Sun Java™ System

Access Manager 6 Migration Guide

2005Q1

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

Part No: 817-7645-10

Contents

List of Figures

Preface

The *Sun Java™ System Access Manager Migration Guide* describes how to upgrade to Sun Java System Access Manager 6 2005Q1 from previous versions of Access Manager, including Sun Java System Identity Server, Sun™ Open Net Environment (Sun ONE) Identity Server, and iPlanet™ Directory Server Access Management Edition (DSAME).

This preface contains the following sections:

Who Should Use This Guide

The *Migration Guide* is intended for system administrators who are upgrading previous versions of Access Manager to Access Manager 6 2005Q1. Readers should understand the following technologies:

- Sun Java Enterprise System
- Web container that is running Access Manager: Sun Java System Application Server, Sun Java System Web Server, BEA WebLogic, or IBM WebSphere Application Server.
- Solaris™ or Linux operating system concepts
- Lightweight Directory Access Protocol (LDAP) directory server concepts
- Java™ technology
- JavaServer Pages™ (JSP) technology
- HyperText Transfer Protocol (HTTP)
- HyperText Markup Language (HTML)
- eXtensible Markup Language (XML)

Before You Read This Guide

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`

How This Guide Is Organized

The following table summarizes the contents of this guide:

Table 1 Organization of the Access Manager Migration Guide

Chapter or Appendix	Description
Chapter 1, "Introduction"	Introduces the Access Manager upgrade process and include an upgrade roadmap.
Chapter 2, "Upgrading to Access Manager 6 2005Q1"	Describes how to upgrade to Access Manager 6 2005Q1 from previous versions of Access Manager, including Sun Java™ System Identity Server, Sun™ Open Net Environment (Sun ONE) Identity Server, and iPlanet™ Directory Server Access Management Edition (DSAME).
Chapter 3, "Configuring Access Manager With an Existing Directory Server"	Describes how to configure Access Manager with an existing Directory Server that is provisioned with data.
Appendix A, "Access Manager Upgrade Worksheets"	Provides worksheets that you can use to plan your upgrade.
Glossary	Provides a link to the latest *Sun Java™ Enterprise System Glossary*.

Conventions Used in This Guide

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

Typographic Conventions

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

Table 2 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.`

`AaBbCc123` (Monospace bold)	What you type, when contrasted with onscreen computer output.	`%` **`su`**
		`Password:`

Table 2 Typographic Conventions *(Continued)*

Typeface	Meaning	Examples
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 guide.

Table 3 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 guide:

Table 4 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 guide.

Table 5 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 6 Access Manager 6 2005Q1 Documentation Set

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 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 (this 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 6 Access Manager 6 2005Q1 Documentation Set *(Continued)*

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/ApplicationServer8_ee_04q4
- 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/

Sun Java System Services Suite
http://www.sun.com/service/sunjavasystem/sjsservicessuite.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 guide is *Sun Java System Access Manager 6 2005Q1 Migration Guide*, and the part number is 817-7645.

Chapter 1

Introduction

This guide describes how to upgrade to Sun Java™ System Access Manager 6 2005Q1 from previous versions of Access Manager, including Sun Java™ System Identity Server, Sun™ Open Net Environment (Sun ONE) Identity Server, and iPlanet™ Directory Server Access Management Edition (DSAME).

Topics include:

- "Access Manager Upgrade Roadmap" on page 20.
- Chapter 2, "Upgrading to Access Manager 6 2005Q1" describes how to upgrade to Sun Java™ System Access Manager 6 2005Q1 from previous versions.
- Chapter 3, "Configuring Access Manager With an Existing Directory Server" describes how to install and configure Access Manager 6 2005Q1 with a directory that is already provisioned with user data.
- Appendix A, "Access Manager Upgrade Worksheets" is provided to help you to plan and perform an upgrade to Access Manager 6 2005Q1.

Access Manager Upgrade Roadmap

Table 1-1 shows how to upgrade previous versions of Access Manger.

Table 1-1 Access Manager 6 2005Q1 Upgrade Roadmap

Previous Version	How to Upgrade to Access Manager 6 2005Q1
Sun Java System Identity Server 2004Q2 (6.2)	Follow the steps in "Upgrading Identity Server 2004Q2" on page 26 in this guide.
Sun Java System Identity Server 2004Q2 (6.2) SP1	Back out SP1 and then follow the steps in "Upgrading Identity Server 2004Q2" on page 26 in this guide.
Sun ONE Identity Server 6.1	Follow the steps in "Upgrading Identity Server 6.1" on page 30 in this guide. You can also upgrade from Identity Server 6.1 to Identity Server 2004Q2 (6.2) and then to Access Manager 6 2005Q1 (6.3): 1. Upgrade to Identity Server 2004Q2 (6.2), as described in Chapter 1 of the *Sun Java System Identity Server 2004Q2 Migration Guide*: `http://docs.sun.com/doc/817-5708` 2. Upgrade to Access Manager 6 2005Q1, as described in "Upgrading Identity Server 2004Q2" on page 26 in this guide.
Sun ONE Identity Server 6.0 or 6.0 SP 1 or iPlanet Directory Server Access Management Edition (DSAME) 5.1	Upgrade to Identity Server 6.1, by following the process in the *Sun ONE Identity Server 6.1 Migration Guide*: `http://docs.sun.com/doc/816-6771-10` After you upgrade to Identity Server 6.1, follow the steps in "Upgrading Identity Server 6.1" on page 30 in this guide.

Chapter 2

Upgrading to Access Manager 6 2005Q1

This chapter describes how to upgrade to Sun Java™ System Access Manager 6 2005Q1 from previous versions of Access Manager:

- "Before You Begin the Access Manager Upgrade" on page 22
- "Upgrading Identity Server 2004Q2" on page 26
- "Upgrading Identity Server 6.1" on page 30
- "Upgrading Multiple Instances" on page 37
- "Verifying the Upgrade" on page 38
- "Upgrading an Access Manager SDK Installation" on page 35
- "Access Manager Coexistence" on page 39

For information about upgrading other Sun Java Enterprise System components, see the *Sun Java Enterprise System 2005Q1 Upgrade and Migration Guide* (`http://docs.sun.com/doc/819-0062`).

For a roadmap about upgrading Access Manager, see the "Access Manager Upgrade Roadmap" on page 20.

Before You Begin the Access Manager Upgrade

Before you upgrade Access Manager, perform these preliminary steps:

1. Obtain the Java Enterprise System 2005Q1 Installation Software
2. Obtain All Required Patches
3. Obtain the Required Information and Passwords
4. Back Up Your Directory Server Data
5. Back Up Any Web Container Customized Files
6. Upgrade the Shared Components
7. Upgrade the Web Container Software
8. Use a Non-SSL Port for Directory Server
9. Upgrade Directory Server (Optional)

Obtain the Java Enterprise System 2005Q1 Installation Software

Obtain the Sun Java Enterprise System (Java ES) 2005Q1 installation software. You can download the software from the Sun Download Center:

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

Or, request a media kit containing the software on CDs or a DVD from your Sun sales representative.

For more information about obtaining the Java ES installation software, see the *Sun Java Enterprise System 2005Q1 Installation Guide* (http://docs.sun.com/doc/819-0056).

Obtain All Required Patches

If you plan to upgrade to Access Manager 6 2005Q1, you need the following patches:

- Solaris™ OS, SPARC® Platform Edition: 118217, 118218, 117112, 117585
- Solaris OS, x86 Platform Edition: 118217, 118218, 117584, 117585

  Note 118217, 118218 and 117585 are common patches that applies to both the SPARC and x86 platforms. Apply patches 118217 and 118218 first, before you apply 117585.

- Linux OS: 17588 (patch that contains the required Linux RPM packages)
- Shared components: See "Upgrade the Shared Components" on page 24.

For information about other patches that might be required, see the Access Manager *Release Notes* (`http://docs.sun.com/doc/817-7642`).

To obtain the required patches, download them from the SunSolve site:

`http://sunsolve.sun.com/`

Obtain the Required Information and Passwords

To upgrade Access Manager, you must provide specific information, including administrator names and passwords. For example, you must know the Access Manager administrator and password and Directory Manager name and password for the Directory Server that Access Manager is using.

Before you upgrade, refer to the worksheets in Appendix A, "Access Manager Upgrade Worksheets" to record the required information.

Back Up Your Directory Server Data

The upgrade process uses scripts that modify the Directory Server schema (DIT). Therefore, before you upgrade, back up your Directory Server data using the Directory Server Console or a command-line utility such as `db2bak`.

For more information about backing up Directory Server, see the *Sun Java System Directory Server Administration Guide* (`http://docs.sun.com/doc/817-7613`).

Back Up Any Web Container Customized Files

Before you upgrade, back up any web container customized files related to previous versions of Access Manager, including:

- Customized console JSP pages
- Customized authentication JSP pages
- JAR files for authentication and customized modules
- Customized XML files in `/etc/opt/SUNWam/config/xml` on Solaris systems or `/etc/opt/sun/identity/config/xml` on Linux systems.

Tip Make a list of your customizations so you can redo them after you upgrade and then verify that they work correctly.

Upgrade the Shared Components

Patches to upgrade the shared components are not required to upgrade Access Manager, but they are required when you upgrade other Java ES components such as the Access Manager web containers.

Note: If you upgrade to JDK 1.5 you must upgrade the Netscape Security Services (NSS) and Java Security Services (JSS) packages, including SUNWtls, SUNWjss, and SUNWpr, by applying the shared component cluster for your specific operating system.

For information about upgrading the shared components, see the *Sun Java Enterprise System 2005Q1 Upgrade and Migration Guide* (`http://docs.sun.com/doc/819-0062`).

Upgrade the Web Container Software

If you are upgrading both the web container (Web Server or Application Server) and Access Manager, upgrade the web container first, or the Access Manager `amconfig` script will configure and redeploy Access Manager to the existing (old) web container. Access Manager 6 2005Q1 supports these web containers:

- Sun Java System Web Server 6.1 2005Q1 SP4
- Sun Java System Application Server Enterprise Edition 8.1 2005Q1

For information about upgrading the web container, refer to the respective web container documentation in the *Sun Java Enterprise System 2005Q1 Upgrade and Migration Guide* (`http://docs.sun.com/doc/819-0062`).

Also, if you saved any customized files under "Back Up Any Web Container Customized Files" on page 24, redo the customizations after you upgrade the web container.

Use a Non-SSL Port for Directory Server

When you upgrade Access Manager, the upgrade process does not finish successfully if you specify the Directory Server SSL port (for example, the default value of 636) when you run the `pre61to62upgrade`, `Upgrade61DitTo62`, or `amupgrade` script.

Therefore, when you run these scripts, specify a non-SSL port such as the 389 default value.

Upgrade Directory Server (Optional)

Upgrading Directory Server is optional. To upgrade from Identity Server 2004Q2 to Access Manager 6 2005Q1, you can be running either of these versions:

- Directory Server 5.1 SP1 or higher
- Directory Server 5.2

If you plan to upgrade Directory Server, refer to the *Sun Java Enterprise System 2005Q1 Upgrade and Migration Guide* (`http://docs.sun.com/doc/819-0062`).

Upgrading Identity Server 2004Q2

In this scenario, you want to upgrade Identity Server 2004Q2 (6.2) or Identity Server 2004Q2 (6.2) SP1 to Access Manager 6 2005Q1 (6.3).

To upgrade Identity Server 2004Q2 to Access Manager 6 2005Q1

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

2. Make sure you have performed the steps listed under "Before You Begin the Access Manager Upgrade" on page 22.

3. If you have installed Identity Server 2004Q2 SP1, you must first back out SP1 before you apply the upgrade patches.

   To determine the release you are running use the `amserver version` command on either a Solaris or Linux system. On Solaris systems, you can also use the showrev command with the -p option to display patch information. For example:

   ```
   # showrev -p | grep SUNWam
   ```

4. On the Solaris 8 or 9 SPARC and x86 platforms, remove the `SUNWamjwsdp` Solaris package. On Linux systems, remove the `sun-identity-jwsdp` RPM package. For example, on a Solaris system:

   ```
   # pkgrm SUNWamjwsdp
   ```

   These packages contain Access Manager 2004Q2 (6.2) components such as JAXP and JAXB for the Java Web Services Developer Pack (JWSDP). Access Manager 2005Q1 (6.3) uses the Java ES shared component packages and RPMs for the JWSDP products instead of bundling its own.

5. Apply the following Access Manager upgrade patches or RPMs, depending on your platform. If you have a multi-server configuration, apply the respective patches or RPMs to each server running an instance of Access Manager.

   - Solaris™ OS, SPARC® Platform Edition: 118217, 118218, 117112, 117585
   - Solaris OS, x86 Platform Edition: 118217, 118218, 117584, 117585

   Note 118217, 118218 and 117585 are common patches that applies to both the SPARC and x86 platforms. Apply patches 118217 and 118218 first, before you apply 117585.

- Linux OS: 117588 (patch that contains the required Linux RPMs)
  To upgrade:

  a. Unzip the 117588 patch file.

  b. Read the README file.

  c. Run the `installpatch` script, which adds the RPMs.

6. Reapply any customized JSPs for the Access Manager console and authentication user interface (UI) that you saved under Back Up Any Web Container Customized Files. Then, copy the customized JSP files to the correct directories. For example, on Solaris systems:

   - Console: *AccessManager-Base*`/SUNWam/web-src/applications/console`

   - Authentication UI:
     AccessManager-Base`/SUNWam/web-src/services/config/auth/default`

   For more information, see the *Sun Java System Access Manager Developer's Guide* (`http://docs.sun.com/doc/817-7649`).

7. Configure Access Manager for your specific web container by running the `amconfig` script.

   Note Before you run `amconfig`, make sure that you have upgraded the Access Manager web container, as described in “Upgrade the Web Container Software” on page 25.

   Before you run `amconfig`, set the configuration variables in the configuration script input file, which is based on the `amsamplesilent` template file:

   - Set DEPLOY_LEVEL=21 and DIRECTORY_MODE=4.

   - The default JDK version for Sun Java Enterprise System 2005Q1 release is 1.5, so make sure you set the JAVA_HOME variable in the configuration script input file to the correct directory.

   - Make sure to set the AM_ENC_PWD variable to the same value you specified when you ran the Java ES installer (which is also the value of the `am.encryption.pwd` parameter in the `AMConfig.properties` file.

- For other values in the configuration script input file, provide the same values that were used for the Identity Server 6.1 configuration that you are upgrading (unless you have changed specific items such as your web container or passwords).

To set the configuration values, consider using the worksheets in Appendix A, "Access Manager Upgrade Worksheets."

The `amconfig` script and the `amsamplesilent` file are installed in the following directories:

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

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

For example, to run amconfig on a Solaris system with Access Manager installed in the base installation directory:

```
# cd /opt/SUNWam/bin
# ./amconfig -s config-file
```

where `config-file` is the configuration script input file.

For information about the `amconfig` script and the `amsamplesilent` file, see the *Sun Java System Access Manager Administration Guide* (`http://docs.sun.com/doc/817-7647`).

8. Upgrade the Access Manager schema (DIT) to Access Manager 6 2005Q1 by running the `amupgrade` script, which is installed in the following directory:

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

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

Before you run `amupgrade`, you will need to know the following information:

- Fully-qualified host name and non-SSL port number of the Directory Server that Access Manager is using
- Directory Manager name (default: cn=Directory Manager) and password for the Directory Server
- Access Manager administrator (default: amadmin) and password

Run the amupgrade script. For example, on Solaris systems:

```
# cd /opt/SUNWam/upgrade/scripts
# ./amupgrade
```

If the upgrade is successful, the script displays “Upgrade completed.”

9. The `amupgrade` script writes status information to the following log file:

   `/var/sadm/install/logs/Sun_Java_System_Identity_Server_upgrade_dit_log.`*mmddhhmm*

   Check this log file for information about the upgrade.

10. Restart the Access Manager web container for the upgrade changes to take effect.

11. If you are using the Security Assertion Markup Language (SAML) service, you must add and enable the SAML authentication module using the Access Manager console. For the steps involved, refer to the *Sun Java System Access Manager Administration Guide* (`http://docs.sun.com/doc/817-7647`).

NOTE In the Access Manager 6 2005Q1 release, the default value for the “Default success login URL” attribute in the core service has changed from “`%protocol://%host:%port/amconsole`” to “`/amconsole`”.

Consequently, The `%protocol`, `%host` and `%port` variables are not supported. For a remote console, you must modify the “Default success login URL” to point to the console page on the actual remote console host, if the console page is expected after a login.

Upgrading Identity Server 6.1

In this scenario, you want to upgrade Identity Server 6.1 to Access Manager 6 2005Q1.

To upgrade Identity Server 6.1 to Access Manager 6 2005Q1

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

2. Make sure you have performed any required steps listed under "Before You Begin the Access Manager Upgrade" on page 22.

3. Before you run the pre-upgrade script, consider using the "Pre-Upgrade Script Worksheet" on page 88 to record the information you will need to provide.

4. To run the pre-upgrade script in the next step, Directory Server must be running. To verify that Directory Server is running:

   ```
   # ps -ef | grep slapd
   ```

   If Directory Server is not running, start it. For example:

   ```
   # cd /var/opt/mps/serverroot/slapd-instance-name
   # ./start-slapd
   ```

5. Run the Identity Server 2004Q2 pre-upgrade script (`pre61to62upgrade`) to perform these functions:

   - Backs up Identity Server 2004Q2 by running the `am2bak` script
   - Removes the Identity Server 2004Q2 packages (but not the Directory Server or web container packages) and then updates the `/var/sadm/install/productregistry` file to reflect that the packages have been removed
   - Writes the `Sun_Java_System_Identity_Server_upgrade_log.`*timestamp* log file to the `/var/sadm/install/logs` directory

   The `pre61to62upgrade` script is part of the Java ES installation software and is available in the following directory:

   JavaES-base`/Solaris_sparc/Product/identity_srv/Tools`

   The *JavaES-base* is the directory where you uncompressed the archive. For example:

   ```
   # cd JavaES2005Q1/Solaris_sparc/Product/identity_srv/Tools
   # ./pre61to62upgrade
   ```

6. When you are prompted by the script, enter the following information:

   - Directory Server fully qualified host name. For example: `ds.example.com`
   - Directory Server non-SSL port number. Default is 389.
   - Distinguished name (DN) and password of the top-level Identity Server administrator. For example: `uid=amAdmin,ou=People,dc=example,dc=com`
   - Directory where the script should back up the Identity Server 6.1 files. For example: `/opt/is_backup`
   - Certificate directory of the web container. For example: `/opt/SUNWwbsvr/alias`

7. Install Access Manager 6 2005Q1 by running the Java ES 2005Q1 installer. On the Configuration Type panel, choose the Configure Later option.

   The Java ES installer then installs the component packages but does not configure the components. For information about the Java ES installer, refer to the *Sun Java Enterprise System 2005Q1 Installation Guide* (`http://docs.sun.com/doc/819-0056`).

8. Configure Access Manager for your specific web container by running the `amconfig` script.

   Note Before you run `amconfig`, make sure that you have upgraded the Access Manager web container, as described in "Upgrade the Web Container Software" on page 25.

   - Set DEPLOY_LEVEL=21 and DIRECTORY_MODE=4.
   - The default JDK version for Sun Java Enterprise System 2005Q1 release is 1.5, so make sure you set the JAVA_HOME variable in the configuration script input file to the correct directory.
   - Make sure to set the AM_ENC_PWD variable to the same value you specified when you ran the Java ES installer (which is also the value of the `am.encryption.pwd` parameter in the `AMConfig.properties` file.

- For other values in the configuration script input file, provide the same values that were used for the Identity Server 6.1 configuration that you are upgrading (unless you have changed specific items such as your web container or passwords).

To set the configuration values, consider using the worksheets in Appendix A, "Access Manager Upgrade Worksheets."

The `amconfig` script and the `amsamplesilent` file are installed in the following directories:

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

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

For information about the `amconfig` script and the `amsamplesilent` file, see the *Sun Java System Access Manager Administration Guide* (`http://docs.sun.com/doc/817-7647`).

9. To run the post-upgrade script in the next step, Directory Server must be running. To verify that Directory Server is running:

```
# ps -ef | grep slapd
```

If Directory Server is not running, start it. For example:

```
# cd /var/opt/mps/serverroot/slapd-instance-name
# ./start-slapd
```

10. Run the Identity Server 2004Q2 post-upgrade script (`Upgrade61DitTo62`) to upgrade the Directory Server schema (DIT) to Identity Server 2004Q2.

This script is available in the following directories:

- Solaris systems: *AccessManager-base*`/SUNWam/migration/61to62/scripts`
- Linux systems: *AccessManager-base*`/identity/migration/61to62/scripts`

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

For example, to run the script on Solaris systems:

```
# cd opt/SUNWam/migration/61to62/scripts
# ./Upgrade61DitTo62
```

11. When you are prompted by the `Upgrade61DitTo62` script, provide the following information:

    - Directory Server fully qualified host name. For example: `ds.example.com`
    - Directory Server non-SSL port number. Default is 389.
    - Distinguished name (DN) and password of the Directory Manager
    - Distinguished name (DN) and password of the top-level Identity Server administrator. For example: `uid=amAdmin,ou=People,dc=example,dc=com`

12. When you are prompted by the `Upgrade61DitTo62` script, restart Directory Server. The script pauses for you to perform the restart.

13. After the `Upgrade61DitTo62` script finishes, restart both Directory Server and the web container for the schema changes to take effect.

14. Upgrade the Access Manager schema (DIT) to Access Manager 6 2005Q1 by running the `amupgrade` script, which is installed in the following directory:

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

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

    Before you run `amupgrade`, you will need to know the following information:

    - Fully-qualified host name and non-SSL port number of the Directory Server that Access Manager is using
    - Directory Manager name (default: cn=Directory Manager) and password for the Directory Server
    - Access Manager administrator (default: amadmin) and password

    Run the amupgrade script. For example, on Solaris systems:

    ```
    # cd opt/SUNWam/upgrade/scripts
    # ./amupgrade
    ```

    If the upgrade is successful, the script displays “Upgrade completed.”

15. The `amupgrade` script writes status information to the following log file:

    `/var/sadm/install/logs/Sun_Java_System_Identity_Server_upgrade_dit_log.`*mmddhhmm*

    Check this log file for information about the upgrade.

16. If you are using the Security Assertion Markup Language (SAML) service, you must add and enable the SAML authentication module using the Access Manager console. For the steps involved, refer to the *Sun Java System Access Manager Administration Guide* (`http://docs.sun.com/doc/817-7647`).

You have now upgraded to Access Manager 6 2005Q1.

Upgrading an Access Manager SDK Installation

This section describes how to upgrade an SDK only installation to the Access Manager 6 2005Q1 SDK, including:

- To upgrade an Identity Server 6.1 SDK only installation
- To upgrade an Identity Server 2004Q2 (6.2) SDK only installation

CAUTION The SDK upgrade process will not affect your user data; however, before you upgrade, back up your `AMConfig.properties` and `serverconfig.xml` configuration files.

To upgrade an Identity Server 6.1 SDK only installation

1. Log in as or become superuser (root).
2. Make sure you have saved the Identity Server 6.1 `AMConfig.properties` and `serverconfig.xml` configuration files.
3. Uninstall the Identity Server 6.1 SDK by following the instructions in the *Sun Java Enterprise System 2003Q4 Installation Guide* (`http://docs.sun.com/doc/816-6874`).
4. Install the Access Manager 6 2005Q1 SDK by following the instructions in the *Sun Java Enterprise System 2005Q1 Installation Guide* (`http://docs.sun.com/doc/819-0056`).

   You can also install the Identity Server 2004Q2 SDK and then apply the patches described in To upgrade an Identity Server 2004Q2 (6.2) SDK only installation.
5. Incorporate the configuration changes you saved in Step 2 into the new Access Manager 6 2005Q1 configuration files.

To upgrade an Identity Server 2004Q2 (6.2) SDK only installation

1. Make sure you have saved the Identity Server 2004Q2 `AMConfig.properties` and `serverconfig.xml` configuration files.
2. Apply the following Access Manager upgrade patches on the server where the SDK is installed, depending on your platform:
   - Solaris™ OS, SPARC® Platform Edition: 118217, 118218, 117112, 117585

- Solaris OS, x86 Platform Edition: 118217, 118218, 117584, 117585

Note 118217, 118218 and 117585 are common patches that applies to both the SPARC and x86 platforms. Apply patches 118217 and 118218 first, before you apply 117585.

- Linux OS: 117588 (patch that contains the required Linux RPMs)
  To upgrade:

  a. Unzip the 117588 patch file.

  b. Read the README file.

  c. Run the `installpatch` script, which adds the RPMs.

3. Configure the Access Manager SDK for your specific deployment by running the `amconfig` script. Before you run `amconfig`, set the configuration variables in the configuration script input file, which is based on the `amsamplesilent` template file. Set DEPLOY_LEVEL as follows:

   - DEPLOY_LEVEL=3 to upgrade the SDK only
   - DEPLOY_LEVEL=4 to upgrade the SDK and configure the web container

   For other values in the configuration script input file, provide the same values that were used for the Identity Server 6.1 SDK configuration that you are upgrading (unless you have changed specific items such as your web container or passwords).

   The default JDK version for Sun Java Enterprise System 2005Q1 release is 1.5, so make sure you set the JAVA_HOME variable in the configuration script input file to the correct directory.

   To set the configuration values, consider using the worksheets in Appendix A, "Access Manager Upgrade Worksheets."

   The `amconfig` script and the `amsamplesilent` file are installed in the following directories:

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

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

   For information about the `amconfig` script and the `amsamplesilent` file, see the *Sun Java System Access Manager Administration Guide* (`http://docs.sun.com/doc/817-7647`).

4. Incorporate the configuration changes you saved in Step 1 into the new Access Manager 6 2005Q1 configuration files.

5. If you are using the Security Assertion Markup Language (SAML) service, you must add and enable the SAML authentication module using the Access Manager console. For the steps involved, refer to the *Sun Java System Access Manager Administration Guide* (`http://docs.sun.com/doc/817-7647`).

NOTE In the Access Manager 6 2005Q1 release, the default value for the "Default success login URL" attribute in the core service has changed from "`%protocol://%host:%port/amconsole`" to "`/amconsole`".

Consequently, The `%protocol`, `%host` and `%port` variables are not supported. For a remote console, you must modify the "Default success login URL" to point to the console page on the actual remote console host, if the console page is expected after a login.

Upgrading Multiple Instances

This section describes how to upgrade multiple Identity Server instances running on different hosts that share the same Directory Server.

The upgrade process supports multiple instances of Identity Server installed on different host systems. Upgrading multiple instances of Identity Server installed on the same host system is not supported in the current release. If you have multiple instances on the same host, after you upgrade the main instance, you must then recreate the additional instances.

To Upgrade an Instance

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

2. Stop all Identity Server instances that access Directory Server. For example, on a Solaris system that uses the default installation directory:

```
# cd /opt/SUNWam/bin
# ./amserver stop
```

Stopping all instances prevents Identity Server from making changes to the Directory Server while you are performing the upgrade.

3. Start the Identity Server instance you want to upgrade.

4. Upgrade the Identity Server instance you started in Step 3, following the process shown in the "Access Manager Upgrade Roadmap" on page 20.

   During the upgrade of the first instance, the post-upgrade scripts upgrade the Directory Server to include the Access Manager 6 2005Q1 schema elements. During subsequent upgrades of other instances, however, the scripts detect that Directory Server has already been upgraded and do not try to upgrade it again.

5. Restart the instance you just upgraded.

6. Repeat Step 3 through Step 5 for each Identity Server instance on a different host that you want to upgrade.

7. If there are any Identity Server 2004Q2 instances you did not upgrade, restart those instances. For information about the co-existence of Identity Server 2004Q2 and Access Manager 6 2005Q1, see Access Manager Coexistence.

Verifying the Upgrade

After you finish the upgrade process, verify that the upgrade was successful as follows:

1. Log in to the Access Manager 6 2005Q1 console as amadmin using the following URL:

   `http://`*host-name.domain-name*`:`*port*`/amconsole`

   where *host-name.domain-name*:*port* is the fully qualified host name and port number of the web container you are using.

   Verify that new services under the "Service Configuration" tab are available.

2. Review the status of the upgrade by checking the following log files in the `/var/sadm/install/logs` directory:

   pre61to62upgrade script:
   Sun_Java_System_Identity_Server_upgrade_log.*timestamp*

Sun Java Enterprise System installer:

–Java_Shared_Component_Install.*timestamp*

–Java_Enterprise_System_install.A*timestamp*

–Java_Enterprise_System_install.B*timestamp*

–Java_Enterprise_System_Summary_Report_install.*timestamp*

Upgrade61DitTo62 script:
Sun_Java_System_Identity_Server_upgrade_dit_log.*timestamp*

amupgrade script:
Sun_Java_System_Identity_Server_upgrade_dit_log.*timestamp*

Access Manager Coexistence

The coexistence of Access Manager 6 2005Q1 and Identity Server 2004Q2 is a transitional phase during an Access Manager upgrade. These two versions can coexist and run concurrently against the same shared Directory Server, with these considerations:

- Access Manager 6 2005Q1 and Identity Server 2004Q2 must be installed on different servers.
- When you install Access Manager 6 2005Q1 using the Java ES installer, specify the Configure Later option, since you are using existing Directory Server. After installation, run the `amconfig` script to configure Access Manager and to deploy the web applications. In the `amconfig` configuration script input file (`amsamplesilent`), set DEPLOY_LEVEL=1 and DIRECTORY_MODE=4.
- If you have not upgraded Directory Server to include the Access Manager 6 2005Q1 schema elements, you can use either Access Manager 6 2005Q1 or Identity Server 2004Q2 to access the directory.
- After you have upgraded Directory Server to include the Access Manager 6 2005Q1 schema elements, you must use Access Manager 6 2005Q1 to access new the new Access Manager features, including new services, attributes in existing services, and policy plug-ins. Identity Server 2004Q2, including the console, will not function correctly with the Manager 6 2005Q1 schema.

Chapter 3

Configuring Access Manager With an Existing Directory Server

This chapter describes how to install and configure Sun Java™ System Access Manager 6 2005Q1 with an existing Directory Server that is already provisioned with user data, including:

- "Directory Server Considerations" on page 42
- "Installing Access Manager" on page 44
- "Configuring Directory Server" on page 49
- "Configuring the Directory Server Schema Files" on page 47
- "Adding Access Manager Object Classes" on page 52
- "Adding Custom Object Classes" on page 71
- "Loading Access Manager LDIF Files" on page 84
- "Enabling Access Manager User Management" on page 85
- "Verifying the Access Manager Installation" on page 85

Directory Server Considerations

Before you get started, consider the following Directory Server issues:

Administrator Privileges to Directory Server

To install, upgrade, or configure Directory Server or to migrate your existing Directory Server data, you must have the appropriate Directory Server administrator privileges.

You can make modifications by using Directory Server Console, Directory Server command-line utilities such as `ldapmodify` or `db2ldif` utilities, or the sample scripts that are included with Access Manager.

Directory Server Upgrade

If you are using an earlier version of Directory Server (before the 2005Q1 release), consider upgrading to Directory Server 5 2005Q1 and then migrating your existing data to the upgraded directory. For information about upgrading Directory Server, see the *Sun Java Enterprise System 2005Q1 Upgrade and Migration Guide* (`http://docs.sun.com/doc/819-0062`).

Directory Server Backup

Before you make changes to your existing Directory Server data, back up your data using a utility such as `db2ldif` or `db2bak` or the Directory Server Console. For information about backing up your directory, refer to the Directory Server Administration Guide in the "Directory Server Online Documentation" on page 43.

Directory Server Utilities

Make sure that you're using the correct version of Directory Server utilities such as `ldapmodify` and `ldapsearch`. To determine which utilities you are using, use the `which` command.

Do not use the Directory Server utilities included with either the Solaris or Linux operating system, which are in the `/bin` or `/usr/bin` directory. These directories might not contain the most recent version of these utilities.

Instead, use the Directory Server utilities in either of these directories:

- Utilities included with Access Manager:

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

- Utilities included with Directory Server:

  Solaris systems: *DirectoryServer-base*/shared/bin/
  Linux systems: *DirectoryServer-base*/sun/directory-server/shared/bin

Directory Server Libraries

To access the Directory Server libraries that are included with either Directory Server or Access Manager, set the LD_LIBRARY_PATH environment variable accordingly. For example, on Solaris systems, add *AccessManager-base*/SUNWam/ldaplib/ldapsdk and /usr/lib/mps/secv1 to your LD_LIBRARY_PATH to access the libraries included with Access Manager.

Directory Server Online Documentation

Directory Server online documentation is available on the following Web site:

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

Installing Access Manager

Access Manager installation involves Running the Java ES Installer to install the first instance or Access Manager and Running the amconfig Script to deploy additional instances or to configure the first Access Manager instance if you specified the "Configure Later" option during installation.

Running the Java ES Installer

To install the first instance of Access Manager 6 2005Q1, run the Sun Java Enterprise System (Java ES) 2005Q1 installer. When you run the installer, you can also install other Java ES component products such as Application Server or Web Server.

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

During installation, if you the specify the "Configure Now" option, the "Access Manager: Directory Server Information (6 of 6)" panel asks the question: "Is Directory Server provisioned with user data?". Choose "Yes" and then specify values for the object classes and naming attributes shown in Table 3-1 with the values that your are currently using in your Directory Server.

Table 3-1 Object Classes and Naming Attributes For a Provisioned Directory

Item	Description
Organization Marker Object Class	Object class defined for the organization in the existing provisioned directory. Default value is sunISManagedOrganization.
Organization Naming Attribute	Naming attribute used to define organizations in the existing provisioned directory. Default value is o.
User Marker Object Class	Object class defined for users in the existing provisioned directory. Default value is inetorgperson.
User Naming Attribute	Naming attribute used for users in the existing provisioned directory. Default value is uid.

If you the specify the “Configure Now” option, the Java ES installer also displays a popup dialog box that asks:

```
The Directory Server does not have the Access Manager 6.3 directory
information tree (DIT). Do you want the installer to load the DIT into your
Directory Server?
```

If you click Yes, the installer loads the Access Manager LDIF files, including `install.ldif`, `installExisting.ldif`, `ds_remote_schema.ldif`, `ds_remote_schema_uninstall.ldif`, `sunAMClient_data.ldif`, `sunAMClient_schema.ldif`, and `sunone_schema2.ldif`.

If you specify No, you must manually load these files after installation. See “Configuring the Directory Server Schema Files” on page 47 for more information.

Running the amconfig Script

Run the `amconfig` script to deploy additional instances of Access Manager on other host servers or to configure the first instance if you specified the “Configure Later” option during installation.

First, copy the `amsamplesilent` and then set the variables in the new file to configure the new instance you want to deploy or configure. The `amconfig` script reads the configuration script input file and then calls other scripts as needed to create or configure the Access Manager instance.

If you are using an existing Directory Server, set the DIRECTORY_MODE in the configuration script input file as shown in Table 3-2.

Table 3-2 DIRECTORY_MODE Values for an Existing Directory Server

Value	Description
DIRECTORY_MODE=2	Use for an existing Directory Server DIT. The naming attributes and object classes are the same, so the configuration scripts load the installExisting.ldif and umsExisting.xml files. The configuration scripts also update the LDIF and properties files with the actual values entered during configuration (for example, BASE_DIR, SERVER_HOST, and ROOT_SUFFIX). This update is also referred to as “tag swapping,” because the configuration scripts replace the placeholder tags in the files with the actual configuration values.

Table 3-2 DIRECTORY_MODE Values for an Existing Directory Server *(Continued)*

Value	Description
DIRECTORY_MODE=3	Use for an existing Directory Server DIT when you want to do a manual load. The object classes and naming attributes are different, so the configuration scripts do not load the installExisting.ldif and umsExisting.xml files. The scripts perform tag swapping (described for mode 2). You should inspect and modify (if needed) the LDIF files and then manually load the LDIF files and services. For more information, see "Configuring the Directory Server Schema Files" on page 47.
DIRECTORY_MODE=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 add a server entry in the platform list.

Table 3-3 describes other variables that you need to set in the configuration script input file for an existing Directory Server.

Table 3-3 Configuration Script Input File Variables for an Existing Directory Server

Variable	Description
USER_NAMING_ATTR	Naming attribute used for users in the existing provisioned directory. Default value is uid.
ORG_NAMING_ATTR	Naming attribute used to define organizations in the existing provisioned directory. Default value is o.
ORG_OBJECT_CLASS	Object class defined for the organization in the existing provisioned directory. Default value is sunISManagedOrganization.
USER_OBJECT_CLASS	Object class defined for users in the existing provisioned directory. Default value is inetorgperson.

Set other variables in the configuration script input file as required for the instance you want to configure or create. For a description of the `amconfig` script and the variables in the `amsamplesilent` file, refer to the *Sun Java System Access Manager Administration Guide* (`http://docs.sun.com/doc/817-7647`).

Important considerations for other variables are:

- NEW_INSTANCE specifies whether the configuration script should deploy Access Manager to a new user-created web container instance. Set to true to update the platform server list and DNS alias.

- AM_ENC_PWD specifies the password encryption key.

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

Configuring the Directory Server Schema Files

In this scenario, you ran the `amconfig` script with DIRECTORY_MODE = 3 or 4, or you specified "No" when the Java ES installer asked this question:

```
The Directory Server does not have the Access Manager 6.3 directory
information tree (DIT). Do you want the installer to load the DIT into your
Directory Server?
```

Follow these steps, in order, to configure the Directory Server schema on Solaris systems:

1. Load the LDIF files for the Access Manager specific schema changes. For example:

   ```
   # ldapmodify -h ds-host -p port -D"cn=directory manager" -w passwd -c -a
   -f /etc/opt/SUNWam/config/ldif/sunone_schema2.ldif
   # ldapmodify -h ds-host -p port -D"cn=directory manager" -w passwd -c -a
   -f /etc/opt/SUNWam/config/ldif/ds_remote_schema.ldif
   ```

2. Load the LDIF files for the client data and its schema. For example:

   ```
   # ldapmodify -h ds-host -p port -D"cn=directory manager" -w passwd -c -a
   -f /etc/opt/SUNWam/config/ldif/sunAMClient_schema.ldif
   # ldapmodify -h ds-host -p port -D"cn=directory manager" -w passwd -c -a
   -f /etc/opt/SUNWam/config/ldif/sunAMClient_data.ldif
   ```

3. Load the LDIF file for the Access Manager specific entries. For example:

   ```
   # ldapmodify -h ds-host -p port -D"cn=directory manager" -w passwd -c -a
   -f /etc/opt/SUNWam/config/ldif/installExisting.ldif
   ```

4. Add the Directory Server indexes and enable the plug-in:

   ```
   # ldapmodify -h ds-host -p port -D"cn=directory manager" -w passwd -c -a
   -f /etc/opt/SUNWam/config/ldif/index.ldif
   # ldapmodify -h ds-host -p port -D"cn=directory manager" -w passwd -c -a
   -f /etc/opt/SUNWam/config/ldif/plugin.ldif
   ```

   For more information, see "Configuring Directory Server" on page 49.

5. Optionally, add any schema customizations that you might want for your deployment.

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

   a. Change to the directory where the script is located. For example, on Solaris systems:
   ```
   cd /etc/opt/SUNWam/config/ums
   ```

   b. Open the `umsExisting.xml` file and verify that it has the correct naming attribute.

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

   d. Run the `amserveradmin` script. For example:
   ```
   # cd /etc/opt/SUNWam/config/ums/
   # ./amserveradmin "cn=amadmin,ou=people,dc=example,dc=com"
   "passwd_amadmin"
   ```

7. If there are no errors in the Step a through Step d, restart the container and authentication helpers. You should be now able to login to the admin console.

Customizing After Installation

In this scenario, you specified Yes when the Java ES installer asked this question:

```
The Directory Server does not have the Access Manager 6.3 directory
information tree (DIT). Do you want the installer to load the DIT into your
Directory Server?
```

Then after installation, if you want to customize the Directory Server schema, follow these general steps:

1. Remove the DAI service.

2. Add your customizations to the XML files.

3. Re-add the DAI service.

For more detailed information, see "Directory Access Instructions (DAI) Service" on page 74.

Configuring Directory Server

After you have installed Access Manager, perform the following tasks (if they have not already been done) to configure Directory Server to work with Access Manager:

- Enable the Referential Integrity Plug-In
- Add Indexes for Access Manager

To perform these tasks, Directory Server must be running.

Enable the Referential Integrity Plug-In

When enabled, the referential integrity plug-in maintains integrity on specific attributes immediately after a Directory Server delete or modify operation. This plug-in ensures that relationships between related entries are maintained throughout the Directory Server database.

To enable the referential integrity plug-in, use the Directory Server Console or the `ldapmodify` command-line utility to load the `plugin.ldif` script, which is available in the following directory, depending on your platform:

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

To enable the referential integrity plug-in using the Console

1. Make sure that Directory Server is running.
2. In Directory Server Console, click the Configuration tab.
3. In the left tree, expand the Plug-ins node.
4. In list of Plug-ins, click the “referential integrity postoperation” plug-in. Add the following attributes (if missing) to the arguments list of plugin (on right hand panel):
   - iplanet-am-modifiable-by
   - iplanet-am-static-group-dn
   - memberOf
5. In the properties area, check Enable plug-in.

6. Set the following arguments:

   - Argument 1 specifies the update interval in seconds. For example, 90 updates occur every 90 seconds, 3600 updates occur every hour, and so on.

     The default for Argument 1 is 0, which updates immediately after every operation. However, be aware that immediate referential integrity checks after every operation can significantly impact server performance.

     To determine a value that considers both data integrity and overall performance, refer to the *Directory Server Performance Tuning Guide* in the "Directory Server Online Documentation" on page 43.

   - Argument 2 specifies the absolute path of the referential integrity log file you plan to use.

   - Argument 3 is not used, but it must be present.

7. Click Save.

8. Restart Directory Server.

Add Indexes for Access Manager

Directory Server indexes improve the performance for searches of Directory Server data. Table 3-4 lists the recommended attributes that you should consider indexing for Access Manager (if they are not already indexed).

Table 3-4 Directory Server Attributes to Index for Access Manager

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

You can add indexes using either the Directory Server Console or the `ldapmodify` command-line utility to load the `index.ldif` file, which is available in the following directory, depending on your platform:

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

For more information about both the Console and `ldapmodify`, see the *Sun Java System Directory Server Administration Guide* in the "Directory Server Online Documentation" on page 43.

To add Directory Server indexes using the Console

1. Make sure that Directory Server is running.
2. In Directory Server Console, click the Configuration tab.
3. In the left tree, expand the Data node and select the suffix you want to index.
4. Click the Indexes tab in the right panel.
5. To add an index on an attribute that is not yet indexed, click Add Attribute. In the subsequent dialog, select one or more attributes to index, and click OK.
6. Check the index type for the attribute, as shown in Table 3-4.
7. Repeat these steps for other indexes you want to add.
8. Click Save.
9. When you are finished creating new indexes, click Close in the Indexes window.
10. For the new indexes to take effect, restart Directory Server.

Adding Access Manager Object Classes

Before Access Manager can recognize the data in an existing directory, you must add special object classes and attributes to entries for all organizations, groups, and users that will be managed by Access Manager. Access Manager includes the Directory Server Sample Data Migration Scripts to add these object classes to your directory.

You might consider the Access Manager object classes as markers that indicate the directory entries that you want to manage through Access Manager. These markers enable Access Manager to recognize the entries in your directory.

Approaches to Modifying Your Directory Tree

To modify your existing directory tree, consider using one of these approaches:

- Make a few modifications in your LDIF and XML files and then start Access Manager to make sure that the modifications were done correctly before you continue with other modifications.

  For example, you may want to add the Access Manager object classes for each of your organizations, restart Access Manager, and verify that your organizations appear in the Access Manager Administration Console. Then, add marker object classes for groups, check them, and so forth.

- If you have experience modifying an LDAP directory server, consider making the required modifications to your directory tree before loading the Access Manager LDIF and XML configuration files. This approach is often prone to errors, but it is usually faster.

You can make these modifications by using Directory Server Console, or by using the `ldapmodify` or `db2ldif` utilities that are included with Directory Server. You can also use the Directory Server Sample Data Migration Scripts that are included with Access Manager.

Directory Server Sample Data Migration Scripts

Table 3-5 describes the Directory Server data migration scripts, which are available the following directory, depending on your platform:

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

To run these sample scripts, Perl 5.x or later is required. Each script generates an LDIF file that you can inspect before making actual changes to your Directory Server directory tree.

Before you run a script, set your LD_LIBRARY_PATH appropriately, depending on your platform. For more information, see Directory Server Libraries.

Detailed steps for running each sample script are included under the description of each script. The general steps to run a script are:

1. In the script you plan to run, set the `following` variables:
   - `$base` to the base suffix of the directory tree to be managed by Access Manager. For example: `"dc=MadisonParc,dc=com"`
   - `$LDAP_SEARCH` to the path of your ldapsearch command. For example: `"/opt/SUNWam/bin/ldapsearch"`.
   - `$LDAP_MODIFY` to the path of your ldapmodify command. For example: `"/opt/SUNWam/bin/ldapmodify"`.

   See "Directory Server Utilities" on page 42 for more information.
2. Run the script "as is" and then inspect the generated LDIF file.
3. If the changes in the LDIF file are satisfactory, uncomment the `ldapmodify` command in the last line of the script.
4. Run the script a second time to make the actual changes.

CAUTION These scripts make changes to your Directory Server data that cannot be automatically undone. Be sure to back up your data before running a script.

Table 3-5 Scripts for Adding Access Manager Marker Object Classes

Script	What it Does
`update-o.pl`	Adds the following to each organization entry: • sunManagedOrganization • sunISManagedOrganization • sunNameSpace • inetDomain • inetDomainStatus See “Marking Organizations” on page 55.
`update-people.pl`	Adds iplanet-am-managed-people-container to each people container. See “Marking People Containers” on page 57.
`update-ou.pl`	Adds iplanet-am-managed-org-unit to each organizational unit. See “Marking Organizational Units” on page 59.
`update-users.pl`	Adds the following to each user entry: • inetadmin • iplanet-am-managed-person • iplanet-am-user-service • inetuser • iPlanetPreferences • inetOrgPerson See “Marking Users” on page 61.
`udpate-static-groups.pl`	Adds the following to each static group: • iplanet-am-managed-static-group • iplanet-am-managed-group See “Marking Static Groups” on page 63.
`update-filtered-groups`	Adds the following to each dynamic, or filtered, group: • iplanet-am-managed-group • iplanet-am-managed-filtered-group See “Marking Dynamic (Filtered) Groups” on page 65.
`update-assignable-dynamic-groups`	Adds the following to each assignable dynamic group: • iplanet-am-managed-group • iplanet-am-managed-assignable group See “Marking Assignable Dynamic Groups” on page 67.

Table 3-5 Scripts for Adding Access Manager Marker Object Classes *(Continued)*

Script	What it Does
`update-groups.pl`	Adds iplanet-am-managed-group-container to each organizational unit that contains groups. See "Marking Group Containers" on page 68.

Marking Organizations

Script: update-o.pl

If you used an existing organization as your default organization during installation, you do not have to make these changes. The installation program automatically adds these object classes and attributes. Continue with "Marking People Containers" on page 57.

If you have sub-organizations or custom organizations, make the following changes:

- Add the following object classes to each organization entry:
  - `sunManagedOrganization`
  - `sunNameSpace`
  - `inetDomain`
- Add the `inetDomainStatus` attribute to each organization entry.

In the MadisonParc example, these object classes and their attributes are added to the `dc=Customers` and `dc=Suppliers` organizations.

To run the update-o.pl script

1. Copy `update-o.pl` to the following directory:

   DirectoryServer-base`/shared/bin`

2. In the script, set the `following` variables:
   - `$base` to the base suffix of the directory tree to be managed by Access Manager. For example: "`dc=MadisonParc,dc=com`"
   - $LDAP_SEARCH to the path of your ldapsearch command. For example: "/opt/SUNWam/bin/ldapsearch"

- $LDAP_MODIFY to the path of your ldapmodify command. For example: "/usr/iplanet/servers/shared/bin/ldapmodify"

3. In the directory where the script is located, enter the following command:

   ```
   perl update-o.pl
   ```

4. When prompted, provide the following information:

   Enter Host Name: Enter the name of the computer system in which your Directory Server is installed.

   Enter Bind User Name: Enter a user name that has sufficient privileges for accessing the entire directory. Example: `cn=Directory Manager`

   Enter Bind password: Enter the password for the user you specified above.

   Enter port number: Enter the Directory Server port number. Example: `389`

5. Check the results in the file `orgs-updated.ldif` that is generated by the script, and verify that the appropriate changes are listed. The changes contained in this file will automatically be made in the directory in the next step.

   Important Note: If there are organizations that you do not want to be managed by Access Manager, you should delete those entries from this `orgs-updated.ldif` now. Then, instead of going on to the next step, manually load the file using the following command:

   ```
   ldapmodify -h hostname -p port -D bind_user -w password -a -c -f
       orgs-updated.ldif
   ```

6. In the script `update-o.pl`, uncomment the last line and replace variables appropriately. For example, to add marker object classes to MadisonParc directory entries, the last line of the script is changed from this:

   ```
   #system("$LDAP_MODIFY -h'$host' -p'$port' -D'$bind_user'
       -w'$bind_pwd' -a -c -f orgs-updated.ldif");
   ```

   to this:

   ```
   system("$LDAP_MODIFY -h'ginac.MadisonParc.com' -p'389'
       -D'cn=Directory Manager' -w'password' -a -c -f
        orgs-updated.ldif");
   ```

   where *password* is the Directory Manager password.

In Code Example 2-1, the modifications to the MadisonParc directory entries are indicated in bold:

Code Example 3-1 Organization Entries With Marker Object Classes

```
...
dn: dc=Customers,dc=MadisonParc,dc=com
dc: Customers
objectClass: top
objectClass: domain
objectClass: external
objectClass: sunManagedOrganization
objectClass: sunNameSpace
objectClass: inetDomain
inetDomainStatus: Active


dn: dc=Suppliers,dc=MadisonParc,dc=com
dc: Suppliers
objectClass: top
objectClass: domain
objectClass: external
objectClass: sunManagedOrganization
objectClass: sunNameSpace
objectClass: inetDomain
inetDomainStatus: Active
...
```

Marking People Containers

Script: update-people.pl

People containers are typically assigned the `ou` attribute and are used to store all user entries for a branch of the directory. To each people container, add the `iplanet-am-managed-people-container` object class.

To run the update-people.pl script

1. Copy `update-people.pl` to the following directory:

   DirectoryServer-base`/shared/bin`

2. Make sure that the $base variable is set to the base suffix of the directory tree to be managed by Access Manager. For example: dc=MadisonParc,dc=com

   Also, make sure that the LDAP_SEARCH and LDAP_MODIFY variables are set correctly. For more information see "Directory Server Utilities" on page 42.

   In the MadisonParc example, the script was also modified to include people containers located under the organizations. In Code Example 2-14, bold indicates the change in the search scope.

   Code Example 3-2 Scope in `update-people-container.pl` is Modified

   ```
   # run search to find all people containers, putting their DNs in to a
   file
   system("$LDAP_SEARCH -h \"$host\" -p \"$port\" -D \"$bind_user\"
       -w \"$bind_pwd\" -b \"$base\" -s sub -T \"(&(ou=$people)
          (!(objectclass=iplanet-am-*)))\" dn > people.dn");
   ```

3. In the directory where the script is located, at the command line enter the following:

   `perl update-people.pl`

4. When prompted, provide the following information:

   Enter Host Name: Enter the name of the computer system in which your Directory Server is installed.

   Enter Bind User Name: Enter a user name that has sufficient privileges for accessing the entire directory. Example: cn=Directory Manager

   Enter Bind password: Enter the password for the user you specified above.

   Enter port number: Enter the Directory Server port number. For example: 389

   Enter People Container: Enter the name of the people container that contains the uids you want to modify. For example: People

5. Check the results in the file `people-updated.ldif` which is created in the same directory as the script, and verify that the appropriate changes were made. The changes contained in this file will automatically be made in the directory in the next step.

   Important Note: If there are people containers that you do not want to be managed by Access Manager, you should delete those entries from `people-updated.ldif` now. Then, instead of going on the to next step, manually load the file using the following command:

   ```
   ldapmodify -h hostname -p port -D bind_user -w password -a -c -f
     people-updated.ldif
   ```

6. In the script `update-people.pl`, uncomment the last line and replace variables appropriately. In the MadisonParc example, the last line of the script is changed from this:

   ```
   #system("$LDAP_MODIFY -h'$host' -p'$port' -D'$bind_user'
     -w'$bind_pwd' -a -c -f people-updated.ldif");
   ```

   to this:

   ```
   system("$LDAP_MODIFY -h'ginac.MadisonParc.com' -p'389'
     -D'cn=Directory Manager' -w'password' -a -c -f
     people-updated.ldif");
   ```

   In Code Example 2-15, marker object class for the people container under dc=Customers is indicated in bold.

Code Example 3-3 People container entry with marker object class.

```
...
dn: ou=People,dc=Customers,dc=MadisonParc,dc=com
ou: People
objectClass: top
objectClass: organizationalunit
objectClass: iplanet-am-managed-people-container
```

Marking Organizational Units

Script: update-ou.pl

Organizational units are typically assigned the ou attribute. To each container that is an organizational unit, add the following object class:

```
iplanet-am-managed-org-unit
```

To run the update-ou.pl script

1. Copy `update-ou.pl` to the following directory:

DirectoryServer-base/shared/bin

2. Make sure that the $base variable is set to the base suffix of the directory tree to be managed by Access Manager. For example: dc=MadisonParc,dc=com

   Also, make sure that the LDAP_SEARCH and LDAP_MODIFY variables are set correctly. For more information see "Directory Server Utilities" on page 42.

3. In the directory where the script is located, at the command line enter the following:

   ```
   perl update-ou.pl
   ```

4. When prompted, provide the following information:

   Enter Host Name: Enter the name of the computer system in which your Directory Server is installed.

   Enter Bind User Name: Enter a user name that has sufficient privileges for accessing the entire directory. For example: cn=Directory Manager

   Enter Bind password: Enter the password for the user you specified above.

   Enter port number: Enter the Directory Server port number. For example: 389

5. Check the results in the file orgunit-updated.ldif which is created in the same directory as the script, and verify that the appropriate changes are listed. The changes contained in this file will automatically be made in the directory in the next step.

   Important Note: If there are organizational units that you do not want to be managed by Access Manager, you should delete those entries from this ou-updated.ldif now. Then, instead of going on the to next step, manually load the file using the following command:

   ```
   ldapmodify -h hostname -p port -D bind_user -w password -a -c -f
     orgunit-updated.ldif
   ```

6. In the script update-ou.pl, uncomment the last line and replace variables appropriately. In the MadisonParc example, the last line of the script is changed from this:

   ```
   #system("$LDAP_MODIFY -h'$host' -p'$port' -D'$bind_user'
     -w'$bind_pwd' -a -c -f orgunit-updated.ldif");
   ```

   to this:

   ```
   system("$LDAP_MODIFY -h'ginac.MadisonParc.com' -p'389'
     -D'cn=Directory Manager' -w'password' -a -c -f
     orgunit-updated.ldif");
   ```

In Code Example 2-16, marker object class for the organizational units under `dc=MadisonParc,dc=com` is indicated in bold.

Code Example 3-4 Organizational Unit Entry With Marker Object Class

```
...
dn: ou=People,dc=Customers,dc=MadisonParc,dc=com
ou: People
objectClass: top
objectClass: organizationalunit
objectClass: iplanet-am-managed-people-container
...
```

Marking Users

Script: update-users.pl

To each user entry, add the following object classes:

- iplanet-am-managed-person
- iplanet-am-user-service
- inetuser
- iPlanetPreferences
- inetOrgPerson
- inetadmin

To run the update-users.pl script

1. Copy `update-users.pl` to the following directory:

   DirectoryServer-base`/shared/bin`

2. Make sure that the `$base` variable is set to the base suffix of the directory tree to be managed by Access Manager. For example: `dc=MadisonParc,dc=com`

   Also, make sure that the LDAP_SEARCH and LDAP_MODIFY variables are set correctly. For more information see “Directory Server Utilities” on page 42.

3. In the directory where the script is located, at the command line enter the following:

   ```
   perl udpate-users.pl
   ```

4. Check the results in the file `users-updated.ldif` which is created in the same directory as the script, and verify that the appropriate changes were made. The changes contained in this file will automatically be made in the directory in the next step.

   Important Note: If there are users that you do not want to be managed by Access Manager, you should delete those entries from `users-updated.ldif` now. Then, instead of going on the to next step, manually load the file using the following command:

   ```
   ldapmodify -h hostname -p port -D bind_user -w password -a -c -f
     users-updated.ldif
   ```

5. In the script `update-users.pl`, uncomment the last line and replace variables appropriately. In the MadisonParc example, the last line of the script is changed from this:

   ```
   #system("$LDAP_MODIFY -h'$host' -p'$port' -D'$bind_user'
     -w'$bind_pwd' -a -c -f users-updated.ldif");
   ```

   to this:

   ```
   system("$LDAP_MODIFY -h'ginac.MadisonParc.com' -p'389'
     -D'cn=Directory Manager' -w'password' -a -c -f
     users-updated.ldif");
   ```

In Code Example 2-17, the user marker object class is indicated in bold.

Code Example 3-5 User Entry With User Marker Object Class

```
dn: uid=scarter, ou=People, dc=MadisonParc,dc=com
nsUniqueId: d8855082-1dd111b2-8024a6c9-802bec30
givenName: Sam
telephoneNumber: +1 408 555 4798
sn: Carter
ou: Accounting
ou: People
l: Sunnyvale
roomNumber: 4612
mail: scarter@MadisonParc.com
facsimileTelephoneNumber: +1 408 555 9751
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetuser
objectClass: inetadmin
objectClass: iplanet-am-managed-person
objectClass: iplanetPreferences
objectClass: iplanet-am-user-service
uid: scarter
```

Code Example 3-5 User Entry With User Marker Object Class *(Continued)*

```
cn: Sam Carter
userPassword: {SSHA}3XwjhBgbt6ae5syCndDeANoossEGRJ1NdnLyZw==
employeeType: Manager
departmentNumber: 1000
businessCategory: East
inetUserStatus: Active
```

Marking Static Groups

Script: update-static-groups.pl

Static groups formed by adding uids to the group entry. To each group entry containing values for the `uniquemember` attribute, add the following object classes:

- `iplanet-am-managed-static-group`
- `iplanet-am-managed-group`

To run the update-static-groups.pl script

1. Copy `update-static-groups.pl` to the following directory:

   DirectoryServer-base`/shared/bin`

2. Make sure that the `$base` variable is set to the base suffix of the directory tree to be managed by Access Manager. For example: `dc=MadisonParc,dc=com`

   Also, make sure that the LDAP_SEARCH and LDAP_MODIFY variables are set correctly. For more information see “Directory Server Utilities” on page 42.

3. In the directory where the script is located, at the command line enter the following:

   `perl update-static-groups.pl`

   When prompted, provide the following information:

   Enter Host Name: Enter the name of the computer system in which your Directory Server is installed.

   Enter Bind User Name: Enter a user name that has sufficient privileges for accessing the entire directory. For example: `cn=Directory Manager`

   Enter Bind password: Enter the password for the user you specified above.

   Enter port number: Enter the Directory Server port number. For example: 389

4. Check the results in the file `static-groups-updated.ldif` which is created in the same directory as the script, and verify that the appropriate changes are listed. The changes contained in this file will automatically be made in the directory in the next step.

   Important Note: If there are static groups that you do not want to be managed by Access Manager, you should delete those entries from `static-groups-updated.ldif` now. Then, instead of going on the to next step, manually load the file using the following command:

   ```
   ldapmodify -h hostname -p port -D bind_user -w password -a -c -f
     static-groups-updated.ldif
   ```

5. In the script `update-static-groups.pl`, uncomment the last line, and replace variables appropriately. For example, in the MadisonParc example, the last line of the script is changed from this:

   ```
   #system("$LDAP_MODIFY -h'$host' -p'$port' -D'$bind_user'
     -w'$bind_pwd' -a -c -f static-groups-updated.ldif");
   ```

   to this:

   ```
   system("$LDAP_MODIFY -h'ginac.MadisonParc.com' -p'389'
     -D'cn=Directory Manager' -w'password' -a -c -f
     static-groups-updated.ldif");
   ```

In Code Example 2-18, marker object class for static groups is indicated in bold.

Code Example 3-6 Static Group Entry With Marker Object Classes

```
dn: cn=Directory Administrators, dc=MadisonParc,dc=com
nsUniqueId: 60a72e02-1dd211b2-8003a6c9-802bec30
objectClass: top
objectClass: groupofuniquenames
objectClass: iplanet-am-managed-group
objectClass: iplanet-am-managed-static-group
cn: Directory Administrators
uniqueMember: uid=kvaughan, ou=People, dc=MadisonParc,dc=com
uniqueMember: uid=alutz, ou=People, dc=MadisonParc,dc=com
uniqueMember: uid=gjensen, ou=People, dc=MadisonParc,dc=com
uniqueMember: uid=tcouzens, ou=People, dc=MadisonParc,dc=com
```

Marking Dynamic (Filtered) Groups

Script: update-filtered-groups.pl

Dynamic or filtered groups are formed by building a search construct to find all user entries containing a specific attribute. These groups contain the `memberURL` attribute. To each group containing the attribute `memberURL`, add the following object classes:

- `iplanet-am-managed-group`
- `iplanet-am-managed-filtered-group`

To run the update-filtered-groups.pl script

1. Copy `update-filtered-groups.pl` to the following directory:

   `DirectoryServer-base/shared/bin`

2. Make sure that the `$base` variable is set to the base suffix of the directory tree to be managed by Access Manager. For example: `dc=MadisonParc,dc=com`

   Also, make sure that the LDAP_SEARCH and LDAP_MODIFY variables are set correctly. For more information see "Directory Server Utilities" on page 42.

3. In the directory where the script is located, at the command line enter the following:

   ```
   perl update-filtered-groups.pl
   ```

4. When prompted, provide the following information:

   Enter Host Name: Enter the name of the computer system in which your Directory Server is installed.

   Enter Bind User Name: Enter a user name that has sufficient privileges for accessing the entire directory. For example: `cn=Directory Manager`

   Enter Bind password: Enter the password for the user you specified above.

   Enter port number: Enter the Directory Server port number. For example: `389`

5. Check the results in the file `filtered-groups-updated.ldif` which is created in the same directory as the script, and verify that the appropriate changes were made. The changes contained in this file will automatically be made in the directory in the next step.

   Important Note: If there are filtered or dynamic groups that you do not want to be managed by Access Manager, you should delete those entries from `filtered-groups-updated.ldif` now. Then, instead of going on the to next step, manually load the file using the following command:

   ```
   ldapmodify -h hostname -p port -D bind_user -w password -a -c -f
     filtered-groups-updated.ldif
   ```

6. In the script `update-filtered-groups.pl`, uncomment the last line in the `update-o.pl` file, and replace variables appropriately. In the MadisonParc example, the last line of the script is changed from this:

   ```
   #system("$LDAP_MODIFY -h'$host' -p'$port' -D'$bind_user'
     -w'$bind_pwd' -a -c -f filtered-groups-updated.ldif");
   ```

   to this:

   ```
   system("$LDAP_MODIFY -h'ginac.MadisonParc.com' -p'389'
     -D'cn=Directory Manager' -w'password' -a -c -f
     filtered-groups-updated.ldif");
   ```

In Code Example 2-19, marker object class for a filtered group is indicated in bold.

Code Example 3-7 Dynamic or Filtered Group With Marker Object Classes

```
dn: cn=North,ou=groups,dc=MadisonParc,dc=com
nsUniqueId: 60a72e35-1dd211b2-8003a6c9-802bec30
objectClass: top
objectClass: groupOfUniqueNames
objectClass: groupofurls
objectClass: iplanet-am-managed-group
objectClass: iplanet-am-managed-filtered-group
ou: groups
cn: North
memberURL:
ldap:///dc=MadisonParc,dc=com??sub?(&(|(objectclass=person)(objectc
 lass=groupofuniquenames))(businessCategory=*North*))
```

Marking Assignable Dynamic Groups

Script: update-assignable-dynamic-groups.pl

The *assignable* dynamic group is an Access Manager concept. In Access Manager, users in this type of group are typically allowed limited self-registration and account management privileges. In the MadisonParc example, users at the top level have administrators to create and manage their entries to comply with corporate specifications. Users under the Customers or Suppliers organizations are placed in assignable dynamic groups. The users can acquire membership by themselves when they log into the MadisonParc portal. Their membership entitles them to limited access to the MadisonParc portal; the information they provide at registration is minimal.

Add the following object classes to each dynamic group that you want to use as an assignable dynamic group in Access Manager:

- `iplanet-am-managed-group`
- `iplanet-am-managed-assignable-group`

To run the update-assignable-dynamic-groups.pl script

1. Copy `update-assignable-dynamic-groups.pl` to the following directory:

   DirectoryServer-base`/shared/bin`

2. Make sure that the `$base` variable is set to the base suffix of the directory tree to be managed by Access Manager. For example: `dc=MadisonParc,dc=com`

   Also, make sure that the LDAP_SEARCH and LDAP_MODIFY variables are set correctly. For more information see "Directory Server Utilities" on page 42.

3. In the directory where the script is located, at the command line enter the following:

   ```
   perl update-assignable-dynamic-groups.pl
   ```

4. When prompted, provide the following information:

   Enter Host Name: Enter the name of the computer system on which your Directory Server is installed.

   Enter Bind User Name: Enter a user name that has sufficient privileges for accessing the entire directory. For example: `cn=Directory Manager`

   Enter Bind password: Enter the password for the user you specified above.

   Enter port number: Enter the Directory Server port number. For example: `389`

5. Check the results in the file `assignable-dynamic-groups-updated.ldif` which is created in the same directory as the script, and verify that the appropriate changes were made. The changes contained in this file will automatically be made in the directory in the next step.

   Important Note: If there are assignable dynamic groups that you do not want to be managed by Access Manager, you should delete those entries from this `assignable-dynamic-groups-updated.ldif` now. Then, instead of going on the to next step, manually load the file using the following command:

   ```
   ldapmodify -h hostname -p port -D bind_user -w password -a -c -f
     assignable-dynamic-groups-updated.ldif
   ```

6. In the script `update-assignable-dynamic-groups.pl`, uncomment the last line in the `update-assignable-dynamic-groups.pl` file, and replace variables appropriately. In the MadisonParc example, the last line of the script is changed from this:

   ```
   #system("$LDAP_MODIFY -h'$host' -p'$port' -D'$bind_user'
     -w'$bind_pwd' -a -c -f
     assignable-dynamic-groups-updated.ldif");
   ```

   to this:

   ```
   system("$LDAP_MODIFY -h'ginac.MadisonParc.com' -p'389'
     -D'cn=Directory Manager' -w'password' -a -c -f
     assignable-dynamic-groups-updated.ldif");
   ```

Marking Group Containers

Script: update-groups.pl

Group containers are organizational units (`ou`) that contain groups. To each group container that includes the `ou:Groups` attribute, add the following object class:

```
iplanet-am-managed-group-container
```

To run the update-groups.pl script

1. Copy `update-groups.pl` to the following directory:

   DirectoryServer-base`/shared/bin`

2. Make sure that the $base variable is set to the base suffix of the directory tree to be managed by Access Manager. For example: dc=MadisonParc,dc=com

   Also, make sure that the LDAP_SEARCH and LDAP_MODIFY variables are set correctly. For more information see "Directory Server Utilities" on page 42.

   In the MadisonParc example, the script was also modified to include all group containers located under organizations. In Code Example 2-20, the script changes are indicated in bold.

Code Example 3-8 Scope in update-groups.pl is Modified.

```
# run search to find all group containers, putting their DNs in to a file
system("$LDAP_SEARCH -h \"$host\" -p \"$port\" -D \"$bind_user\"
   -w \"$bind_pwd\" -b \"$base\" -T \"(&(ou=groups)
    (!(objectclass=iplanet-am-*))(objectclass=organizationalunit))\
      " dn > group-container-updated.dn");
```

3. In the directory where the script is located, at the command line enter the following command:

   `perl update-groups.pl`

4. When prompted, provide the following information:

   Enter Host Name: Enter the name of the computer system in which your Directory Server is installed.

   Enter Bind User Name: Enter a user name that has sufficient privileges for accessing the entire directory. For example: cn=Directory Manager

   Enter Bind password: Enter the password for the user you specified above.

   Enter port number: Enter the Directory Server port number. For example: 389

5. Check the results in the file groups-updated.ldif which is created in the same directory as the script, and verify that the appropriate changes were made. The changes contained in this file will automatically be made in the directory in the next step.

   Important Note: If there are group containers that you do not want to be managed by Access Manager, you should delete those entries from this groups-updated.ldif now. Then, instead of going on the to next step, manually load the file using the following command:

   ldapmodify -h *hostname* -p *port* -D *bind_user* -w *password* -a -c -f
     groups-updated.ldif

6. In the script `update-groups.pl`, uncomment the last line, and replace variables appropriately. In the MadisonParc example, the last line of the script is changed from this:

   ```
   #system("$LDAP_MODIFY -h'$host' -p'$port' -D'$bind_user'
     -w'$bind_pwd' -a -c -f groups-updated.ldif");
   ```

   to this:

   ```
   system("$LDAP_MODIFY -h'ginac.MadisonParc.com' -p'389'
     -D'cn=Directory Manager' -w'password' -a -c -f
     groups-updated.ldif");
   ```

In Code Example 2-21, marker object class for a group under `dc=Customers` is indicated in bold.

Code Example 3-9 Group Container With Marker Object Class

```
...
dn: ou=Groups,dc=Customers,dc=MadisonParc,dc=com
nsUniqueId: 7880b101-1dd211b2-8007a6c9-802bec30
ou: Groups
objectClass: top
objectClass: organizationalunit
objectClass: iplanet-am-managed-group-container
...
```

Adding Custom Object Classes

If your existing directory tree contains custom object classes and attributes that are not included with Directory Server, you must add them to the Access Manager schema, as follows:

- Modifying the Creation Templates
- Adding Attributes to the Organization Schema
- Adding Attributes to the User Schema

MadisonParc Examples

The examples in this section use a fictitious company named MadisonParc to show modifications you can make to your Directory Server and Access Manager.

The MadisonParc directory information tree (DIT) includes three organizational units (ou) at the top level of the tree: Groups, People, and Special Users. These organizational units contain entries for the MadisonParc employees. Two organizations (dc), Customers and Suppliers, which were created under the root level, contain entries for non-employees.

The MadisonParc example has two custom object classes: madisonparc-org and company, and three custom attributes: madisonparc-org-description, acctNumber, and companyName.These object classes and attributes are not included in the Access Manager or Directory Server schema.

These object classes and attributes distinguish MadisonParc employees at the top level of the directory tree from non-employees in the Customers and Suppliers organizations.

Figure 3-1 shows the MadisonParc DIT.

Figure 3-1 MadisonParc Existing Directory Information Tree (DIT)

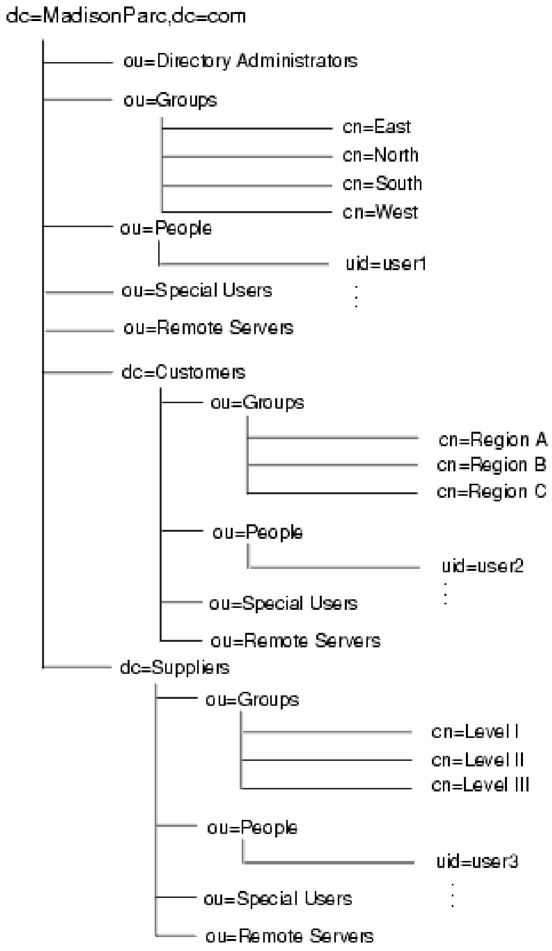

Table 3-6 describes the user-defined objects and attributes the MadisonParc directory information tree (DIT).

Table 3-6 User-Defined Objects Used in the MadisonParc DIT

Object	Description
madisonparc-org	Object class added to all organization entries.
madisonparc-org-description	Attribute added to each organization entry; required by madisonparc-org.
company	Object class added to all user entries.
acctNumber	Attribute added to each user entry; required by the company object class.
companyName	Attribute added to each user entry; required by the company object class.

Before a MadisonParc administrator can use Access Manager to manage these custom objects, the following modifications must be made to the Access Manager schema:

- Add the two custom object classes and three custom attributes to the umsExisting.xml file.
- Add madisonparc-org to the amEntrySpecific.xml file.
- Add madisonparc-org-description to the amEntrySpecific.properties file.
- Add companyName and acctNumber to the amUser.xml and amUser.properties files.

If your existing directory tree contains custom object classes or attributes, you can make similar changes in your directory and in your Access Manager XML files.

For more information about the Access Manager schema and customizing Access Manager, see the *Sun Java System Access Manager Developer's Guide*.

Modifying the Creation Templates

The creation templates configure Access Manager to add or allow specific object classes and attributes when these entries are created. To expose custom object classes in the Access Manager console, you must modify the creation templates for both users and organizations in the `umsExisting.xml` file.

In the MadisonParc example, the existing directory tree has new object classes for both users and organizations.

Directory Access Instructions (DAI) Service

When you install Access Manager, the `ums.xml file` is stored in Directory Server as the Directory Access Instructions (DAI) service. Access Manager does not allow you to load the `umsExisting.xml` file if the DAI service is already installed in Directory Server. Always remove the DAI service before modifying the `umsExisting.xml` file. Then after you have finished modifying the files, reload the DAI service into Directory Server.

To Modify the Creation Templates

1. If you loaded the Access Manager schema during installation or if you have already run the `amserveradmin` command for any reason, skip to Step 2.

   Otherwise, remove the DAI service. For example on Solaris systems:

   ```
   # cd opt/SUNWam/bin
   # ./amadmin -u "user_naming_attibute=amadmin,ou=people,root_suffix"
     -w password -r DAI
   ```

2. Locate the `umsExisting.xml` file. For example, on Solaris systems, the file is:

   ```
   /etc/opt/SUNWam/config/ums/umsExisting.xml
   ```

3. Modify any custom naming attributes. For example, the MadisonParc directory tree uses the `domain` attribute instead of the `organization` attribute.

   In the `umsExisting.xml` file, find the following SubConfiguration:

   ```
   <SubConfiguration name="BasicOrganization" id="CreationUmsObjects">
   ```

   Change `<Value>objectClass=organization</Value>` to `<Value>objectClass=domain</Value>`

   In Code Example 2-22, the bold text indicates the changed value. Note that three lines down, the naming attribute `dc` was changed by Access Manager during installation.

Code Example 3-10 Changing the Organization Naming Attribute in the Creation Template

```
<SubConfiguration name="BasicOrganization" id="CreationUmsObjects">
                        <AttributeValuePair> <Attribute name="name" />
                            <Value>BasicOrganization</Value>
                        </AttributeValuePair>
                        <AttributeValuePair> <Attribute name="javaclass" />
                            <Value>com.iplanet.ums.Organization</Value>
                        </AttributeValuePair>
                        <AttributeValuePair> <Attribute name="required" />
                            <Value>objectClass=top</Value>
```

Code Example 3-10 Changing the Organization Naming Attribute in the Creation Template *(Continued)*

```
                              <Value>objectClass=domain</Value>
                            <Value>objectClass=sunManagedOrganization</Value>
                            <Value>objectClass=sunNameSpace</Value>
                            <Value>dc</Value>
                            <Value>inetdomainstatus=Active</Value>
                        </AttributeValuePair>
                        <AttributeValuePair> <Attribute name="namingattribute"/>
                            <Value>dc</Value>
                        </AttributeValuePair>
                        <AttributeValuePair> <Attribute name="optional" />
                            <Value>*</Value>
                        </AttributeValuePair>
                    </SubConfiguration>
```

4. Add custom organization object classes.

   In the MadisonParc example, madisonparc-org is added to the organization creation template. Under the following SubConfiguration:

   `"BasicOrganiation" id="CreationUmsObjects">`

   under the following element:

   `<AttributeValuePair><Attribute name="required" />`

   add the following:

   ```
   <Value>objectClass=madisonparc-org</Value>
   <Value>madisonparc-org-description</Value>
   ```

   For example:

Code Example 3-11 Changing the Organization in the Creation Template

```
<SubConfiguration name="BasicOrganization" id="CreationUmsObjects">
                        <AttributeValuePair> <Attribute name="name" />
                            <Value>BasicOrganization</Value>
                        </AttributeValuePair>
                        <AttributeValuePair> <Attribute name="javaclass" />
                            <Value>com.iplanet.ums.Organization</Value>
                        </AttributeValuePair>
                        <AttributeValuePair> <Attribute name="required" />
                            <Value>objectClass=top</Value>
                            <Value>objectClass=domain</Value>
                            <Value>objectClass=sunManagedOrganization</Value>
                            <Value>objectClass=sunNameSpace</Value>
                            <Value>objectClass=madisonparc-org</Value>
                            <Value>dc</Value>
                            <Value>inetdomainstatus=Active</Value>
```

Code Example 3-11 Changing the Organization in the Creation Template

```
                        </AttributeValuePair>
                        <AttributeValuePair> <Attribute name="namingattribute"/>
                            <Value>dc</Value>
                        </AttributeValuePair>
                        <AttributeValuePair> <Attribute name="optional" />
                            <Value>*</Value>
                            <Value>madisonparc-org-description</Value>
                        </AttributeValuePair>
                    </SubConfiguration>
```

5. Add custom user object classes.

   In the MadisonParc example, company is added to the user creation template. Under the following SubConfiguration:

   `"BasicUser" id="CreationUmsObjects">`

   under the following element:

   `<AttributeValuePair><Attribute name="required" />`

   add the following:

   `<Value>objectClass=company</Value>`

   For example:

Code Example 3-12 Adding Custom User Object Classes in the Creation Template

```
<SubConfiguration name="CreationTemplates" >
                    <SubConfiguration name="BasicUser" id="CreationUmsObjects">
                        <AttributeValuePair> <Attribute name="name" />
                            <Value>BasicUser</Value>
                        </AttributeValuePair>
                        <AttributeValuePair> <Attribute name="javaclass" />
                            <Value>com.iplanet.ums.User</Value>
                        </AttributeValuePair>
                        <AttributeValuePair> <Attribute name="required" />
                            <Value>objectClass=top</Value>
                            <Value>objectClass=person</Value>
                            <Value>objectClass=organizationalPerson</Value>
                            <Value>objectClass=inetOrgPerson</Value>
                            <Value>objectClass=iPlanetPreferences</Value>
                            <Value>objectClass=iplanet-am-user-service</Value>
                            <Value>objectClass=inetuser</Value>
                            <Value>objectClass=inetAdmin</Value>
                            <Value>objectClass=iplanet-am-managed-person</Value>
                            <Value>objectClass=company</Value>
                            <Value>cn=default</Value>
```

Code Example 3-12 Adding Custom User Object Classes in the Creation Template

```
                <Value>sn=default</Value>
                <Value>uid</Value>
                <Value>inetuserstatus=Active</Value>
            </AttributeValuePair>
            <AttributeValuePair> <Attribute name="optional" />
                <Value>*</Value>
                <Value>companyname</Value>
                <Value>acctname</Value>
            </AttributeValuePair>
            <AttributeValuePair> <Attribute name="namingattribute"/>
                <Value>uid</Value>
            </AttributeValuePair>
        </SubConfiguration>
```

6. Reload the DAI service (`ums.xml` file or `umsExisting.xml` file). For example on Solaris systems:

```
# cd opt/SUNWam/bin
# ./amadmin -u "user_naming_attibute=amadmin,ou=people,root_suffix" -w password
  -s /etc/opt/SUNWam/config/ums/umsExisting.xml
```

Adding Attributes to the Organization Schema

To add attributes to the Organization schema, you must modify the following services files:

- `amEntrySpecific.xml`
- `amEntrySpecific.properties`

The Access Manager Console uses the information in `amEntrySpecific.xml` for display purposes. Each Access Manager abstract entry may have a subschema in this XML file. In the following example, you would add the object class `external` to the organization subschema. If the directory tree contained customized organizational units, groups, or people containers, you would add or modify their subschemas in the same XML file.

The subschema name for an organizational unit will be `OrganizationalUnit`. The subschema name for a people container will be `PeopleContainer`.

NOTE The User subschema is not configured here in the `amEntrySpecific.xml` file, but in the `amUser.xml` file (see "Adding Attributes to the User Schema" on page 81.) Although any service XML file may describe an attribute that is only for a user, the `amEntrySpecific.xml` file can serve as a default place holder for user attributes that are not tied to a particular service.

any attribute

The any attribute in the XML descriptions can have five possible values: `filter`, `display`, `adminDisplay`, `userReadOnly`, `required`, or `optional`. These values tell the Console whether the attribute should appear in the GUI. Typically, `required` and `optional` are not both displayed at the same time; they are mutually exclusive.

filter. The attribute is displayed in a search page.

display. The attribute is read / write for administrators and regular users.

adminDisplay. The attribute is read / write for administrators and is not displayed for regular users.

userReadOnly. The attribute is read / write for administrators but is read only for regular users. It is displayed as a label for regular users so that it is not editable. For example, the `display`, `adminDisplay`, and `userReadOnly` settings are used when displaying the user profile page and can be used to customize the page.

required. The attribute is displayed in the create page and requires a value during creation of the entry. If `any=required`, the attribute must have a value or the Console will not allow the Create operation. In the user interface, required fields are indicated with an asterisk (*). Use an empty string (" ") to tell the Administration Console to display nothing.

optional. The attribute is displayed in the create page but does not require a value during creation of the entry. If `any=optional`, the attribute will appear on the Create page without an asterisk. This would indicate that you don't have to give it a value to create the entry. In the Create User page, the User ID is a required attribute but the First Name is optional.

In the following MadisonParc example, the attribute `madisonparc-org-description` will be displayed on the Organization page and will be required for creation. This is indicated by the use of the `required` value. It will also be used on the Search page in Access Manager Console, as indicated by the use of the `filter` value.

```
<AttributeSchema name="madisonparc-org-description"
    type="single"
    syntax="string"
    any=required|filter
    i18nKey="o3"
/>
```

type attribute

The type attribute can use a string, string list, single choice, multiple choice, or boolean value. For example, the `madisonparc-org-description` attribute can have only one of two descriptions: internal or external). You would make this attribute a single choice; each description would be one of the choices. The Access Manager Console would display a list containing only these cities. If multiple cities were allowed, the attribute could be a multiple choice.

To add attributes from a custom organization to the organization subschema

1. In the following file add the custom object class to the subschema Organization. For example, on Solaris systems:

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

   In this example, the custom object class `madisonparc-org-description` was added to `amEntrySpecific.xml`.

```
<AttributeSchema name="madisonparc-org-description"
    type="single"
    syntax="string"
    any=required|filter
/>
```

2. In the same `amEntrySpecific.xml` file, create internationalization (i18n) keys (also called index keys or localization keys) for each attribute. All i18n Keys in an organization must be made up of unique strings. The Access Manager Administration Console will use this key to look up the display name for the attribute.

```
<AttributeSchema name="madisonparc-org-description"
    type="single"
    syntax="string"
    any="required|filter"
    i18nKey="o3"
/>
```

3. In the following file on Solaris systems:

   AccessManager-base`/SUNWam/locale/amEntrySpecific.properties`

   Add the value for i18n Key you created in Step 2. This is the name that will be displayed in the graphical user interface. For example:

```
iplanet-am-entry-specific-service-description=Access Manager Entry Specific
g1=Member List
g2=Users Can Subscribe to this Group
dg1=Membership Filter
r1=Membership Filter
o1=Full DNS name
o2=Organization Status
o3=Organization Description
```

   All the attributes listed in the subschema are displayed in the Administration Console when an organization is displayed. If an attribute is not listed, the Administration Console will not display the attribute.

TIP If an attribute has no i18n Key, it will not be displayed on the administration console. If you add an attribute, and you don’t see it in the administration console, be sure to check the i18n Key and properties.

4. Load all XML files.

   In the `/bin` directory, execute the following command:

   `./amserveradmin -u "`*user_naming_attibute*`=amadmin,ou=people,`*root_suffix*`"`
   `-w password`

If you see any parsing errors, you should go back and double-check the changes you made in the previous steps. Also examine the syntax in the `amEntrySpecific.xml` file, and make sure you’ve used the correct syntax. If you need to look at syntax examples, look at the other service XML files located in the following directory on Solaris systems:

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

Adding Attributes to the User Schema

In this step, you will modify two files for services:

- amUser.xml
- amUser.properties

The amUser.xml file is where user attributes are described, just as organization and group schema are described in the amEntrySpecific.xml (see Step 2). The file amUser.xml describes the User service for Access Manager. Note that any service may describe an attribute that is for a user only. This file is just the default placeholder for user attributes that are not tied to a particular service.

When displaying a user's attributes, the Access Manager Administration Console gets all attributes from all services that are subschema type User, and displays them using the same values as used in the amEntrySpecific.xml file (see the "any attribute" on page 78 and the "type attribute" on page 79). In the following examples, a few attributes from the madisonparc-user object class are added to the file, thus it is not necessary to create a new service. It's only necessary to modify, or extend, the iplanetamuserservice.

Additional Notes About the amUser.xml File

The file amUser.xml contains a special attribute. The any=display attribute tells Access Manager whether to display the attribute in the user profile page. This is a misleading name since it implies access control. It is strictly used for display. If this attribute is set to no then the console will not display the attribute.

Also note that the attributes are defined under subschema User and not Dynamic. Any attribute defined under User is physically an attribute in the user entry. If you want the attribute to be a role-based or organization-based attribute, then you would define it under the Dynamic subschema. For detailed information, see the *Java System Access Manager Developer's Guide*.

To add attributes from a custom organization to the user subschema

1. In the following file, add the attributes from the custom object class to the User subschema on Solaris systems:

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

   For example, the following two attributes from the custom object class `company` were added to the file:

```
<AttributeSchema name="companyname"
    type=single
    syntax=string
    any=required|display
    />
<AttributeSchema name="acctnumber"
    type=single
    syntax=string
    any=required|filter|display
```

2. In the same `amUser.xml` file, create i18n Keys (also called *index keys* or *localization keys*) for each attribute. All i18n Keys in an organization must be made up of unique strings. The Access Manager Console will use this key to look up the display name for the attribute.

```
<AttributeSchema name="companyname"
    type=single
    syntax=string
    any=required|display
    i18nKey=u120
/>
<AttributeSchema name="acctnumber"
    type=single
    syntax=string
    any=required|filter|display
    i18nKey=u121
```

3. Add values for the i18n Keys you created in Step 2 to the following file on Solaris systems:

   AccessManager-base/SUNWam/locale/amUser.properties

   For example:

```
iplanet-am-user-service-description=User
iwtUser-desc=Default User Profile
u101=UserId
u102=First Name
u103=Last Name
u104=Full Name
u105=Password
u106=Email Address
u107=Employee Number
u108=Telephone Number
u109=Manager
u110=Home Address
u111=User Status
u112=Account Expiration date (mm/dd/yyyy  hh:mm)
u113=User Authentication Configuration
u114=User Alias List
u115=Preferred Locale
u116=Success URL
u117=Failure URL
u118=Federation Information Key
u119=Federation Information
u120=Company Name
u121=Account Number
```

Examine the syntax in the amUser.xml file, and make sure you've used the correct syntax. If you need to look at syntax examples, look at the other service XML files located in the following directory on Solaris systems:

/etc/opt/SUNWam/config/xml

Loading Access Manager LDIF Files

Access Manager provides LDIF files to make modifications to your existing Directory Server in the following directory, depending on your platform:

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

The install.ldif and installExisting.ldif files contain Access Manager specific entries that are loaded into Directory Server by the Java Enterprise System installer and the amconfig script. Usually, you do not need to modify these files before they are loaded.

Other files include ds_remote_schema.ldif, ds_remote_schema_uninstall.ldif, sunAMClient_data.ldif, sunAMClient_schema.ldif, and sunone_schema2.ldif.

You must load the LDIF files using the ldapmodify utility, if you did not upgrade the Directory Server schema during installation.

Before you load the install.ldif and installExisting.ldif files, first change the place-holder administrator passwords to the actual passwords you are using.

For example, to load an LDIF file on Solaris systems:

```
# cd /etc/opt/SUNWam/config/ldif
# ldapmodify -v -c -D "cn=Directory manager" -w password -a -f name.ldif
```

where *password* is the Directory Manager password and *name*.ldif is the name of the LDIF file you want to load.

MadisonParc Example

In the MadisonParc example, when you load the LDIF files, the following occurs:

- Users and marker object classes required for Access Manager are added to dc=MadisonParc,dc=com and to dc=Customers and dc=Suppliers.
- Default roles for organization and help desk administrators are created at the top level.
- Default Access Control Instructions (ACIs) for those administrator entries are set up.

The Access Manager administration user amAdmin is created under the ou=People,dc=MadisonParc,dc=com people container. This is the top level administrator for Access Manager. This administrator has read and write access to the entire dc=MadisonParc,dc=com root suffix. You can add one of your users to this top level administrator role after the Access Manager Console is started.

Enabling Access Manager User Management

User Management includes all the components needed to view and manage the Access Manager objects (organizations, groups, users, services, roles policies, container objects, and agents). By default, User Management is enabled after you install Access Manager. However, if disabled, you can enable User Management using the Access Manager console.

To enable User Management

1. If necessary, restart Access Manager for any previous changes to take effect. Both Directory Server and the web container that runs Access Manager must also be running.
2. Log in to the Access Manager console as `amAdmin`.
3. In the Access Manager console, click Service Configuration and then Administration.
4. In the Administration pane under Global, check Enable User Management.
5. Click Save.

You should now be able to see your existing groups and users under Organization in the Administration pane.

Verifying the Access Manager Installation

After you've installed Access Manager and configured Directory Server, you can verify the installation.

To verify the Access Manager installation

1. If necessary, start Directory Server.
2. Start Access Manager and the web container that Access Manager is using (Web Server or Application Server). For more information, the refer to the *Sun Java System Access Manager Administration Guide* (`http://docs.sun.com/doc/817-7647`).
3. Login to the Access Manager console as `amAdmin` using the password you specified during installation, with the following URL:

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

where *host-name.domain-name:port* is the fully qualified host name and port of the web container you are using. For example:

```
http://amhost.example.com:58080/amserver/
```

You should see the root suffix and organizations you specified during installation.

For the MadisonParc example used in this chapter, you would see the root suffix MadisonParc, and the two organizations `Customers` and `Suppliers`.

Appendix A

Access Manager Upgrade Worksheets

Use the following worksheets to plan and perform an upgrade to Sun Java™ System Access Manager 6 2005Q1:

- "Pre-Upgrade Script Worksheet" on page 88
- "Access Manager 6 2005Q1 Installation Worksheets" on page 89
  - Administration Worksheet
  - Directory Server Worksheets
  - Web Container Worksheets
- "Schema Upgrade Scripts Worksheet" on page 94

Pre-Upgrade Script Worksheet

Table A-1 Pre-Upgrade Script (pre61to62upgrade) Worksheet

Script Option	Value and Example
Directory Server Host Name (fully qualified)	Your data: ____ Example: ds.example.com
Directory Server non-SSL Port Number	Your data: ____ Example: 389 (default)
DN of the Access Manager Top-level Administrator (amadmin)	Your data: ____ Example: uid=amadmin,ou=people,dc=example,dc=com
Password for the DN of the Access Manager Top-level Administrator	Your data: ____ Caution: Consider using a hint rather than the actual password.
Backup Directory (where the script should back up Access Manager files)	Your data: ____ Example: /opt/is_backup
Certificate Directory of the Web Container	Your data: ____ Example: /opt/SUNWwbsvr/alias

Access Manager 6 2005Q1 Installation Worksheets

Use the following worksheets before you run the Sun Java Enterprise System 2005Q1 installer or `amconfig` script:

- Administration Worksheet
- Web Container Worksheets
- Directory Server Worksheets

Administration Worksheet

Table A-2 Access Manager: Administration (1 of 6) Worksheet

Installation Option	Access Manager Value
Administrator User ID (amadmin)	Your data: ________________________________ Example: amadmin (default)
Administrator User ID (amadmin) Password	Your data: ________________________________ Caution: Consider using a hint rather than the actual password.
LDAP User ID (amldapuser)	________________________________ Example: amldapuser (default)
LDAP User ID (amldapuser) Password	Your data: ________________________________
Password Encryption Key	Your data: ________________________________ **Important** If you are deploying multiple instances of Access Manager, all instances must use the same password encryption key.

Web Container Worksheets

Table A-3 Access Manager: Web Container (2 of 6) Worksheet

Installation Option	Access Manager Value
Web Container for Access Manager	Your data: ______________________________ Example: Web Server or Application Server

Table A-4 Access Manager: Sun Java System Web Server (3 of 6) Worksheet

Installation Option	Access Manager Value
Host Name	Your data: ______________________________ Example: webhost.example.com
Web Server Port	Your data: ______________________________ Example: 80 (default)
Web Server Instance Directory	Your data: ______________________________ Example: /opt/SUNWwbsvr/https-myinstance
Document Root Directory	Your data: ______________________________ Example: /opt/SUNWwbsvr/docs
Secure Server Instance Port	Your data: ______________________________

Table A-5 Access Manager: Web Container for Running Access Manager Services (4 of 6) Worksheet

Installation Option	Access Manager Value
Host Name	Your data: ____________________
Services Deployment URI	Your data: ____________________ Example: amserver (default)
Common Domain Deployment URI	Your data: ____________________ Example: amcommon (default)
Cookie Domain	Your data: ____________________ Example: .example.com
Administration Console	Your response (required): Deploy new console
Console Deployment URI	Your data: ____________________ Example: amconsole (default)
Password Deployment URI	Your data: ____________________ Example: ampassword (default)
Console Host Name	Your data: ____________________ Example: console.example.com
Console Port	Your data: ____________________ Example: 80

Directory Server Worksheets

Table A-6 Access Manager: Directory Server Information (5 of 6) Worksheet

Installation Option	Access Manager Value
Directory Server Host Name (fully qualified)	Your data: ____ Example: ds.example.com
Directory Server Port	Your data: ____ Example: 389 (default)
Access Manager Directory Server Root Suffix	Your data: ____ Example: dc-example,dc=com
Directory Manager DN	Your data: ____ Example: ”cn=Directory Manager” (default)
Directory Manager DN Password	Your data: ____ Caution: Consider using a hint rather than the actual password.

Table A-7 Access Manager: Directory Server Information (6 of 6) Worksheet

Installation Option	Access Manager Value
Is Directory Server provisioned with user data?	Your response: Yes. Also, provide values for the marker and naming attributes in the next row.
Marker and Naming Attributes	Organization Marker Object Class Default: SunISManagedOrganization Your data: ____ Organization Naming Attribute Default: o Your data: ____ User Marker Object Class Default: inetorgperson Your data: ____ User Naming Attribute Default: uid Your data: ____

Schema Upgrade Scripts Worksheet

Table A-8 Schema Upgrade Scripts (Upgrade61DitTo62 and amupgrade) Worksheet

Option	Value
Directory Server Hostname (fully qualified)	Your data: ____ Example: ds.example.com
Directory Server non-SSL Port Number	Your data: ____ Example: 389 (default)
Directory Manager DN	Your data: ____ Example: ”cn=Directory Manager” (default)
Directory Manager DN Password	Your data: ____ Caution: Consider using a hint rather than the actual password.
DN of the Top-level Access Manager Administrator (amadmin)	Your data: ____ Example: uid=amadmin,ou=people,dc=example,dc=com
Password of the Top-level Access Manager Administrator	Your data: ____

Glossary

Refer to the *Sun Java™ Enterprise System Glossary* (`http://docs.sun.com/doc/816-6873`) for a list of terms that are used in this documentation set.

Index