrainian
zh	Chinese

Organization Authentication Configuration

This attribute sets the authentication module for the organization. The default authentication module is LDAP. One or more authentication modules can be selected by clicking the Edit link. If more than one module is selected, then the user will have to pass through the chain of all selected modules.

The modules configured in this attribute are used for authentication when users access the authentication module using the `/server_deploy_uri/UL/Login` format. See the Access Manager Developer's Guide for more information.

Enable Login Failure Lockout Mode

This feature specifies whether a user can attempt a second authentication if the first attempt failed. Selecting this attribute enables a lockout and the user will have only one chance at authentication. By default, the lockout feature is not enabled. This attribute works in conjunction with Lockout-related and notification attributes.

Login Failure Lockout Count

This attribute defines the number of attempts that a user may try to authenticate, within the time interval defined in Login Failure Lockout Interval, before being locked out.

Login Failure Lockout Interval

This attribute defines (in minutes) the time between two failed login attempts. If a login fails and is followed by another failed login that occurs within the lockout interval, then the lockout count is incremented. Otherwise, the lockout count is reset.

Email Address to Send Lockout Notification

This attribute specifies an email address that will receive notification if a user lockout occurs. To send email notification to multiple addresses, separate each email address with a space. For non-English locales, the format is:

```
email_address|locale|charset
```

Warn User After N Failures

This attribute specifies the number of authentication failures that can occur before Access Manager sends a warning message that the user will be locked out.

Login Failure Lockout Duration

This attribute enables memory locking. By default, the lockout mechanism will inactivate the User Profile (after a login failure) defined in Lockout Attribute Name. If the value of Login Failure Lockout Duration is greater than 0, then its memory locking and the user account will be locked for the number of minutes specified.

Lockout Attribute Name

This attribute designates any LDAP attribute that is to be set for lockout. The value in Lockout Attribute Value must also be changed to enable lockout for this attribute name. By default, Lockout Attribute Name is empty in the Access Manager Console. The default implementation values are `inetuserstatus` (LDAP attribute) and `inactive` when the user is locked out and Login Failure Lockout Duration is set to 0.

Lockout Attribute Value

This attribute specifies whether lockout is enabled or disabled for the attribute defined in Lockout Attribute Name. By default, the value is set to inactive for `inetuserstatus`.

Default Success Login URL

This field accepts a list of multiple values that specify the URL to which users are redirected after successful authentication. The format of this attribute is `clientType|URL`, although you can specify only the value of the URL which assumes a default type of HTML.

NOTE The default value is `/amconsole`. The *protocol*, *host* and *port* values are not longer needed in this release.

In the case of a remote console, this attribute should be manually modified to point to the console page on the actual remote console host.

Default Failure Login URL

This field accepts a list of multiple values that specify the URL to which users are redirected after an unsuccessful authentication. The format of this attribute is `clientType|URL`, although you can specify only the value of the URL which assumes a default type of HTML.

Authentication PostProcessing Class

This field specifies the name of the Java class used to customize post authentication processes for successful or unsuccessful logins. Example:

```
com.abc.authentication.PostProcessClass
```

The Java class must implement the following Java interface:

`com.sun.identity.authentication.spi.AMPostAuthProcessInterface`

Additionally, you must add the path to where the class is located to the Web Server's Java Classpath attribute.

Enable Generate UserID Mode

This attribute is used by the Membership authentication module. If this attribute field is enabled, the Membership module is able to generate user IDs, during the Self Registration process, for a specific user if the user ID already exists. The user IDs are generated from the Java class specified in Pluggable User Name Generator Class.

Pluggable User Name Generator Class

The field specifies the name of the Java class that will be used to generate user IDs when Enable Generate UserID Mode is enabled.

Default Authentication Level

The authentication level value indicates how much to trust authentications. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application can use the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level.

The authentication level should be set within the organization's specific authentication template. The Default Auth Level value described here will apply only when no authentication level has been specified in the Authentication Level field for a specific organization's authentication template. The Default Auth Level default value is 0. (The value in this attribute is not used by Access Manager but by any external application that may chose to use it.)

HTTP Basic Authentication Attributes

The HTTP Basic Authentication attribute is an organization attributes. The values applied to it under Service Configuration becomes the default value for the HTTP Basic Authentication template. The service template needs to be created after registering the service for the organization. The default value can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the organization.

The HTTP Basic Authentication attributes is:

Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is `0`.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See "Default Authentication Level" on page 306 for details.

Chapter 23

JDBC Authentication Attributes

The JDBC (Java Database Connectivity) Authentication attributes are organization attributes. The values applied to them under Service Configuration become the default values for the JDBC Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the organization. The JDBC Authentication attributes are:

Connection Type

This field specifies the connection type to the SQL database, using either a JNDI (Java Naming and Directory Interface) connection pool or JDBC driver. The options are as follows:

- `Connection pool is retrieved via JDNI`
- `Non-persistant JDBC connection`

The JNDI connection pool utilizes the configuration from the underlying web container.

Connection Pool JNDI Name

If JNDI is selected in Connection Type, this field specifies the connection pool name. Because JDBC authentication uses the JNDI connection pool provided by the web container, the setup of JNDI connection pool may not be consistent among other web containers.

The following example shows how to set up a connection pool for Web Server and MySQL 4.0:

1. In the Web Server console, create a JDBC connection pool with the following attributes:

   poolName: `samplePool`

   DataSource Classname: `com.mysql.jdbc.jdbc2.optional.MysqlDatacSource`

   serverName: server name of the mySQL server

   port: port number on which mySQL server is running

   user: user name of the database

   password: password of the user

   databaseName: database name

NOTE The jar file which contain the DataSource class and the JDBC Driver class mentioned in the following steps should be added to the application class path.

2. Configure the JDBC Resources. In the Web Server console, create a JDBC resource with the following attributes:

   JNDI name: `jdbc/samplePool`

   Pool name: `samplePool`

   Data Resource Enabled: `on`

3. Add the following lines to the `sun-web.xml` file of the application:

```
<resource-ref>

   <res-ref-name>jdbc/mySQL</res-ref-name>

   <jndi-name>jdbc/samplePool</jndi-name>

</resource-ref>
```

4. Add the following lines to the `web.xml` file of the application:

```
<resource-ref>

  <description>mySQL Database</description>

  <res-ref-name>jdbc/mySQL</res-ref-name>

  <res-type>javax.sql.DataSource</res-type>

  <res-auth>Container</res-auth>

</resource-ref>
```

Once you have completed the settings.the value for this attribute is as follows:

`java:comp/env/jdbc/mySQL`

JDBC Driver

If JDBC is selected in Connection Type, this field specifies the JDBC driver provided by the SQL database. For example:

```
com.mysql.jdbc.Driver
```

JDBC URL

This field specifies the database URL if JDBC is select in Connection Type. For example, the URL for mySQL is:

```
jdbc.mysql://hostname:port/databaseName
```

User to Connect to Database

This field specifies the user name from whom the database connection is made for the JDBC connection.

Password to Connect to Database

The field defines the password for the user specified in User to Connect to Database.

Password to Connect to Database (Confirm)

Confirm the password.

Password Column in Database

This field specifies the password column name in the SQL database.

Prepared Statement

This field specifies the SQL statement that retrieves the password of the user that is logging in. For example:

```
select Password from Employees where USERNAME = ?
```

Class to Transform Password Syntax

This attribute specifies the class name that transforms the password retrieved from the database, to the format of the user input, for password comparison. This class must implement the `JDBCPasswordSyntaxTransform` interface.

Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is `0`.

Chapter 24

LDAP Authentication Attributes

The LDAP Authentication attributes are organization attributes. The values applied to them under Service Configuration become the default values for the LDAP Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the organization. The LDAP Authentication attributes are:

Primary LDAP Server

This field specifies the host name and port number of the primary LDAP server specified during Access Manager installation. This is the first server contacted for LDAP authentication. The format is `hostname:port`. (If there is no port number, assume `389`.).

If you have Access Manager deployed with multiple domains, you can specify the communication link between specific instances of Access Manager and Directory Server in the following format (multiple entries must be prefixed by the local server name):

```
local_servername|server:port local_servername2|server2:port2 ...
```

For example, if you have two Access Manager instances deployed in different locations (`L1-machine1-IS` and `L2- machine2-IS`) communicating with different instances of Directory Server (`L1-machine1-DS` and `L2-machine2-DS`), it would look the following:

```
L1-machine1-IS.example.com|L1-machine1-DS.example.com:389
L2-machine2-IS.example.com|L2-machine2-DS.example.com:389
```

Secondary LDAP Server

This field specifies the host name and port number of a secondary LDAP server available to the Access Manager platform. If the primary LDAP server does not respond to a request for authentication, this server would then be contacted. If the primary server is up, Access Manager will switch back to the primary server. The format is also `hostname:port`. Multiple entries must be prefixed by the local server name.

CAUTION When authenticating users from a Directory Server that is remote from the Access Manager enterprise, it is important that both the Primary and Secondary LDAP Server Ports have values. The value for one Directory Server location can be used for both fields.

DN to Start User Search

This field specifies the DN of the node where the search for a user would start. (For performance reasons, this DN should be as specific as possible.) The default value is the root of the directory tree. Any valid DN will be recognized. If `OBJECT` is selected in the Search Scope attribute, the DN should specify one level above the level in which the profile exists.

Multiple entries must be prefixed by the local server name. The format is as follows:

```
servername|search dn
```

For multiple entries

```
servername1|search dn servername2|search dn servername3|search dn...
```

If multiple users are found for the same search, authentication will fail.

DN for Root User Bind

This field specifies the DN of the user that will be used to bind to the Directory Server specified in the Primary LDAP Server and Port field as administrator. The authentication service needs to bind as this DN in order to search for a matching user DN based on the user login ID. The default value is `amldapuser`. Any valid DN will be recognized.

Make sure that password is correct before you logout, because if it is incorrect, you will be locked out. If this should occur, you can login with the super user DN in the `com.iplanet.authentication.super.user` property in the `AMConfig.Properties` file. By default, this the amAdmin account with which you would normally log in, although you will use the full DN. For example:

`uid_amAdmin,ou=People,`*`AccessManager-base`*

Password for Root User Bind

This field carries the password for the administrator profile specified in the DN for Root User Bind field. There is no default value. Only the administrator's valid LDAP password will be recognized.

Password For Root User Bind (Confirm)

Confirmation of the password.

LDAP Attribute Used to Retrieve User Profile

After successful authentication by a user, the user's profile is retrieved. The value of this attribute is used to perform the search. The field specifies the LDAP attribute to use. By default, Access Manager assumes that user entries are identified by the `uid` attribute. If your Directory Server uses a different attribute (such as `givenname`)specify the attribute name in this field.

NOTE The user search filter will be a combination of the Search Filter attribute and the LDAP Attribute Used to Retrieve User Profile.

LDAP Attributes Used to Search for a User to be Authenticated

This field lists the attributes to be used to form the search filter for a user that is to be authenticated, and allows the user to authenticate with more than one attribute in the user's entry. For example, if this field is set to `uid`, `employeenumber` and `mail`, the user could authenticate with any of these names.

User Search Filter

This field specifies an attribute to be used to find the user under the DN to Start User Search field. It works with the User Entry Naming Attribute. There is no default value. Any valid user entry attribute will be recognized.

Search Scope

This menu indicates the number of levels in the Directory Server that will be searched for a matching user profile. The search begins from the node specified in the attribute "DN to Start User Search" on page 317. The default value is SUBTREE. One of the following choices can be selected from the list:

- OBJECT - Searches only the specified node

- ONELEVEL - Searches at the level of the specified node and one level down
- SUBTREE - Search all entries at and below the specified node

CAUTION Users from suborganizations may be able to login even if the sub organization's status is inactive. To avoid this, make sure that the Search Scope and the Base DN are set to the specific organization to which the user belongs.

Enable SSL Access to LDAP Server

This option enables SSL access to the Directory Server specified in the Primary and Secondary LDAP Server and Port field. By default, this is not enabled and the SSL protocol will not be used to access the Directory Server. However, if this attribute is enabled, you can bind to a non-SSL server.

If the LDAP Server is running with SSL enabled (LDAPS), you must make sure that Access Manager is configured with proper SSL trusted certificates so that AM could connect to Directory server over LDAPS protocol

Return User DN To Authenticate

When the Access Manager directory is the same as the directory configured for LDAP, this option may be enabled. If enabled, this option allows the LDAP authentication module to return the DN instead of the `userId`, and no search is necessary. Normally, an authentication module returns only the `userId`, and the authentication service searches for the user in the local Access Manager LDAP. If an external LDAP directory is used, this option is typically not enabled.

LDAP Server Check Interval

This attribute is used for LDAP Server failback. It defines the number of minutes in which a thread will "sleep" before verifying that the LDAP primary server is running.

User Creation Attributes List

This attribute is used by the LDAP authentication module when the LDAP server is configured as an external LDAP server. It contains a mapping of attributes between a local and an external Directory Server. This attribute has the following format:

```
attr1|externalattr1
attr2|externalattr2
```

When this attribute is populated, the values of the external attributes are read from the external Directory Server and are set for the internal Directory Server attributes. The values of the external attributes are set in the internal attributes only when the User Profile attribute (in the Core Authentication module) is set to “Dynamically Created” and the user does not exist in local Directory Server instance. The newly created user will contain the values for internal attributes, as specified in User Creation Attributes List, with the external attribute values to which they map.

Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is `0`.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See “Default Authentication Level” on page 306 for details.

Chapter 25

Membership Authentication Attributes

The Membership Authentication attributes are organization attributes. The values applied to them under Service Configuration become the default values for the Membership Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the subtrees of the organization. The Membership Authentication attributes are:

- “Authentication Level” on page 326

Minimum Password Length

This field specifies the minimum number of characters required for a password set during self-registration. The default value is 8.

If this value is changed, it should also be changed in the registration and error text in the following file:

AcessManager-base/locale/amAuthMembership.properties *(*PasswdMinChars *entry)*

Default User Roles

This field specifies the roles assigned to new users whose profiles are created through self-registration. There is no default value. The administrator must specify the DNs of the roles that will be assigned to the new user.

NOTE The role specified must be under the organization for which authentication is being configured. Only the roles that can be assigned to the user will be added during self-registration. All other DNs will be ignored. The role can be either an Access Manager role or an LDAP role, but filtered roles are not accepted.

User Status After Registration

This menu specifies whether services are immediately made available to a user who has self-registered. The default value is Active and services are available to the new user. By selecting Inactive, the administrator chooses to make no services available to a new user.

Primary LDAP Server

This field specifies the host name and port number of the primary LDAP server specified during Access Manager installation. This is the first server contacted for LDAP authentication. The format is hostname:port. (If there is no port number, assume 389.).

If you have Access Manager deployed with multiple domains, you can specify the communication link between specific instances of Access Manager and Directory Server in the following format (multiple entries must be prefixed by the local server name):

```
local_servername|server:port local_servername2|server:port ...
```

For example, if you have two Access Managers deployed in different locations (`L1-machine1-IS` and `L2- machine2-IS`) communicating with different instances of Access Manager (`L1-machine1-DS` and `L2-machine2-DS`), it would look the following:

```
L1-machine1-IS.example.com|L1-machine1-DS.example.com:389
L2-machine2-IS.example.com|L2-machine2-DS.example.com:389
```

Secondary LDAP Server

This field specifies the host name and port number of a secondary LDAP server available to the Access Manager platform. If the primary LDAP server does not respond to a request for authentication, this server would then be contacted. If the primary server is up, Access Manager will switch back to the primary server. The format is also `hostname:port`. Multiple entries must be prefixed by the local server name.

CAUTION When authenticating users from a Directory Server that is remote from the Access Manager enterprise, it is important that both the Primary and Secondary LDAP Server Ports have values. The value for one Directory Server location can be used for both fields.

DN to Start User Search

This field specifies the DN of the node where the search for a user would start. (For performance reasons, this DN should be as specific as possible.) The default value is the root of the directory tree. Any valid DN will be recognized. If OBJECT is selected in the Search Scope attribute, the DN should specify one level above the level in which the profile exists.

If you use multiple entries, the entries must be prefixed by the local server name.The format is as follows:

```
servername|search dn
```

For multiple entries

servername1|search dn servername2|search dn servername3|search dn...

If multiple users are found for the same search, authentication will fail.

DN for Root User Bind

This field specifies the DN of the user that will be used to bind to the Directory Server specified in the Primary LDAP Server and Port field as administrator. The authentication service needs to bind as this DN in order to search for a matching user DN based on the user login ID. The default is `amldapuser`. Any valid DN will be recognized.

Password for Root User Bind

This field carries the password for the administrator profile specified in the DN for Root User Bind field. There is no default value. Only the administrator's valid LDAP password will be recognized.

Password for Root User Bind (Confirm)

Confirmation of the password.

LDAP Attribute Used to Retrieve User Profile

This field specifies the attribute used for the naming convention of user entries. By default, Access Manager assumes that user entries are identified by the `uid` attribute. If your Directory Server uses a different attribute (such as `givenname`) specify the attribute name in this field.

LDAP Attributes Used to Search for a User to be Authenticated

This field lists the attributes to be used to form the search filter for a user that is to be authenticated, and allows the user to authenticate with more than one attribute in the user's entry. For example, if this field is set to `uid`, `employeenumber` and `mail`, the user could authenticate with any of these names.

User Search Filter

This field specifies an attribute to be used to find the user under the DN to Start User Search field. It works with the User Naming Attribute. There is no default value. Any valid user entry attribute will be recognized.

Search Scope

This menu indicates the number of levels in the Directory Server that will be searched for a matching user profile. The search begins from the node specified in the attribute "DN to Start User Search" on page 323. The default value is SUBTREE. One of the following choices can be selected from the list:

- OBJECT — Searches only the specified node
- ONELEVEL — Searches at the level of the specified node and one level down
- SUBTREE — Search all entries at and below the specified node

Enable SSL Access to LDAP Server

This option enables SSL access to the Directory Server specified in the Primary and Secondary LDAP Server and Port field. By default, the box is not checked and the SSL protocol will not be used to access the Directory Server.

If the LDAP Server is running with SSL enabled (LDAPS), you must make sure that Access Manager is configured with proper SSL trusted certificates so that AM could connect to Directory server over LDAPS protocol

Return User DN To Authenticate

When the Access Manager directory is the same as the directory configured for LDAP, this option may be enabled. If enabled, this option allows the LDAP authentication module to return the DN instead of the `userId`, and no search is necessary. Normally, an authentication module returns only the `userId`, and the authentication service searches for the user in the local Access Manager LDAP. If an external LDAP directory is used, this option is typically not enabled.

Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is `0`.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See "Default Authentication Level" on page 306 for details.

Chapter 26

MSISDN Authentication Attributes

The MSISDN Authentication attributes are organization attributes. The values applied to them under Service Configuration become the default values for the MSISDN Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the organization. The MSISDN Authentication attributes are:

Trusted Gateway IP Address

This attribute specifies a list of IP addresses of trusted clients that can access MSIDSN modules. You can set the IP addresses of all clients allows to access the MSISDN module by entering the address (for example, `123.456.123.111`) in the entry field and clicking Add. By default, the list is empty. If the attribute is left empty, then all clients are allowed. If you specify `none`, no clients are allowed.

MSISDN Number Argument

This field specifies a list of parameter names that identify which parameters to search in the request header or cookie header for the MSISDN number. For example, if you define `x-Cookie-Param`, `AM_NUMBER` and `COOKIE-ID`, the MSISDN authentication services will search those parameters for the MSISDN number.

LDAP Server and Port

This field specifies the host name and port number of the Directory Server in which the search will occur for the users with MSISDN numbers. The format is `hostname:port`. (If there is no port number, assume `389`.).

If you have Access Manager deployed with multiple domains, you can specify the communication link between specific instances of Access Manager and Directory Server in the following format (multiple entries must be prefixed by the local server name):

```
local_servername|server:port local_servername2|server2:port2 ...
```

For example, if you have two Access Manager instances deployed in different locations (`L1-machine1-IS` and `L2- machine2-IS`) communicating with different instances of Directory Server (`L1-machine1-DS` and `L2-machine2-DS`), it would look the following:

```
L1-machine1-IS.example.com|L1-machine1-DS.example.com:389
L2-machine2-IS.example.com|L2-machine2-DS.example.com:389
```

LDAP Start Search DN

This field specifies the DN of the node where the search for the user's MSISDN number should start. There is no default value. The field will recognize any valid DN. Multiple entries must be prefixed by the local server name. The format is as follows:

```
servername|search dn
```

For multiple entries

```
servername1|search dn servername2|search dn servername3|search dn...
```

If multiple users are found for the same search, authentication will fail.

Attribute To Use To Search LDAP

This field specifies the name of the attribute in the user's profile that contains MSISDN number to search for a particular user. The default value is `sunIdentityMSISDNNumber`. This value should not be changed, unless you are certain that another attribute in the user's profile contains the same MSISDN number.

LDAP Server Principal User

This attribute specifies the LDAP bind DN to allow MSISDN searches in the Directory Server. The default bind DN is `cn=amldapuser,ou=DSAME Users,dc=sun,dc=com`.

LDAP Server Principal Password

This attribute specifies the LDAP bind password for the bind DN, as defined in LDAP Server Principal User.

LDAP Server Principal Password (confirm)

Confirm the password.

SSL On For LDAP Access

This option enables SSL access to the Directory Server specified in the LDAP Server and Port attribute. By default, this is not enabled and the SSL protocol will not be used to access the Directory Server. However, if this attribute is enabled, you can bind to a non-SSL server.

MSISDN Header Search Attribute

This attribute specifies the headers to use for searching the request for the MSISDN number. The supported values are as follows:

- `SearchCookieHeader` - performs the search in the cookie.
- `SearchRequestHeader` - performs the search in the request header.
- `SearchRequestParameter` - performs the search in the request parameter.

By default, all options are selected.

Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses

the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is `0`.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See "Default Authentication Level" on page 306 for details.

Chapter 27

NT Authentication Attributes

The NT Authentication Attributes are organization attributes. The values applied to them under Service Configuration become the default values for the NT Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the subtrees of the organization.

In order to activate the NT Authentication module, Samba Client 2.2.2 must be downloaded and installed to the following directory:

AcessManager-base/SUNWam/bin

Samba Client is a file and print server for blending Windows and UNIX machines together without requiring a separate Windows NT/2000 Server. More information, and the download itself, can be accessed at http://wwws.sun.com/software/download/products/3e3af224.html.

Red Hat Linux ships with a Samba client, located in the following directory:

/usr/bin

In order to authenticate using the NT Authentication service for Linux, copy the client binary to the following Access Manager directory:

AccessManager-base/identity/bin

The NT Authentication attributes are:

NT Authentication Domain

This attribute defines the Domain name to which the user belongs.

NT Authentication Host

This attribute defines the NT authentication hostname. The hostname should be the netBIOS name, as opposed to the fully qualified domain name (FQDN). By default, the first part of the FQDN is the netBIOS name.

If the DHCP (Dynamic Host Configuration Protocol) is used, you would put a suitable entry in the HOSTS file on the Windows 2000 machine.

Name resolution will be performed based on the netBIOS name. If you do not have any server on your subnet supplying netBIOS name resolution, the mappings should be hardcoded.

For example, the hostname should be `example1` not `example1.company1.com`.

NT Samba Configuration File Name

This attribute defines the Samba configuration filename and supports the `-s` option in the `smbclient` command. The value must be the full directory path where the Samba configuration file is located. For example:

```
/etc/opt/SUNWam/config/smb.conf
```

Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is `0`.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See "Default Authentication Level" on page 306 for details. .

RADIUS Authentication Attributes

The RADIUS Authentication attributes are organization attributes. The values applied to them under Service Configuration become the default values for the RADIUS Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the organization. The RADIUS Authentication attributes are:

RADIUS Server 1

This field displays the IP address or fully qualified host name of the primary RADIUS server. The default IP address is `127.0.0.1`. The field will recognize any valid IP address or host name. Multiple entries must be prefixed by the local server name as in the following syntax:

```
local_servername|ip_address local_servername2|ip_adress ...
```

RADIUS Server 2

This field displays the IP address or fully qualified domain name (FQDN) of the secondary RADIUS server. It is a failover server which will be contacted if the primary server could not be contacted. The default IP address is `127.0.0.1`. Multiple entries must be prefixed by the local server name as in the following syntax:

```
local_servername|ip_address local_servername2|ip_adress ...
```

RADIUS Shared Secret

This field carries the shared secret for RADIUS authentication. The shared secret should have the same qualifications as a well-chosen password. There is no default value for this field.

RADIUS Shared Secret (Confirm)

Confirmation of the shared secret for RADIUS authentication.

RADIUS Server's Port

This field specifies the port on which the RADIUS server is listening. The default value is `1645`.

Timeout

This field specifies the time interval in seconds to wait for the RADIUS server to respond before a timeout. The default value is `3` seconds. It will recognize any number specifying the timeout in seconds.

Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses

the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is `0`.

SafeWord Authentication Attributes

The SafeWord Authentication Attributes are organization attributes. The values applied to them under Service Configuration become the default values for the SafeWord Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the subtrees of the organization.

This service allows for authenticating users using Secure Computing's SafeWord or SafeWord PremierAccess authentication servers.The SafeWord Authentication attributes are:

SafeWord Server

This field specifies the SafeWord or SafeWord PremiereAccess server name and port. Port 7482 is set as the default for a SafeWord server. The default port number for a SafeWord PremierAccess server is 5030.

SafeWord Server Verification Files Directory

This field specifies the directory into which the SafeWord client library places its verification files. The default is as follows:

```
/var/opt/SUNWam/auth/safeword/serverVerification
```

If a different directory is specified in this field, the directory must exist before attempting SafeWord authentication.

SafeWord Logging Enable

If selected, this attribute enables SafeWord logging. By default, SafeWord logging is enabled.

SafeWord Logging Level

This field specifies the SafeWord logging level. Select a level in the pulldown menu. The levels are `DEBUG`, `ERROR`, `INFO` and `NONE`.

SafeWord Log File

This attribute specifies the directory path and log file name for SafeWord client logging. The default path is as follows:

```
/var/opt/SUNWam/auth/safeword/safe.log
```

If a different path or filename is specified, they must exist before attempting SafeWord authentication.

If more than one organization is configured for SafeWord authentication, and different SafeWord servers are used, then different paths must be specified, or only the first organization where SafeWord authentication occurs will work. Likewise, if an organization changes SafeWord servers, the `swec.dat` file in the specified directory must be deleted before authentications to the newly configured SafeWord server will work.

SafeWord Authentication Connection Timeout

This attribute defines the timeout period (in seconds) between the SafeWord client (Access Manager) and the SafeWord server. The default is 120 seconds.

SafeWord Client Type

This attribute defines the Client Type that the SafeWord server uses to communicate with different clients, such as Mobile Client, VPN, Fixed Password, Challenge / Response, etc.

SafeWord eassp Version

This attribute specifies the Extended Authentication and Single Sign-on Protocol (EASSP) version. This field accepts either the standard (101) or premier access (201) protocol versions.

Minimum SafeWord Authenticator Strength

This attribute defines the minimum authenticator strength for the client / SafeWord server authentication. Each client type has a different authenticator value, and the higher the value, the higher the authenticator strength. 20 is the highest value possible. 0 is the lowest value possible.

Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses

the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is `0`.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See "Default Authentication Level" on page 306 for details.

Chapter 30

SAML Authentication Attributes

The SAML Authentication attribute is an organization attributes. The value applied to it under Service Configuration becomes the default value for the SAML Authentication template. The service template needs to be created after registering the service for the organization. The default value can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the organization.

The SAML Authentication attributes is:

Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is `0`.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See "Default Authentication Level" on page 306 for details.

Chapter 31

SecurID Authentication Attributes

The SecurID Authentication Attributes are organization attributes. The values applied to them under Service Configuration become the default values for the SecurID Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the subtrees of the organization.

This service allows for authenticating users using RSA's ACE/Server authentication server. The SecurID Authentication attributes are:

NOTE For this release of Access Manager, the SecurID Authentication module is not available for the Linux or Solaris x86 platforms and this should not be registered, configured, or enabled on these two platforms. It is only available for Solaris.

SecurID ACE/Server Configuration Path

This field specifies the directory in which the SecurID ACE/Server `sdconf.rec` file is located. The default is as follows:

`/opt/ace/data`

If a different directory is specified in this field, the directory must exist before attempting SecurID authentication.

SecurID Helper Configuration Port

This attribute specifies the port on which the SecurID helper 'listens' upon startup for the configuration information contained in the SecurID Helper Authentication Port attribute. The default is 58943.

If this attribute is changed, you must also change the `securidHelper.ports` entry in the `AMConfig.properties` file, and restart Access Manager. The entry in the `AMConfig.properties` file is a space-separated list of the ports for the instances of SecurID helpers. For each organization that communicates with a different ACE/Server (which has a different `sdconf.rec` file), there must be a separate SecurID helper.

SecurID Helper Authentication Port

This attribute specifies the port that the organization's SecurID authentication module will configure its SecurID helper instance to 'listen' for authentication requests. This port number must be unique across all organizations using SecurID or Unix authentication. The default port is 57943.

Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is `0`.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See "Default Authentication Level" on page 306 for details.

Chapter 32

Unix Authentication Attributes

The Unix Authentication Service consists of global and organization attributes. The values applied to the global attributes are applied across the Sun Java System Acceess Manager configuration, and are inherited by every configured organization. They can not be applied directly to roles or organizations, as the goal of global attributes is to customize the Access Manager application. Values applied to the organization attributes are default values for each organization configured and can be changed when the service is registered to the organization. The organization attributes are not inherited by entries of the organization. The Unix Authentication Attributes are divided into:

NOTE If any of the Unix authentication attributes are modified, both Access Manager and the `amunixd` helper must be restarted.

Global Attributes

The global attributes in the Unix Authentication service are:

Unix Helper Configuration Port

This attribute specifies the port to which the Unix Helper ‘listens’ upon startup for the configuration information contained in the Unix Helper Authentication Port, Unix Helper Timeout, and Unix Helper Threads attributes. The default is 58946.

If this attribute is changed, you must also change the `unixHelper.port` entry in the `AMConfig.properties` file, and restart Access Manager.

Unix Helper Authentication Port

This attribute specifies the port to which the Unix Helper ‘listens’ for authentication requests after configuration. The default port is 57946.

Unix Helper Timeout

This attribute specifies the number of minutes that users have to complete authentication. If users surpass the allotted time, authentication automatically fails. The default time is set to 3 minutes.

Unix Helper Threads

This attribute specifies the maximum number of permitted simultaneous Unix authentication sessions. If the maximum is reached at a given moment, subsequent authentication attempts are not allowed until a session is freed up. The default is set to 5.

Organization Attribute

The organization attribute for the Unix Authentication service is:

Authentication Level

The authentication level is set separately for each method of authentication. The value The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO

token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is `0`.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See "Default Authentication Level" on page 306 for details.

Chapter 33

Windows Desktop SSO Authentication Attributes

The Windows Desktop SSO Authentication attributes are organization attributes. The values applied to them under Service Configuration become the default values for the Windows Desktop SSO Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the subtrees of the organization.

This authentication module requires the Kerberos authentication service provided by any Kerberos Domain Controller.

The Windows Desktop SSO Authentication attributes are:

Service Principal

This attribute specifies the Kerberos principal that is used for authentication. Use the following format:

`HTTP/`*`hostname.domainname@dc_domain_name`*

hostname and *domainame* represent the hostname and domain name of the Access Manager instance. *dc_domain_name* is the Kerberos domain in which the Windows 2000 Kerberos server (domain controller) resides. It is possibly different from the domain name of the Access Manager.

Keytab Filename

This attribute specifies the Kerberos keytab file that is used for authentication. Use the following format, although the format is not required:

hostname.HTTP.keytab

hostname is the hostname of the Access Manager instance.

Kerberos Realm

This attribute specifies the Kerberos Distribution Center (domain controller) domain name. Depending up on your configuration, the domain name of the domain controller may be different than the Access Manager domain name.

Kerberos Server Name

This attribute specifies the Kerberos Distribution Center (the domain controller) hostname. You must enter the fully qualified domain name (FQDN) of the domain controller.

Return Principal With Domain Name

If enabled, this attributes allows Access Manager to automatically return the Kerberos principal with the domain controller's domain name during authentication.

Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses

the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is `0`.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See "Default Authentication Level" on page 306 for details.

Chapter 34

Authentication Configuration Service Attributes

The Authentication Configuration Service attributes are dynamic and organization attributes. These attributes can be defined for an organization, service, or role. The organization attributes are defined in the Core Authentication module.

If the role is assigned to a user or a user is assigned to the organization, these attributes, by default, are inherited by the user. The Authentication Configuration Attributes are:

Authentication Configuration

Clicking on the Edit link will display the Authentication Configuration interface. It allows you to configure the authentication modules for role-based or organization-based authentication.

The following table lists the authentication module configuration options:

Module Name	Allows you to select from the list of default authentication modules available to Access Manager.

Flag	This pull-down menu allows you specify the authentication module requirements. It can be one of: • REQUIRED - The authentication module is required to succeed. If it succeeds or fails, authentication continues to proceed down the authentication module list. • REQUISITE - The authentication module is required to succeed. If it succeeds, authentication continues down the authentication module list. If it fails, control returns to the application (authentication does not proceed down the authentication module list.) • SUFFICIENT - The authentication module is not required to succeed. If it does succeed, control immediately returns to the application (authentication does not proceed down the authentication module list.). If it fails, authentication continues down the list. • OPTIONAL - The authentication module is not required to succeed. If it succeeds or fails, authentication still continues to proceed down the list. These flags establish an enforcement criteria for the authentication module for which they are defined. There hierarchy for enforcement, with REQURIED being the highest, and OPTION being the lowest. For example, if an administrator defines an LDAP module with the REQUIRED flag, then the user's credential must pass the LDAP authentication requirements to access a given resource. If you add multiple authentication modules and for each module the Flag is set to REQUIRED, the user must pass all authentication requirements before being granted access. For more information on the flag definitions, refer to the JAAS (Java Authentication and Authorization Service) located at: `http://java.sun.com/security/jaas/doc/module.html`
Option	Allows for additional options for the module as a key=value pair. Multiple options are separated by a space.

Login Success URL

This attribute specifies the URL that the user will be redirected to upon successful authentication.

Login Failure URL

This attribute specifies the URL that the user will be redirected to upon unsuccessful authentication.

Authentication Post Processing Class

This attribute defines the name of the Java class used to customize the post authentication process after a login success or failure.

Conflict Resolution Level

This attribute applies to roles only. Conflict Resolution level sets a priority level for the Authentication Configuration attributes for roles that may contain the same user. For example, if User1 is assigned to both Role1 and Role2, you can define a higher priority level for Role1 so when the user attempts authentication Role1 will have the highest priority for success or failure redirects and for post authentication processes.

Client Detection Service Attributes

The Client Detection Service attributes are global attributes. The values applied to them are applied across the Access Manager configuration and are inherited by every configured organization. (They cannot be applied directly to roles or organizations, as the goal of global attributes is to customize the Access Manager application.) The Client Detection Attributes are:

Client Types

In order to detect client types, Access Manager needs to recognize their identifying characteristics. These characteristics identify the properties of all supported types in the form of client data. This attribute allows you to modify the client data through the Client Manager interface. To access the Client Manager, click the Edit link.

Out of the box, Access Manager contains the following client types:

- HDML
- HTML
- JHTML
- VoiceX
- WML

- XHTML
- cHTML
- iHTML
- For descriptions of these client types, see the Sun Java System Portal Server, Mobile Access 2005Q1 Administration Guide at the following location:

  http://docs.sun.com/app/docs/coll/PortalServer_05q1

Client Manager

The Client Manager is the interface that lists the base clients, styles and associated properties, and allows you to add and configure devices.

Base Client Types

The Base client types are listed at the top of Client Manager. These client types contain the default properties that can be inherited by all devices that belong to the client type.

Style Profile

The Client Manager groups all available clients, including the Base client type itself, in the Styles pulldown menu. The selected Style (or, parent profile) defines properties that are common to its configured child devices. The devices dynamically inherit the properties of the parent profile

The Current Style Properties link launches a read-only Client Editor window for viewing the style properties.

Device Profile

When a style is selected, the Client Manager displays the device profiles configured for that style. Devices are sorted by user agent (device name) and can be filtered by entering the user agent string in the Filter filed (wildcards are accepted).

For each device, you can modify the client properties by clicking on the Edit link located next to each device name. The properties are then displayed in the Client Editor window. To edit the properties, select the following classifications from the pull-down list:

Hardware Platform. Contains properties of the device's hardware, such as display size, supported character sets, and so forth.

Software Platform. Contains properties of the device's application environment, operating system, and installed software.

Network Characteristics. Contains properties describing the network environment, including the supported bearers.

BrowserUA. Contains attributes related to the browser user agent running on the device.

WapCharacteristics. Contains properties of the Wireless Application Protocol (WAP) environment supported by the device.

PushCharacteristicsNames. Contains properties of the WAP environment supported by the device.

Additional Properties. Allows you to add additional properties for the device.

For specific property definitions, see the Open Mobile Alliance Ltd. (OMA) *Wireless Application Protocol, Version 20-Oct-2001* at the following location:

http://www1.wapforum.org/tech/terms.asp?doc=WAP-248-UAProf-20011020-a.pdf

NOTE In order to access the document, you may first have to register with WAP Forum™. For information, please visit http://www.wapforum.org/faqs/index.htm

Once the properties have been modified, click Save. The device will display "**" characters to denote that the device has been customized. Use the Default link to remove the customized properties and reset the device back to the default settings.

To add a new device for a style, click the New Device button. The Create New Device window is displayed with the following fields:

Style. Displays the base style for the device, for example HTML.

Device User Agent. Accepts a name for the device.

Click Next to display the following fields:

Client Type Name. Displays the client type, for example HTML. The client type name must be unique across all devices.

The Immediate Parent For This Device. Accepts the parent (base) client type for the device. For example, HTML.

The HTTP User Agent String. Defines the User-Agent in the HTTP request header. For example, Mozilla/4.0.

Click OK and customize the device properties. For specific property definitions, see the Open Mobile Alliance Ltd. (OMA) *Wireless Application Protocol, Version 20-Oct-2001* at the following location:

http://www1.wapforum.org/tech/

To duplicate a device and its properties, click the Duplicate link. Device names must be unique. By default, Access Manager will rename the device to `copy_of_`*`devicename`*`.`

To delete any device, click the Delete link listed with the device.

Default Client Type

This attribute defines the default client type derived from the list of client types in the Client Types attribute. The default is `genericHTML`.

Client Detection Class

This attribute defines the client detection class for which all client detection requests are routed. The string returned by this attribute should match one of the client types listed in the Client Types attribute. The default client detection class is `com.sun.mobile.cdm.FEDIClientDetector`. Access Manager also contains `com.iplanet.services.cdm.ClientDetectionDefaultImpl`.

Enable Client Detection

This attribute allows you to enable client detection. If client detection is enabled (selected), every request is routed thought the class specified in the Client Detection Class attribute.

By default, the client detection capability is enabled. If this attribute is not selected, Access Manager assumes that the client is genericHTML and will be accessed from a HTML browser.

Chapter 36

Globalization Setting Service Attributes

The Globalization Setting Service attributes are global attributes. The values applied to them are applied across the Access Manager configuration and are inherited by every configured organization. (They cannot be applied directly to roles or organizations, as the goal of global attributes is to customize the Access Manager application.) The Globalization Setting Attributes are:

Charsets Supported By Each Locale

This attribute lists the charset support for each locale, which indicates the mapping between locale and charset. The format is as follows:

`locale=`*localename*`|charset=charset1;charset2;charset3;...;charsetn`

You can add, edit, duplicate and remove charsets with the buttons located at the bottom of the attribute.

Charset Aliases

This attribute lists the codeset names (which map to IANA names) that will be used to send the response. These codeset names do not need to match java codeset names. Currently, there is a hash table to map java character sets into IANA charsets and vice versa. The alias format is as follows:

mimeName=*charset*|javaName=*charset*

For example:

```
mimeName=Shift_JIS|javaName=SJIS
```

This implies that both denote same character set.

You can add, edit, duplicate and remove character set aliases with the buttons located at the bottom of the attribute.

Auto Generated Common Name Format

This display option allows you to define the way in which a name is automatically generated, to accommodate name formats for different locales and character sets. The default syntax is as follows (please note that including commas and / or spaces in the definition will display in the name format):

```
en_us = {givenname} {initials} {sn}
```

For example, if you wanted to display a new name format for a user (User One) with a uid (11111) for the Chinese character set, use the following stands:

```
zh = {sn}{givenname}({uid})
```

This would display as:

```
OneUser 11111
```

Chapter 37

Logging Service Attributes

The Logging Service attributes are global attributes. The values applied to them are applied across the Sun Java System Acceess Manager configuration and are inherited by every configured organization. (They can not be applied directly to roles or organizations as the goal of global attributes is to customize the Access Manager application.) The Logging Attributes are:

- "Buffer Time" on page 369
- "Enable Time Buffering" on page 369

Maximum Log Size

This attribute accepts a value for the maximum size (in bytes) of a Access Manager log file. The default value is `1000000`.

Number of History Files

This attribute has a value equal to the number of backup log files that will be retained for historical analysis. Any integer can be entered depending on the partition size and available disk space of the local system. The default value is 3.

NOTE Entering a value of `0` is interpreted to be the same as a value of `1`, meaning that if you specify `0`, a backup log file will be created.

Log File Location

The file-based logging function needs a location where log files can be stored. This field accepts a full directory path to that location. The default location is:

`/var/opt/SUNWam/logs`

If a non-default directory is being used, this directory must have write permission to the user under which Access Manager is running.

When configuring the log location for DB (database) logging (such as, Oracle or MySQL), part of the log location is case sensitive.

For example, if you are logging to an Oracle database, the log location should be:

`jdbc:oracle:thin:@machine.domain:port:DBName`

`jdbc:oracle:thin` must be lower case.

NOTE To configure logging to DB, add the JDBC driver files to the web container's JVM classpath. You need to manually add JDBC driver files to the classpath of the amadmin script, otherwise amadmin logging can not load the JDBC driver.

Any changes in logging attribute values require a restart of the Access Manager before the changes are activated.

Logging Type

This attribute allows you to specify either File, for flat file logging, or DB for database logging.

Database User Name

This attribute accepts the name of the user that will connect to the database when the Logging Type attribute is set to DB.

Database User Password

This attribute accepts the database user password when the Logging Type attribute is set to DB.

Database User Password (Confirm)

Confirmation of the database password.

Database Driver Name

This attribute allows the user to specify the driver that is to be used for the logging implementation class.

Configurable Log Fields

This parameter represents the list of fields that are to be logged. By default, the following fields are logged:

- Domain
- Hostname
- IPAddress
- LoggedBy
- Loglevel
- LoginID
- ModuleName

Log Verification Frequency

This attribute sets the frequency (in seconds) that the server should verify the logs to detect tampering. The default time is 3600 seconds. This parameter applies to secure logging only.

Log Signature Time

This parameter sets the frequency (in seconds) that the log will be signed. The default time is 900 seconds. This parameter applies to secure logging only.

Enable Secure Logging

This attribute specifies whether or not to enable secure logging. By default, secure logging is off. Secure Logging enables detection of unauthorized changes or tampering of security logs.

Maximum Number of Records

This attribute sets the maximum number of records that the Java LogReader interfaces return, regardless of how many records match the read query. By default, it is set to 500. This attribute can be overridden by the caller of the Logging API through the LogQuery parameter.

Number Of Files Per Archive

This attribute is only applicable to secure logging. It specifies when the log files and keystore need to be archived, and the secure keystore regenerated, for subsequent secure logging. The default is five files per logger.

Buffer Size

This attribute specifies the maximum amount of log records to be buffered in memory before they are sent to the logging service to be logged. The default is one record.

DB Failure Memory Buffer Size

This attribute defines the maximum number of log records held in memory if database (DB) logging fails. This attribute is only applicable when DB logging is specified. When the Access Manager logging service loses connection to the DB, it will buffer up to the number of records specified. This attribute defaults to two times of the value defined in the Buffer Size attribute.

Buffer Time

This attribute defines the amount of time that the log records will buffered in memory before they are sent to the logging service to be logged. The default is 3600 seconds.

Enable Time Buffering

When selected as ON, Access Manager will set a time limit for log records to be buffered in memory. The amount of time is set in the Buffer Time attribute.

Chapter 38

Naming Service Attributes

The Naming Service attributes are global attributes. The values applied to them are carried across the Sun Java System Acceess Manager configuration and inherited by every configured organization. (They can not be applied directly to roles or organizations as the goal of global attributes is to customize the Access Manager application.)

The Naming Service allows clients to find the correct service URL if the platform is running more than one Access Manager. When a naming URL is found, the naming service will decode the session of the user and dynamically replace the protocol, host, and port with the parameters from the session. This ensures that the URL returned for the service is for the host that the user session was created on. The Naming Attributes are:

• "JAXRPC Endpoint URL" on page 374

Profile Service URL

This field takes a value equal to

`%protocol://%host:%port/`*Server_DEPLOY_URI*`/profileservice`

This syntax allows for dynamic substitution of the profile URL based on the specific session parameters.

Session Service URL

This field takes a value equal to

`%protocol://%host:%port/`*Server_DEPLOY_URI*`/sessionservice`

This syntax allows for dynamic substitution of the session URL based on the specific session parameters.

Logging Service URL

This field takes a value equal to

`%protocol://%host:%port/`*Server_DEPLOY_URI*`/loggingservice`

This syntax allows for dynamic substitution of the logging URL based on the specific session parameters.

Policy Service URL

This field takes a value equal to

`%protocol://%host:%port/`*Server_DEPLOY_URI*`/policyservice`

This syntax allows for dynamic substitution of the policy URL based on the specific session parameters.

Auth Service URL

This field takes a value equal to

%protocol://%host:%port/*Server_DEPLOY_URI*/authservice

This syntax allows for dynamic substitution of the authentication URL based on the specific session parameters.

SAML Web Profile/Artifact Service URL

This field takes a value equal to

%protocol://%host:%port/*Server_DEPLOY_URI*/SAMLAwareServlet

This syntax allows for dynamic substitution of the SAML web profile / artifact URL based on the specific session parameters.

SAML SOAP Service URL

This field takes a value equal to

%protocol://%host:%port/*Server_DEPLOY_URI*/SAMLSOAPReceiver

This syntax allows for dynamic substitution of the SAML SOAP URL based on the specific session parameters.

SAML Web Profile/POST Service URL

This field takes a value equal to

%protocol://%host:%port/*Server_DEPLOY_URI*/SAMLPOSTProfileServlet

This syntax allows for dynamic substitution of the SAML web profile / POST URL based on the specific session parameters.

SAML Assertion Manager Service URL

This field takes a value equal to

%protocol://%host:%port/*Server_DEPLOY_URI*/AssertionManagerServlet/AssertionManagerIF

This syntax allows for dynamic substitution of the SAML Assertion Manager Service URL based on the specific session parameters.

Federation Assertion Manager Service URL

This field takes a value equal to

```
%protocol://%host:%port/amserver/FSAssertionManagerServlet/FSAssertionManagerIF
```

This syntax allows for dynamic substitution of the Federation Assertion Manager Service URL based on the specific session parameters.

Identity SDK Service URL

This field takes a value equal to

```
%protocol://%host:%port/amserver/UserManagementServlet/
```

This syntax allows for dynamic substitution of the Identity SDK Service URL based on the specific session parameters.

Security Token Manager URL

This field takes a value equal to

```
%protocol://%host:%port/amserver/SecurityTokenManagerServlet/SecurityTokenManagerIF/
```

This syntax allows for dynamic substitution of the Security Token Manager URL based on the specific session parameters.

JAXRPC Endpoint URL

This field takes a value equal to

```
%protocol://%host:%port/amserver/jaxrpc/
```

This syntax allows for dynamic substitution of the JAXRPC Endpoint URL based on the specific session parameters.

Chapter 39

Password Reset Service Attributes

The Password Reset Service attributes are organization attributes. The values applied to them under Service Configuration become the default values for the Password Reset Service in a given organization. Organization attributes are not inherited by entries in the subtrees of the organization.

The Password Reset attributes are:

User Validation

This attribute specifies the value that is used to search for the user whose password is to be reset.

Secret Question

This field allows you to add a list of questions that the user can use to reset his / her password. To add a question, type it in the Secret Question filed and click Add. The selected questions will appear in the user’s User Profile page. The user can then select a question for resetting the password.

Users may create their own question if the Personal Question Enabled attribute is selected.

Search Filter

This attribute specifies the search filter to be used to find user entries.

Base DN

This attribute specifies the DN from which the user search will start. If no DN is specified, the search will start from the organization DN. You should not use `cn=directorymanager` as the base DN, due to proxy authentication conflicts.

Bind DN

This attribute value is used with Bind Password to reset the user password.

Bind Password

This attribute value is used with Bind DN to reset the user password.

Password Reset Option

This attribute determines the classname for resetting the password. The default classname is:

```
com.sun.identity.password.RandomPasswordGenerator
```

The password reset class can be customized through a plug-in. This class needs to be implemented by the `PasswordGenerator` interface. See the *Access Manager Developer's Guide* for more information.

Password Change Notification Option

This attribute determines the method for user notification of password resetting. The default classname is:

```
com.sun.identity.password.EmailPassword
```

The password notification class can be customized through a plugin. This class needs to be implemented by the `NotifyPassword` interface. See the *Access Manager Developer's Guide* for more information.

Enable Password Reset

Selecting this attribute will enable the password reset feature.

Enable Personal Question

Selecting this attribute will allow a user to create a unique question for password resetting.

Maximum Number of Questions

This value specifies the maximum number of questions to be asked in the password reset page.

Force Change Password on Next Login

When enabled, this option forces the user to change his or her password on the next login. If you want an administrator, other than the top-level administrator, to set the force password reset option, you must modify the Default Permissions ACIs to allow access to that attribute.

Enable Password Reset Failure Lockout

This attribute specifies whether to disallow users to reset their password if that user initially fails to reset the password using the Password Reset application. By default, this feature is not enabled.

Password Reset Failure Lockout Count

This attributes defines the number of attempts that a user may try to reset password, within the time interval defined in Password Reset Failure Lockout Interval, before being locked out.

For example, if Password Reset Failure Lockout Count is set to 5 and Login Failure Lockout Interval is set to 5 minutes, the user has five chances within five minutes to reset the password before being locked out.

Password Reset Failure Lockout Interval

This attribute defines (in minutes) the amount of time in which the number of password reset attempts (as defined in Password Reset Failure Lockout Count) can be completed, before being locked out.

Email Address to Send Lockout Notification

This attribute specifies an email address that will receive notification if a user is locked out from the Password Reset service. Specify multiple email address in a space-separated list.

Warn User After N Failure

This attribute specifies the number of password reset failures that can occur before Access Manager sends a warning message that user will be locked out.

Password Reset Failure Lockout Duration

This attribute defines (in minutes) the duration that user will not be able to attempt a password reset if a lockout has occurred.

Password Reset Lockout Attribute Name

This attribute contains the `inetuserstatus` value that is set in Password Reset Lockout Attribute Value. If a user is locked out from Password Reset, and the Password Reset Failure Lockout Duration (minutes) variable is set to 0, `inetuserstatus` will be set to inactive, prohibiting the user from attempting to reset his or her password.

Password Reset Lockout Attribute Value

This attribute specifies the `inetuserstatus` value (contained in Password Reset Lockout Attribute Name) of the user status, as either active or inactive. If a user is locked out from Password Reset, and the Password Reset Failure Lockout Duration (minutes) variable is set to 0, `inetuserstatus` will be set to inactive, prohibiting the user from attempting to reset his or her password.

Chapter 40

Platform Service Attributes

The Platform Service attributes are global attributes. The values applied to them are carried across the Sun Java System Acceess Manager configuration and inherited by every configured organization. (They can not be applied directly to roles or organizations as the goal of global attributes is to customize the Access Manager application.) The Platform Attributes are:

Server List

The naming service reads this attribute at initialization time. This list contains the Access Manager session servers in a single Access Manager configuration. For example, if two Access Managers are installed and should work as one, they must both be included in this list. If the host specified in a request for a service URL is not in this list, the naming service will reject the request. The first value in the list specifies the host name and port of the server specified during installation. At the end of the list, there is a two-byte value that uniquely identifies the server. Each server that is participating in load balancing or failover needs to have a unique identifier. This is also used to shorten the cookie length by mapping the server URL to the server ID. For example:

protocol://*server_domain*:*port*|01

Additional servers can be added using the format protocol://*server_domain*: *port* |01|*instance_name*

Only the naming service protocol should be used in this attribute.

Platform Locale

The platform locale value is the default language subtype that Access Manager was installed with. The authentication, logging and administration services are administered in the language of this value. The default is en_US. See Table 21-1 on page 301 for a listing of all supported language subtypes.

Cookie Domains

This is the list of domains that will be returned in the cookie header when setting a cookie to the user's browser during authentication. If empty, no cookie domain will be set. In other words, the Access Manager session cookie will only be forwarded to the Access Manager itself and to no other servers in the domain. If SSO is required with other servers in the domain, this attribute must be set with the cookie domain. If you had two interfaces in different domains on one Access Manager then you would need to set both cookie domains in this attribute. If a load balancer is used, the cookie domain must be that of the load balancer's domain, not the servers behind the load balancer. The default value for this field is the domain of the installed Access Manager.

NOTE Make sure that the correct cookie domain is entered. If the cookie domain is incorrect, you will not be able to login to Access Manager.

Login Service URL

This field specifies the URL of the login page. The default value for this attribute is /*Service_DEPLOY_URI*/UI/Login.

Logout Service URL

This field specifies the URL of the logout page. The default value for this attribute is `/`*Service_DEPLOY_URI*`/UI/Logout`.

Available Locales

This attribute stores all available locales configured for the platform. Consider an application that lets the user choose the user's locale. This application would get this attribute from the platform profile and present the list of locales to the user. The user would choose a locale and the application would set this in the user entry `preferredLocale`.

Client Char Sets

This attribute specifies the character set for different clients at the platform level. It contains a list of client types and the corresponding character sets. The format is as follows:

```
clientType|charset
clientType2|charset
```

For example:

```
genericHTML|UTF-8
```

Chapter 41

Policy Configuration Service Attributes

The Policy Configuration Service attributes consist of global and organization attributes.The values applied to the global attributes are applied across the Sun Java System Acceess Manager configuration and are inherited by every configured organization. (They can not be applied directly to roles or organizations as the goal of global attributes is to customize the Access Manager application.) The values applied to the organization attributes under Service Management become the default values for Policy configuration. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the organization. The Policy Configuration attributes are separated into:

- "Global Attributes" on page 385
- "Organization Attributes" on page 386

Global Attributes

The global attributes in the Policy Configurative service are:

- "Resource Comparator" on page 386
- "Continue Evaluation On Deny Decision" on page 386

Resource Comparator

This attribute specifies the resource comparator information, which is used to compare resources specified in a Policy rule definition. Resource comparison is used for both policy creation and evaluation. This attribute contains the following values:

`serviceType`	Specifies the service to which the comparator should be used.
`class`	Defines the java class that implements the resource comparison algorithm.
`wildcard`	Specifies the wildcard that can be defined in resource names
`delimiter`	Specifies the delimiter to be used in the resource name.
`caseSensitivity`	Specifies if the comparison of the two resources should consider or ignore case. `False` ignores case, `True` considers case.

Continue Evaluation On Deny Decision

This attribute specifies whether or not the policy framework should continue evaluating subsequent policies, even if a DENY policy decision exists. If it is unselected (default), policy evaluation would skip subsequent policies once the DENY decision is recognized.

Organization Attributes

The organization attributes in the Policy Configuration service are:

- "LDAP Server and Port" on page 388
- "LDAP Base DN" on page 388
- "LDAP Users Base DN" on page 388
- "Access Manager Roles Base DN" on page 389
- "LDAP Bind DN" on page 389
- "LDAP Bind Password" on page 389
- "LDAP Bind Password (Confirm)" on page 389

LDAP Server and Port

This field specifies the host name and port number of the primary LDAP server specified during Access Manager installation that will be used to search for Policy subjects, such as LDAP users, LDAP roles, LDAP groups, etc. The format is *hostname:port* For example:

`machine1.example.com:389`

For failover configuration to multiple LDAP server hosts, this value can be a space-delimited list of hosts. The format is hostname1:port1 hostname2:port2...

For example:

`machine1.example1.com:389 machine2.example1.com:389`

Multiple entries must be prefixed by the local server name. This is to allow specific Access Managers to be configured to talk to specific Directory Servers.

The format is `servername|hostname:port`

For example:

`machine1.example1.com|machine1.example1.com:389`

`machine1.example2.com|machine1.example2.com:389`

For failover configuration:

`IS_Server1.example1.com|machine1.example1.com:389 machine2.example.com1:389`

`IS_Server2.example2.com|machine1.example2.com:389 machine2.example2.com:389`

LDAP Base DN

This field specifies the base DN in the LDAP server from which to begin the search. By default, it is the top-level organization of the Access Manager installation.

LDAP Users Base DN

This attribute specifies the base DN used by the LDAP Users subject in the LDAP server from which to begin the search. By default, it is the top-level organization of the Access Manager installation base.

Access Manager Roles Base DN

This attribute specifies the base DN used by the Access Manager Roles subject in the LDAP server from which to begin the search. By default, it is the top-level organization of the Access Manager installation base.

LDAP Bind DN

This field specifies the bind DN in the LDAP server.

LDAP Bind Password

This attribute defines the password to be used for binding to the LDAP server. By default, the `amldapuser` password that was entered during installation is used as the bind user.

LDAP Bind Password (Confirm)

Confirmation of the LDAP Bind password.

LDAP Organization Search Filter

Specifies the search filter to be used to find organization entries. The default is `(objectclass=sunMangagedOrganization)`.

LDAP Organization Search Scope

This attribute defines the scope to be used to find organization entries. The scope must be one of the following:

- `SCOPE_BASE`
- `SCOPE_ONE`
- `SCOPE_SUB` (default)

LDAP Groups Search Filter

Specifies the search filter to be used to find group entries. The default is `(objectclass=groupOfUniqueNames)`.

LDAP Groups Search Scope

This attribute defines the scope to be used to find group entries. The scope must be one of the following:

- `SCOPE_BASE`
- `SCOPE_ONE`
- `SCOPE_SUB` (default)

LDAP Users Search Filter

Specifies the search filter to be used to find user entries. The default is `(objectclass=inetorgperson)`.

LDAP Users Search Scope

This attribute defines the scope to be used to find user entries. The scope must be one of the following:

- `SCOPE_BASE`
- `SCOPE_ONE`
- `SCOPE_SUB` (default)

LDAP Roles Search Filter

Specifies the search filter to be used to find entries for roles. The default is `(&(objectclass=ldapsubentry)(objectclass=nsroledefinitions))`

LDAP Roles Search Scope

This attribute defines the scope to be used to find entries for roles. The scope must be one of the following:

- SCOPE_BASE
- SCOPE_ONE
- SCOPE_SUB (default)

Access Manager Roles Search Scope

This attribute defines the scope to be used to find entries for Access Manager Roles subject. The scope must be one of the following:

- SCOPE_BASE
- SCOPE_ONE
- SCOPE_SUB (default)

LDAP Organization Search Attribute

This field defines the attribute type for which to conduct a search on an organization. The default is o.

LDAP Groups Search Attribute

This field defines the attribute type for which to conduct a search on a group. The default is cn.

LDAP Users Search Attribute

This field defines the attribute type for which to conduct a search on a user. The default is uid.

LDAP Roles Search Attribute

This field defines the attribute type for which to conduct a search on a role. The default is cn.

Maximum Results Returned From Search

This field defines the maximum number of results returned from a search. The default value is 100. If the search limit exceeds the amount specified, the entries that have been found to that point will be returned.

Timeout For Search

This attribute specifies the amount of time before a timeout on a search occurs. If the search exceeds the specified time, the entries that have been found to that point will be returned

Enable LDAP SSL

This attribute specifies whether or not the LDAP server is running SSL. Selected enables SSL, unselected (default) disables SSL.

If the LDAP Server is running with SSL enabled (LDAPS), you must make sure that Access Manager is configured with proper SSL-trusted certificates so that Access Manager can connect to Directory server over LDAPS protocol.

LDAP Connection Pool Minimal Size

This attribute specifies the minimal size of connection pools to be used for connecting to the Directory Server, as specified in the LDAP server attribute. The default is 1.

LDAP Connection Pool Maximum Size

This attribute specifies the maximum size of connection pools to be used for connecting to the Directory Server, as specified in the LDAP server attribute. The default is 10.

Selected Policy Subjects

This attribute allows you to select a set of subject types available to be used for policy definition in the organization.

Selected Policy Conditions

This attribute allows you to select a set of conditions types available to be used for policy definition in the organization.

Selected Policy Referrals

This attribute allows you to select a set of referral types available to be used for policy definition in the organization.

Subjects Result Time To Live

This attribute specifies the amount of time (in minutes) that a cached subject result can be used to evaluate the same policy request based on the single sign-on token.

When a policy is initially evaluated for an SSO token, the subject instances in the policy are evaluated to determine whether the policy is applicable to a given user. The subject result, which is keyed by the SSO token ID, is cached in the policy. If another evaluation occurs for the same policy for the same SSO token ID within the time specified in the Subject Result Time To Live attribute, the policy framework retrieves the cached subjects result, instead of evaluating the subject instances. This significantly reduces the time for policy evaluation.

User Alias Enabled

This attribute must be enabled if you create a policy to protect a resource whose subject's member in a remote Directory Server aliases a local user.

This attribute must be enabled, for example, if you create `uid=rmuser` in the remote Directory Server and then add `rmuser` as an alias to a local user (such as `uid=luser`) in Access Manager. When you login as `rmuser`, a session is created with the local user (`luser`) and policy enforcement is successful.

Chapter 42

SAML Service Attributes

The Security Assertion Markup Language (SAML) Service attributes are global attributes. The values applied to them are carried across the Sun Java System Acceess Manager configuration and inherited by every configured organization. (They can not be applied directly to roles or organizations as the goal of global attributes is to customize the Access Manager application.)

For more information about the SAML Service architecture, see the *Access Manager Developer's Guide*.

The SAML attributes are as follows:

Site ID And Site Issuer Name

This attribute contains a list of entries, with each entry containing an instance ID, site ID, and site issuer name. A default value will be assigned during installation. The format is as follows:

`instanceid=serverprotocol://servername:portnumber|siteid=`*site_id*`|issuerName=`*site_issuer_name*

After configuring for this attribute for SSL (in the both source and destination site), make sure that the `instanceid` protocol is `HTTPS//`.

Sign SAML Request

This attribute specifies whether all SAML requests will be digitally signed (`XML DSIG`) before being delivered. Clicking on this option will enable this feature.

Sign SAML Response

This attribute specifies whether all SAML responses will be digitally signed (XML DSIG) before being delivered. Clicking on this option will enable this feature.

All SAML responses used by the SAML Web Post profile will be digitally signed whether this option is enabled or not enabled.

Sign Assertion

This attribute specifies whether all SAML assertions will be digitally signed (`XML DSIG`) before being delivered. Clicking on this option will enable this feature.

SAML Artifact Name

This attribute assigns a variable name to a SAML artifact defined in the SAML Service configuration. A SAML artifact is bounded-size data, which identifies an assertion and a source site. It is carried as part of a URL query string and conveyed by a re-direction to the destination site. The default is `SAMLart`. For example using the default `SAMLart` service configuration, the redirect query string could be:

`http://`*host:port/deploy_URI*`/SamlAwareServlet?TARGET=http://`*URL*`/&SAMLart=artifact123`

Target Specifier

This attribute assigns a variable name to the destination site URL used in the re-direct. The default is `Target`.

Artifact Timeout

This attribute specifies the timeout for an assertion created for an artifact. The default is 400.

Assertion Skew Factor For notBefore Time

This attribute is used to calculate the notBefore time of an assertion. For example, if the IssueInstant is `2002-09024T21:39:49Z`, and the Assertion Skew Factor notBefore Time value is set to 300 seconds (180 is the default value), the notBefore attribute of the conditions element for the assertion would be `2002-09-24T21:34:49Z`.

Assertion Timeout

This attribute specifies the number of seconds before a timeout occurs on an assertion. The default is 420.

NOTE The total valid duration of an assertion is defined by the values set in both the Assertion Skew Factor For notBefore Time and Assertion Timeout attributes.

Trusted Partner Sites

This attribute stores a partner's information so that one site can establish a trusted relationship to communicate with another partner site.

This attribute contains a list of entries, with each entry containing key / value pairs (separated by "|"). The source ID is required for each entry. For example:

`SourceID=siteid|SOAPURL=https://servername:portnumber/amserver/SAMLSOAPReceiver|AuthType=SSL|hostlist=ipaddress` (or, *server DNS name, or cert alias*)

The parameters are:

Table 42-1 Trusted Partner Sites Parameters

SourceID	The 20-byte sequence defined as in the SiteID and Issuer name.
target	This parameter is defined in a specific domain, with or without a port number. If you wish to contact a web page hosted in that specific domain, target specifies the redirect to a URL defined by the SAMLUrl or POSTUrl parameters for further processing. If there are two entries (one containing a port number and one not containing a port number) that have the same domain specified in the Trusted Partner Sites attribute, the entry with the port number has a higher priority. For example, if you have the following two trusted partner sites definitions: target=sun.com|SAMLUrl=http://machine1.sun.com:8080/amserver/SAMLAwareServlet and target=sun.com:8080|SAMLUrl=httyp://machine2.sun.com:80/amserver/SAMLAwareServlet and are seeking a the following page: http://somemachine.sun.com:8080/index.html the second definition will be chosen as the SAML service provider because the matching domain and port coexist in the target parameter.
SAMLUrl	Defines the URL that provides the SAML service. The servlet specified in the URL implements the Web-browser SSO with Artifact profile defined in the OASIS-SAML Bindings and Profiles specification.
POSTUrl	Defines the URL that provides the SAML service. The servlet specified in this URL implements the Web-browser SSO with POST profile defined in the OASIS-SAML Binding and Profiles specification.
issuer	Defines the creator of an assertion generated within Access Manager. The syntax is hostname:port.
SOAPUrl	Specifies the SOAP Receiver service URL.

AuthType	Defines the authentication type used in SAML. It should be one of the following: • NOAUTH • BASICAUTH • SSL • SSLWITHBASICAUTH This parameter is optional, and if not specified, the default is NOAUTH. If BASICAUTH or SSLWITHBASICAUTH is specified, the User parameter is require and the SOAPUrl should be HTTPS.
User	Defines the uid of the partner which is used to protect the partner's SOAP Receiver.
version	Defines the SAML version used to send SAML request. Specify either 1.0 or 1.1 for the SAML version. If this parameter is not defined, the following default values are used from AMConfig.properties: com.example.identity.saml.asertion.version=1.1 com.example.identity.saml.protocol.version=1.1
hostlist	This attribute lists the IP addresses and/or the certAlias for all of the hosts, within the specified partner site, that can send requests to this site. This ensures that the requester is indeed the intended receiver for the SAML artifact. If the requester's host or client certificate is in this list in the receiver's site, the service will continue. If the host or client certificate does not match any of those hosts or certificates in the hostlist, the SAML service will reject the request.
AccountMapper	Specifies a pluggable class which defines how the subject of an Assertion is related to an identity at the destination site. By default, it is: com.sun.identity.saml.plugins.DefaultAccountMapper
PartnerAccountMapper	The class PartnerAccountMapper is an interface that is implemented to map partner account to user account in Sun Java System Access Manager.

attributeMapper	Specifies the class with the path to where the attributeMapper is located. Applications can develop an attributeMapper to obtain either an SSOToken ID or an assertion containing AuthenticationStatement from the query. The mapper is then used to retrieve the attributes for the subject. If no attributeMapper is specified, DefaultAttributeMapper will be used.
actionMapper	Specifies the class with the path to where the actionMapper is located. Applications can develop an actionMapper to obtain either an SSOToken ID or an assertion containing AuthenticationStatement from the query. The mapper is then used to retrieve the authorization decisions for the actions defined in the query. If no actionMapper is specified, DefaultActionMapper will be used.
siteAttributeMapper	Specifies the class with the path where the siteAttributeMapper is located. Applications can develop a siteAttributeMapper to obtain attributes to be included in the assertion during SSO. If no siteAttributeMapper is found, then no attributes will be included in the assertion during SSO.
PartnerSiteAttributeMapper	This interface needs to be implemented by a partner site to return a list of Attribute objects which is requested to be returned as AttributeStatements elements, as part of the Authentication Assertion returned to the partner during the SSO scenario of Browser Artifact and POST profile.
certAlias=*aliasName*	Specifies a certAlias name used for verifying the signature in an assertion, when the assertion is signed by a partner and the certificate of the partner can not be found in the KeyInfo portion of the signed assertion.

The following table lists an example configuration for trusted partner sites. Not all of the parameters are necessary for all use cases, so the optional parameters are contained in brackets.

	Sender	Receiver
artifact	sourceid	sourceid
	target	SOAPUrl
	SAMLUrl	[accountMapper]

	Sender	Receiver
	hostlist	[AuthType]
	[siteAttributeMapper]	[User]
		[certAlias]
POST profile	sourceid	sourceid
	target	issuer
	POSTUrl	[accountMapper]
	[siteAttributeMapper]	[certAlias]
SOAP Request		sourceid
		hostlist
		[attributeMapper]
		[actionMapper]
		[certAlias]
		[issuer]

POST To Target URLs

If the target URL received through SSO (either artifact profile or POST profile) by the site is listed in this attribute, the assertion or assertions that are received from SSO will be sent to the target URL by an http: FORM POST. Avoid using test URLS or any other additional URLs in a POST.

Chapter 43

Session Service Attributes

The Session Service attributes are global and dynamic attributes. The values applied to the global attributes are applied across the Access Manager configuration and are inherited by every configured organization. (They cannot be applied directly to roles or organizations, as the goal of global attributes is to customize the Access Manager application.)

The values applied to the dynamic attributes are applied to either a role or an organization. If the role is assigned to a user or a user is assigned to the organization, these attributes, by default, are inherited by the user. Default session values are set in Service Configuration for all Access Manager registered organizations. These values can be set differently for separate organizations by registering the session service to the specific organization, creating a template and inputting a value other than the default value.

Secondary Configuration Instance

Instance Name

This field defines the name for the secondary instance.

Session Store User

This field defines the database user who is used to retrieve and store the session data.

Session Store Password

This field defines the password for the database user defined in Session Store.

Session Store Password (Confirm)

Confirm the password.

Session Cluster Server List

This attribute lists unique identifiers (two-byte values, corresponding to the entries in Server List of Platform Service) of Access Manager server instances that are participating in the same session failover cluster.

Maximum Wait Time

This field defines the total time a thread is willing to wait for acquiring a JDBC connection object. The value is in milliseconds.

JDBC Driver Implementation Class

This field specifies the name of the repository-dependent factory class used to set up the JDBC connection pool. Out of the box, Access Manager provides the implementation for the Berkeley database by Sleepycat Software, Inc.

JDBC URL

This field specifies the URL of the JDBC.

Minimum Pool Size

This attribute defines the minimum number of JDBC connections to be created in the connection pool.

Maximum Pool Size

This attribute defines the maximum number of JDBC connections to be created in the connection pool.

Global Attributes

The global attributes are:

Maximum Number of Search Results

This attribute specifies the maximum number of results returned by a session search. The default value is 120.

Timeout For Search (Seconds)

This attributed defines the maximum amount of time before a session search terminates. The default value is 5 seconds.

Dynamic Attributes

The dynamic attributes are:

Max Session Time (Minutes)

This attribute accepts a value in minutes to express the maximum time before the session expires and the user must reauthenticate to regain access. A value of 1 or higher will be accepted. The default value is `120`. (To balance the requirements of security and convenience, consider setting the Max Session Time interval to a higher value and setting the Max Idle Time interval to a relatively low value.) Max Session Time limits the validity of the session. It does not get extended beyond the configured value.

Max Idle Time (Minutes)

This attribute accepts a value (in minutes) equal to the maximum amount of time without activity before a session expires and the user must reauthenticate to regain access. A value of 1 or higher will be accepted. The default value is `30`. (To balance the requirements of security and convenience, consider setting the Max Session Time interval to a higher value and setting the Max Idle Time interval to a relatively low value.)

Max Caching Time (Minutes)

This attribute accepts a value (in minutes) equal to the maximum interval before the client contacts Access Manager to refresh cached session information. A value of 0 or higher will be accepted. The default value is `3`. It is recommended that the maximum caching time should always be less than the maximum idle time.

SOAP Binding Service Attributes

The SOAP Binding Service attributes are global attributes. The values applied to them are carried across the Sun Java System Acceess Manager configuration and inherited by every configured organization. (They can not be applied directly to roles or organizations as the goal of global attributes is to customize the Access Manager application.)

The SOAP Binding Service attributes are as follows:

Request Handler List

This attribute stores information about a Web Service Provider (WSP) deployed in Access Manager. It lists entries that contain a key / value pair (separated by " | "). For example:

```
key=disco|class=com.example.identity.liberty.ws.disco.DiscoveryService|soa
pActions=sa1 sa2 sa2
```

To add a new request handler, click the add button. The key and class parameters are required. The parameters are:

key. This defines the second part of the URI path for the SOAP endpoint of the WSP. The first part is defined as Liberty by the SOAP services. For example, if you define `disco` as the key, the SOAP endpoint for the Discovery service is:

protocol`://`*hostname:port/deloy_uri*`/Liberty/disco`

class. This parameter specifies the name of the implementation class for the WSP. The Liberty SOAP layer provides a handler interface to be implemented by each WSP to process the requested message and then return a response.

soapActions. This is an optional parameter that specifies supported SOAPActions. If this parameter is not specified, all SOAPActions are supported. If a Web Service Consumer (WSC) sends a request with an unsupported SOAPAction, the request will be rejected by the SOAP layer without passing it one to the corresponding WSP.

Web Service Authenticator

This attribute defines the implementation class for the `WebServiceAuthenicator` interface, which authenticates and generates a credential for a Web Service Consumer (WSC), based on the request.

Supported Authentication Mechanisms

This attribute specifies the authentication mechanisms supported by the SOAP endpoint. By default, all of the mechanisms are selected. If an authentication mechanism is not selected, and a WSC sends a request using that authentication mechanism, the request will be rejected by the SOAP layer without passing it to the corresponding WSP.

Chapter 45

User Attributes

There are two places which house user attributes: the Service Configuration and User Management windows. The Service Configuration window contains default attributes for registered organizations. The User Management window contains user entry attributes.

User Service Attributes

The User Service Attributes are dynamic attributes. The values applied to dynamic attributes are assigned to a role or an organization that is configured in Access Manager. When the role is assigned to a user or a user is assigned to the organization, the dynamic attributes become a characteristic of the user. The User Attributes are divided into:

- User Preferred Language
- User Preferred Timezone
- Inherited Locale
- Administrator DN Starting View
- Default User Status

Default user values are set for all Access Manager registered organizations. These values can be set differently for separate organizations by registering the user service to the specific organization, creating a template and inputting a value other than the default value.

User Preferred Language

This field specifies the user's choice for the text language displayed in the Access Manager console. The default value is `en`. This value maps a set of localization keys to the user session so that the on-screen text appears in a language appropriate for the user.

User Preferred Timezone

This field specifies the time zone in which the user accesses the Access Manager console. There is no default value.

Inherited Locale

This field specifies the locale for the user. The default value is `en_US`. Any value from Table 21-1 on page 301 can be used.

Administrator DN Starting View

If this user is a Access Manager administrator, this field specifies the node that would be the starting point displayed in the Access Manager console when this user logs in. There is no default value. A valid DN for which the user has, at the least, read access can be used.

Default User Status

This option indicates the default status for any newly created user. This status is superseded by the User Entry status. Only active users can authenticate through Access Manager. The default value is `Active`. Either of the following can be selected from the pull-down menu:

- `Active` – The user can authenticate through Access Manager.
- `Inactive` – The user cannot authenticate through Access Manager, but the user profile remains stored in the directory.

The individual user status is set by registering the User service, choosing the value, applying it to a role and adding the role to the user's profile.

User Profile Attributes

The User Profile Attributes are default attributes for user profiles. These values are set in the User Profile view by an administrator or by the user when they log on. Administrators can add their own user attributes to the user profile or create a new service. For more information see *Access Manager Developer's Guide*.

NOTE Access Manager does not enforce uniqueness for attributes within user entries. For example, `userA` and `userB` are both created in the same organization. For both, the email address attribute can be set `jimb@madisonparc.com`. The administrator can configure Sun Java System Directory Server's attribute uniqueness plug-in to help enforce unique attribute values. For more information, see Unique User IDs at the end of this chapter or the *Sun Java System Directory Server Administrator's Guide*.

First Name

This field takes the first name of the user. (The First Name value and the Last Name value identify the user in the Currently Logged In field in the upper right corner of the Access Manager console.)

Last Name

This field takes the last name of the user. (The First Name value and the Last Name value identify the user in the Currently Logged In field in the upper right corner of the Access Manager console.)

Full Name

This field takes the full name of the user.

Password

This field takes the password for the name specified in the UserId field.

Password (Confirm)

Confirmation of the password.

Email Address

This field takes the email address of the user.

Employee Number

This field takes the employee number of the user.

Telephone Number

This field takes the telephone number of the user.

Home Address

This field can take the home address of the user.

User Status

This option indicates whether the user is allowed to authenticate through Access Manager. Only active users can authenticate through Access Manager. The default value is `Active`. Either of the following can be selected from the pull-down menu:

- `Active` – The user can authenticate through Access Manager.
- `Inactive` – The user cannot authenticate through Access Manager, but the user profile remains stored in the directory.

NOTE Changing the user status to `Inactive` only affects authentication through Access Manager. The Directory Server uses the `nsAccountLock` attribute to determine user account status. User accounts inactivated for Access Manager authentication can still perform tasks that do not require Access Manager. To inactivate a user account in the directory, and not just for Access Manager authentication, set the value of `nsAccountLock` to `true`. If delegated administrators at your site will be inactivating users on a regular basis, consider adding the `nsAccountLock` attribute to the Access Manager User Profile page. See the *Access Manager Developer's Guide* for details.

Account Expiration Date

If this attribute is present, the authentication service will disallow login if the current date and time has passed the specified Account Expiration Date. The format for this attribute is as follows:

```
(mm/dd/yyyy hh:mm)
```

User Authentication Configuration

This attribute sets the authentication method for the user. The default authentication method is LDAP. One or more authentication methods can be selected by clicking the Edit link. If more than on method is selected, then the user may have to successfully authenticate to all of selected methods.

User Alias List

The field defines a list of aliases that may be applied to the user. In order to use any aliases configured in this attribute, the LDAP service has to be modified by adding the `iplanet-am-user-alias-list` attribute to the User Entry Search Attributes field in the LDAP service.

Preferred Locale

This field specifies the locale for the user. The default value is `en_US`. Any value from Table 21-1 on page 301 can be used.

You can use one of the following attributes in the pull-down menu:

- Ignore
- Customize
- Inherit

Success URL

This field accepts a list of multiple values that specify the URL to which users are redirected after successful authentication. The format of this attribute is `clientType|URL`, although you can specify only the value of the URL which assumes a default type of HTML

Failure URL

This field accepts a list of multiple values that specify the URL to which users are redirected after an unsuccessful authentication. The format of this attribute is `clientType|URL`, although you can specify only the value of the URL which assumes a default type of HTML

Unique User IDs

In order to enforce uid uniqueness within the Access Manager application, the plug-in, available in Directory Server, must be configured as follows:

```
dn: cn=uid uniqueness,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: uid uniqueness
nsslapd-pluginPath: /ids908/lib/uid-plugin.so
nsslapd-pluginInitfunc: NSUniqueAttr_Init
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled: on
nsslapd-pluginarg0: attribute=uid
nsslapd-pluginarg1: markerObjectClass=nsManagedDomain
```

```
nsslapd-plugin-depends-on-type: database
nsslapd-pluginId: NSUniqueAttr
nsslapd-pluginVersion: 6.1
nsslapd-pluginVendor: Sun | SunONE
nsslapd-pluginDescription: Enforce unique attribute values
```

It is recommended that the nsManagedDomain object class is used to mark the organization in which uid uniqueness is desired. The plug-in is not enabled by default.

To configure the uniqueness of uids per organization, either add the DN for each organization in the plug-in entry or use the marker object class option and add nsManagedDomain to each top-level organization entry.

```
nsslapd-pluginEnabled: on
nsslapd-pluginarg0: attribute=uid
nsslapd-pluginarg1: markerObjectClass=nsManagedDomain
```

Appendix A

Error Codes

This appendix provides a list of the error messages generated by Sun Java System Acceess Manager. While this list is not exhaustive, the information presented in this chapter will serve as a good starting point for common problems. The tables listed in this appendix provide the error code itself, a description and / or probable cause of the error, and describes the actions that can be taken to fix the encountered problem.

This appendix lists error codes for the following functional areas:

- Access Manager Console Errors
- Authentication Error Codes
- Policy Error Codes
- amadmin Error Codes

If you require further assistance in diagnosing errors, please contact Sun Technical Support:

http: / / www.sun.com / service / sunone / software / index.html

Access Manager Console Errors

The following table describes the error codes generated and displayed by the Access Manager Console.

Table A-1 Access Manager Console Errors

Error Message	Description/Probable Cause	Action
An error has occurred while deleting the following:	The object may have been removed by another user prior to being removed by the current user.	Redisplay the objects that you are trying to delete and try the operation again.

Table A-1 Access Manager Console Errors

Error Message	Description/Probable Cause	Action
You have entered an invalid URL	This occurs if the URL for an Access Manager console window is entered incorrectly.	
There are no entries matching the search criteria.	The parameters entered in the search window, or in the Filter fields, did not match any objects in the directory.	Run the search again with a different set of parameters
There are no attributes to display.	The selected object does not contain any editable attributes defined in its schema.	
There is no information to display for this service.	The services viewed from the Service Configuration module do not have global or organization based attributes	
Search size limit exceeded. Please refine your search.	The parameters specified in the search have returned more entries than are allowed to be returned	Modify the Maximum Results Returned from a Search attribute in the Administration service to a larger value. You can also modify the search parameters to be more restrictive.
Search time limit exceeded. Please refine your search.	The search for the specified parameters has taken longer than the allowed search time.	Modify the Timeout for Search attribute in the Administration service to a larger value. You can also modify the search parameters, so they are less restrictive, to return more values.
Invalid user's start location. Please contact your administrator.	The start location DN in the users entry is no longer valid	In the User Profile page, change the value of the start DN to a valid DN.
Could not create *identity object*. User does not have sufficient access.	An operation was executed by a user with insufficient permissions. The permissions a user has defined determines what operations they can perform.	

Authentication Error Codes

The following table describes the error codes generated by the Authentication service. These errors are displayed to the user / administrator in the Authentication module.

Table A-2 Authentication Error Codes

Error Message	Description/Probable Cause	Action
authentication.already.login.	The user has already logged in and has a valid session, but there is no Success URL redirect defined.	Either logout, or set up some login success redirect URL(s) through the Access Manager Console. Use the 'goto' query parameter with the value as Admin Console URL.
logout.failure.	A user is unable to logout of Access Manager.	Restart the server.
uncaught_exception	An authentication Exception is thrown due to an incorrect handler	Check the Login URL for any invalid or special characters.
redirect.error	Access Manager cannot redirect to Success or Failure redirect URL.	Check the web container's error log to see if there are any errors.
gotoLoginAfterFail	This link is generated when most errors occur. The link will send the user to the original Login URL page.	
invalid.password	The password entered is invalid.	Passwords must contain at least 8 characters. Check that the password contains the appropriate amount of characters and ensure that it has not expired.
auth.failed	Authentication failed. This is the generic error message displayed in the default login failed template. The most common cause is invalid/incorrect credentials.	Enter valid and correct user name/password (the credentials required by the invoked authentication module.)
nouser.profile	No user profile was found matching the the entered user name in the given organization. This error is displayed while logging in to the Membership/Self-registration authentication module.	Enter your login information again. If this is your first login attempt, select New User in the login screen.
notenough.characters	The password entered does not contain enough characters. This error is displayed while logging in to the Membership/Self-registration authentication module.	The login password must contain at least 8 characters by default (this number is configurable through the Membership Authentication module).

Table A-2 Authentication Error Codes

Error Message	Description/Probable Cause	Action
useralready.exists	A user already exists with this name in the given organization. This error is displayed while logging in to the Membership/Self-registration authentication module.	User IDs must be unique within the organization.
uidpasswd.same	The User Name and Password fields cannot have the same value. This error is displayed while logging in to the Membership/Self-registration authentication module.	Make sure that the username and password are different.
nouser.name	No user name was entered.This error is displayed while logging in to the Membership/Self-registration authentication module.	Make sure to enter the user name.
no.password	No password was entered.This error is displayed while logging in to the Membership/Self-registration authentication module.	Make sure to enter the password.
missing.confirm.passwd	Missing the confirmation password field. This error is displayed while logging in to the Membership/Self-registration authentication module.	Make sure to enter the password in the Confirm Password field.
password.mismatch	The password and the confirm password do not match. This error is displayed while logging in to the Membership/Self-registration authentication module.	Make sure that the password and confirmation password match.
An error occurred while storing the user profile.	An error occurred while storing the user profile.This error is displayed while logging in to the Membership/Self-registration authentication module.	Make sure that the attributes and elements are valid and correct for Self Registration in the `Membership.xml` file.
orginactive	This organization is not active.	Activate the organization through the Access Manager console by changing the organization status from `inactive` to `active`.
internal.auth.error	Internal Authentication Error. This is a generic Authentication error which may be caused by different and multiple environmental and/or configuration issues.	

Table A-2 Authentication Error Codes

Error Message	Description/Probable Cause	Action
usernot.active	The user no longer has an active status.	Activate the user through the Admin Console by changing the user status from `inactive` to `active`. if the user is locked out by Memory Locking, restart the server.
user.not.inrole	User does not belong to the specified role. This error is displayed during role-based authentication.	Make sure that the login user belongs to the role specified for the role-based authentication.
session.timeout	The user session has timed out.	Login in again.
authmodule.denied	The specified authentication module is denied.	Make sure that the required authentication module is registered under the required organization, that the template is created and saved for the module, and that the module is selected in the Organization Authentication Modules list in the Core Authentication module.
noconfig.found	No configuration found.	Check the Authentication Configuration service for the required authentication method.
cookie.notpersistent	Persistent Cookie Username does not exist in the Persistent Cookie Domain.	
nosuch.domain	The organization found.	Make sure that the requested organization is valid and correct.
userhasnoprofile.org	User has no profile in the specified organization.	Make sure that the user exists and is valid in the specified organization in the local Directory Server.
reqfield.missing	One of the required fields was not completed. Please make sure all required fields are entered.	Make sure that all required fields are entered.
session.max.limit	Maximum Sessions Limit Reached.	Logout and login again.

Policy Error Codes

The following table describes the error codes generated by the Policy framework and displayed in the Access Manager Console.

Table A-3 Policy Error Codes

Error Message	Description/Probable Cause	Action
illegal_character_/_in_name	Illegal character "/" in the policy name.	Make sure that the policy name does not contain the '/' character.
policy_already_exists_in_org	A rule with the same name already exists.	Use a different name for policy creation.
rule_name_already_present	Another rule with the given name already exists	Use a different rule name for policy creation.
rule_already_present	A rule with the same rule value already exists.	Use a different rule value.
no_referral_can_not_create_policy	No referral exists to the organization.	In order to create policies under a sub organization, you must create a referral policy at its parent organization to indicate what resources can be referred to this sub organization.
ldap_search_exceed_size_limit	LDAP search size limit exceeded. An error occurred because the search found more than the maximum number of results.	Change the search pattern or policy configuration of the organization for the search control parameters.The Search Size Limit is located in the Policy Configuration service.
ldap_search_exceed_time_limit	LDAP search time limit exceeded. An error occurred because the search found more than the maximum number of results.	Change the search pattern or policy configuration of the organization for the search control parameters.The Search Time Limit is located in the Policy Configuration service.
ldap_invalid_password	Invalid LDAP Bind password.	The password for LDAP Bind user defined in Policy Configuration is incorrect. This leads to the inability to get an authenticated LDAP connection to perform policy operations.
app_sso_token_invalid	Application SSO token is invalid.	The server could not validate the Application SSO token. Most likely the SSO token is expired.

Table A-3 Policy Error Codes

Error Message	Description/Probable Cause	Action
user_sso_token_invalid	User SSO token is invalid.	The server could not validate the User SSO token. Most likely the SSO token is expired.
property_is_not_an_Integer	Property value not an integer.	The value for this plugin's property should be an integer.
property_value_not_defined	Property value should be defined.	Provide a value for the given property.
start_ip_can_not_be_greater_than_end_ip	Start IP is larger than End IP	An attempt was made to set end IP Address to be larger than start IP Address in IP Address condition. The Start IP cannot be larger than the End IP.
start_date_can_not_be_larger_than_end_date	Start Date is larger than End Date	An attempt was made to set end Date to be larger than start Date in the policy's Time Condition. The Start Date cannot be larger than the End Date.
policy_not_found_in_organization	Policy not found in organization. An error occurred trying to locate a non-existing policy in an organization.	Make sure that the policy exists under the specified organization.
insufficient_access_rights	User does not have sufficient access. The user does not have sufficient right to perform policy operations.	Perform policy operations with the user who has appropriate access rights.
invalid_ldap_server_host	Invalid LDAP Server host.	Change the invalid LDAP Server host that was entered in the Policy Configuration service.

amadmin Error Codes

The following table describes the error codes generated by the `amadmin` command line tool to Access Manager's debug file.

Table A-4 amadmin error codes

Error Message	Code	Description/Probable Cause	Action
nocomptype	1	Too few arguments.	Make sure that the mandatory arguments (`--runasdn`, `--password`, `--passwordfile`, `--schema`, `--data`, and `--addAttributes`) and their values are supplied in the command line.
file	2	The input XML file was not found.	Check the syntax and make sure that the input XML is valid.
nodnforadmin	3	The user DN for the `--runasdn` value is missing.	Provide the user DN as the value for `--runasdn`.
noservicename	4	The service name for the `--deletservice` value is missing.	Provide the service name as the value for `--deleteservice`.
nopwdforadmin	5	The password for the `--password` value is missing.	Provide the password as the value for `--password`.
nolocalename	6	The locale name was not provided. The locale will default to en_US.	See Default Authentication Locale for a list of locales.
nofile	7	Missing XML input file.	Provide at least one input XML filename to process.
invopt	8	One or more arguments are incorrect.	Check that all arguments are valid. For a set of valid arguments, type `amadmin --help`.
oprfailed	9	Operation failed.	When `amadmin` fails, it produces more precise error codes to indicate the specific error. Refer to those error codes to evaluate the problem.
execfailed	10	Cannot process requests.	When `amadmin` fails, it produces more precise error codes to indicate the specific error. Refer to those error codes to evaluate the problem.
policycreatexception	12	Policy cannot be created.	`amadmin` produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.

Table A-4 amadmin error codes

Error Message	Code	Description/Probable Cause	Action
policydelexception	13	Policy cannot be deleted.	`amadmin` produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
smsdelexception	14	Service cannot be deleted.	`amadmin` produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
ldapauthfail	15	Cannot authenticate user.	Make sure the user DN and password are correct.
parserror	16	Cannot parse the input XML file.	Make sure that the XML is formatted correctly and adheres to the `amAdmin.dtd`.
parseiniterror	17	Cannot parse due to an application error or a parser initialization error.	Make sure that the XML is formatted correctly and adheres to the `amAdmin.dtd`.
parsebuilterror	18	Cannot parse because a parser with specified options cannot be built.	`amadmin` produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
ioexception	19	Cannot read the input XML file.	`amadmin` produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
fatalvalidationerror	20	Cannot parse because the XML file is not a valid file.	Check the syntax and make sure that the input XML is valid.
nonfatalvalidationerror	21	Cannot parse because the XML file is not a valid file.	`amadmin` produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
validwarn	22	XML file validation warnings for the file.	`amadmin` produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
failedToProcessXML	23	Cannot process the XML file.	`amadmin` produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
nodataschemawarning	24	Neither `--data` or `--schema` options are in the command.	Check that all arguments are valid. For a set of valid arguments, type `amadmin --help`.

Table A-4 amadmin error codes

Error Message	Code	Description/Probable Cause	Action
doctyperror	25	The XML file does not follow the correct DTD.	Check the XML file for the DOCTYPE element.
statusmsg9	26	LDAP Authentication failed due to invalid DN, password, hostname, or portnumber.	Make sure the user DN and password are correct.
statusmsg13	28	Service Manager exception (SSO exception).	amadmin produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
statusmsg14	29	Service Manager exception.	amadmin produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
statusmsg15	30	Schema file inputstream exception.	amadmin produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
statusmsg30	31	Policy Manager exception (SSO exception).	amadmin produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
statusmsg31	32	Policy Manager exception.	amadmin produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
dbugerror	33	More than one debug option is specified.	Only one debug option should be specified.
loginFalied	34	Login failed.	amadmin produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
levelerr	36	Invalid attribute value.	Check the level set for the LDAP search. It should be either SCOPE_SUB or SCOPE_ONE.
failToGetObjType	37	Error in getting object type.	Make sure that the DN in the XML file is value and contains the correct object type.
invalidOrgDN	38	Invalid organization DN.	Make sure that the DN in the XML file is valid and is an organization object.

Table A-4 amadmin error codes

Error Message	Code	Description/Probable Cause	Action
invalidRoleDN	39	Invalid role DN.	Make sure that the DN in the XML file is valid and is a role object.
invalidStaticGroupDN	40	Invalid static group DN.	Make sure that the DN in the XML file is valid and is a static group object.
invalidPeopleContainerDN	41	Invalid people container DN.	Make sure the DN in the XML file is valid and is a people container object.
invalidOrgUnitDN	42	Invalid organizational unit DN.	Make sure that the DN in the XML file is valid and is a container object.
invalidServiceHostName	43	Invalid service host name.	Make sure that the hostname for retrieving valid sessions is correct.
subschemaexception	44	Subschema error.	Subcschema is only supported for global and organization attributes.
serviceschemaexception	45	Cannot locate service schema for service.	Make sure that the sub schema in the XML file is valid.
roletemplateexception	46	The role template can be true only if the schema type is dynamic.	Make sure that the role template in the XML file is valid.
cannotAddusersToFileredRole	47	Cannot add users to a filtered role.	Made sure that the role DN in the XML file is not a filtered role.
templateDoesNotExist	48	Template does not exist.	Make sure that the service template in the XML file is valid.
cannotAdduUersToDynamicGroup	49	Cannot add users to a dynamic group.	Made sure that the group DN in the XML file is not a dynamic group.
cannotCreatePolicyUnderContainer	50	Policies can not be created in an organization that is a child organization of a container.	Make sure that the organization in which the policy is to be created is not a child of a container.
defaultGroupContainerNotFound	51	The group container was not found.	Create a group container for the parent organization or container.
cannotRemoveUserFromFilteredRole	52	Cannot remove a user from a filtered role.	Make sure that the role DN in the XML file is not filtered role.
cannotRemoveUsersFromDynamicGroup	53	Cannot remove users from a dynamic group.	Make sure that the group DN in the XML file is not a dynamic group.
subSchemStringDoesNotExist	54	The subschema string does not exist.	Make sure that the subschema string exists in the XML file.

Table A-4 amadmin error codes

Error Message	Code	Description/Probable Cause	Action
defaultPeopleContainerNotFound	59	You are trying to add user to an organization or container. And default people container does not exists in an organization or container.	Make sure the default people container exists.
nodefaulturlprefix	60	Default URL prefix is not found following --defaultURLPrefix argument	provide the default URL prefix accordingly.
nometaalias	61	Meta Alias is not found following --metaalias argument	provide the Meta Alias accordingly.
missingEntityName	62	Entity Name is not specified.	provide the entity name.
missingLibertyMetaInputFile	63	File name for importing meta data is missing.	provide the file name that contains meta data.
missingLibertyMetaOutputFile	64	File name for storing exported meta data is missing.	provide the file name for storing meta data.
cannotObtainMetaHandler	65	Unable to get a handler to Meta attribute. Specified user name and password may be incorrect.	ensure that user name and password are correct.
missingResourceBundleName	66	Missing resource bundle name when adding, viewing or deleting resource bundle that is store in directory server.	provide the resource bundle name
missingResourceFileName	67	Missing file name of file that contains the resource strings when adding resource bundle to directory server.	Please provide a valid file name.
failLoadLibertyMeta	68	Failed to load liberty meta to Directory Server.	Please check the meta data again before loading it again

Glossary

For a list of terms used in this documentation set, refer to the latest *Sun Java™ Enterprise System Glossary*:

http://docs.sun.com/doc/816-6873

SYMBOLS

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W