Migration Guide

Sun™ ONE Identity Server

Version 6.1

816-6771-10
December 2003

Copyright © 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

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

THIS PRODUCT CONTAINS CONFIDENTIAL INFORMATION AND TRADE SECRETS OF SUN MICROSYSTEMS, INC. USE, DISCLOSURE OR REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR EXPRESS WRITTEN PERMISSION OF SUN MICROSYSTEMS, INC.

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

This distribution may include materials developed by third parties.

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

Sun, Sun Microsystems, the Sun logo, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, the Duke logo, the Java Coffee Cup logo, the Solaris logo, the SunTone Certified logo and the Sun ONE logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems, Inc.

Legato and the Legato logo are registered trademarks, and Legato NetWorker, are trademarks or registered trademarks of Legato Systems, Inc. The Netscape Communications Corp logo is a trademark or registered trademark of Netscape Communications Corporation.

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

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

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

Copyright © 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuels relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plus des brevets américains listés à l'adresse http://www.sun.com/patents et un ou les brevets supplémentaires ou les applications de brevet en attente aux Etats - Unis et dans les autres pays.

CE PRODUIT CONTIENT DES INFORMATIONS CONFIDENTIELLES ET DES SECRETS COMMERCIAUX DE SUN MICROSYSTEMS, INC. SON UTILISATION, SA DIVULGATION ET SA REPRODUCTION SONT INTERDITES SANS L AUTORISATION EXPRESSE, ECRITE ET PREALABLE DE SUN MICROSYSTEMS, INC.

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

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, le logo Duke, le logo Java Coffee Cup, le logo Solaris, le logo SunTone Certified et le logo Sun[tm] ONE sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

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

Le logo Netscape Communications Corp est une marque de fabrique ou une marque déposée de Netscape Communications Corporation.

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

Les produits qui font l'objet de ce manuel d'entretien et les informations qu'il contient sont regis par la legislation americaine en matiere de controle des exportations et peuvent etre soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucleaires, des missiles, des armes biologiques et chimiques ou du nucleaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou reexportations vers des pays sous embargo des Etats-Unis, ou vers des entites figurant sur les listes d'exclusion d'exportation americaines, y compris, mais de maniere non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une facon directe ou indirecte, aux exportations des produits ou des services qui sont regi par la legislation americaine en matiere de controle des exportations et la liste de ressortissants specifiquement designes, sont rigoureusement interdites.

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

Contents

About This Guide

The *Sun*™ ONE *Identity Server Migration Guide* provides instructions for using Sun Open Net Environment (Sun ONE) Identity Server. The guide includes instructions for upgrading an existing Identity Server deployment to version 6.1, and for migrating user data from an existing Sun ONE Directory Server. This preface contains the following sections:

- Audience for This Guide
- Identity Server 6.1 Documentation Set
- Documentation Conventions Used in This Guide
- Related Information

Audience for This Guide

This *Migration Guide* is intended for use by IT administrators and software developers who implement an integrated identity management and web access platform using Sun ONE servers and software. It is recommended that administrators understand the following technologies:

- Lightweight Directory Access Protocol (LDAP)
- Java™
- JavaServer Pages™ (JSP)
- HyperText Transfer Protocol (HTTP)
- HyperText Markup Language (HTML)
- eXtensible Markup Language (XML)

Because Sun ONE Directory Server is used as the data store in an Identity Server deployment, administrators should also be familiar with the documentation provided with that product. The latest Directory Server documentation can be accessed online.

Identity Server 6.1 Documentation Set

The Identity Server documentation set is separated into two core sets of manuals: the Sun ONE Identity Server 6.1 core application manuals and the Sun ONE Identity Server Policy Agents books.

Identity Server Core Documentation

The Identity Server documentation set contains the following titles:

- *Product Brief* provides an overview of the Identity Server application and its features and functions.
- *Migration Guide* provides details on how to migrate existing data and Sun ONE product deployments to the latest version of Identity Server. For instructions on installing Identity Server, see the *Sun Java Enterprise System 2003Q4 Installation Guide*.
- *Administration Guide* describes how to use the Identity Server console as well as manage user and service data via the command line.
- *Customization and API Guide* documents how to customize an Identity Server installation. It also includes instructions on how to augment the application with new services using the public APIs.
- *Deployment Guide* provides information on planning an Identity Server deployment within an existing information technology infrastructure.
- The *Release Notes* will be available online after the product is released. They gather an assortment of last-minute information, including a description of what is new in this current release, known problems and limitations, installation notes, and how to report issues with the software or the documentation.

Updates to the *Release Notes* and links to modifications of the core documentation can be found on the Identity Server page at the Sun ONE documentation web site. Updated documents will be marked with a revision date.

Identity Server Policy Agent Documentation Set

Policy agents for Identity Server 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 Identity Server documentation. The following titles are included in the set:

- *Policy Agents For Web And Proxy Servers Guide* documents how to install and configure an Identity Server 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 Identity Server policy agent that can protect a variety of hosted J2EE applications. It also includes troubleshooting and information specific to each agent.
- The *Release Notes* will be available online after the set of agents is released. There is generally one *Release Notes* file for each agent type release. The *Release Notes* gather an assortment of last-minute information, including a description of what is new in this current release, known problems and limitations, installation notes, and how to report issues with the software or the documentation.

Updates to the *Release Notes* and modifications to the policy agent documentation can be found on the Policy Agents page at the Sun ONE documentation web site. Updated documents will be marked with a revision date.

Your Feedback on the Documentation

Sun Microsystems and the Identity Server technical writers are interested in improving the documentation and welcome any comments and suggestions. Please email these comments to `docfeedback@sun.com`.

Documentation Conventions Used in This Guide

In the Identity Server documentation, certain typographic conventions and terminology are used. These conventions are described in the following sections.

Typographic Conventions

This book uses the following typographic conventions:

- *Italic type* is used within text for book titles, new terminology, emphasis, and words used in the literal sense.
- `Monospace font` is used for sample code and code listings, API and language elements (such as function names and class names), filenames, pathnames, directory names, HTML tags, and any text that must be typed on the screen.
- *Italic serif font* is used within code and code fragments to indicate variable placeholders. For example, the following command uses *filename* as a variable placeholder for an argument to the `gunzip` command:

  ```
  gunzip -d filename.tar.gz
  ```

Terminology

Below is a list of general terms used in the Identity Server documentation set:

- *Identity Server* refers to Identity Server and any installed instances of the Identity Server software.
- *Policy and Management services* refers to the collective set of Identity Server components and software that are installed and running on a dedicated deployment container such as a web server.
- *Directory Server* refers to an installed instance of Sun ONE Directory Server.
- *Application Server* refers to an installed instance of Sun ONE Application Server.
- *Web Server* refers to an installed instance of Sun ONE Web Server.
- *IdentityServer_base* is a variable place holder for the home directory where you have installed Identity Server.
- *DirectoryServer_base* is a variable place holder for the home directory where you have installed Sun ONE Directory Server.

- *ApplicationServer_base* is a variable place holder for the home directory where you have installed Sun ONE Application Server.
- *WebServer_base* is a variable place holder for the home directory where you have installed Sun ONE Web Server.
- *Web container that runs Identity Server* refers to the dedicated J2EE container (such as Web Server or Application Server) where the Policy and Management Services are installed.

Related Information

In addition to the documentation provided with Identity Server, there are several other sets of documentation that might be helpful. Table 1 lists these and additional sources of information.

Table 1 Where to Find Related Sun ONE Resources

Information or Resource	**Internet Location**
Directory Server documentation	http://docs.sun.com/coll/S1_DirectoryServer_52
Web Server documentation	http://docs.sun.com/coll/S1_websvr61_en
Web Proxy Server documentation	http://docs.sun.com/prod/s1.webproxys#hic
Sun ONE Download Center	http://wwws.sun.com/software/download/
Sun ONE Technical Support	http://www.sun.com/service/sunone/software/index.html
Sun ONE Professional Services Information	http://www.sun.com/service/sunps/sunone/index.html
Sun Enterprise Services, Solaris Patches and Support	http://sunsolve.sun.com/
Developer Information	http://developers.sun.com/prodtech/index.html

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 by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Chapter 1

Upgrading from Identity Server 6.0 to Identity Server 6.1

This chapter provides instructions for upgrading Sun ONE Identity Server versions 6.0 and 6.0 SP1 to Identity Server 6.1. Topics include:

- Before You Begin
- Overview of Upgrade Tasks
- Installing Sun ONE Web Server 6.1
- Installing Identity Server 6.1

Before You Begin

In order to upgrade from Identity Server 6.0 to version 6.1, you must run the Java Enterprise System installer. Use the *Java Enterprise System Installation Guide* for detailed information about hardware and software requirements, and about other pre-installation issues you need to resolve. Also make note of the following points, and resolve any additional issues that apply to your Identity Server 6.0 deployment:

- The Identity Server 6.1 upgrade procedure described in this chapter can be performed only on UNIX platforms. There is no upgrade procedure for Windows platforms at this time.
- If your Identity Server 6.0 deployment is based on Sun ONE Web Server 6.0, you must upgrade to Web Server 6.1. **This is required.** See "Installing Sun ONE Web Server 6.1" on page 16.

- If your Identity Server 6.0 deployment is based on Sun ONE Application Server 7.0, you do not have upgrade the Application Server. Identity Server 6.1 will run against Application Server 7.0.
- This chapter provides separate instructions for installing Web Server 6.1 and Identity Server 6.1. This is intended to make the instructions more clear when you upgrade an instance of Identity Server for the first time. In practice, you can use the Java Enterprise System installer to install both products during the same session. This will save you time when you need to update multiple instances of Identity Server.
- When upgrading multiple instances of Identity Server, be sure to stop all instances of Identity Server that are connected to the same Directory Server. Upgrade the first Identity Server, and then restart it. Continue to upgrade and restart each additional Identity Server instance one at a time.
- If your Identity Server 6.0 deployment is based on Sun ONE Directory Server 5.1, you should determine whether you want to upgrade to Directory Server 5.2. Identity Server 6.1 supports both versions. If you want to upgrade to Directory Server 5.2, follow the instructions in the *Sun ONE Directory Server 5.2 Installation Guide*.

Overview of Upgrade Tasks

Note that you can combine steps 2 and 3 into one step by using the Java Enterprise System installer to install both Web Server 6.1 and Identity Server 6.1 during the same session. This chapter provides separate installation instructions for each product simply to make the instructions more clear when you're upgrading your first Identity Server instance.

1. Run the Identity Server pre-upgrade script.

   This script backs up your current Identity Server 6.0 configuration files, removes all Identity Server 6.0 packages, removes Web Server 6.0 packages, and cleans up the product registry.

   See "To Run the Pre-Upgrade Script" on page 15 for detailed information.

2. Run the Java Enterprise System installer to install Web Server 6.1.

   If you installed Identity Server 6.0 against Application Server 7.0, this step is not necessary. Identity Server 6.1 will work with Application Server 7.0.

   See "Installing Sun ONE Web Server 6.1" on page 16 for detailed information.

3. Run the Java Enterprise System installer to install Identity Server 6.1.

   During installation, the new packages used in Identity Server 6.1 are added.

   See "To Install Identity Server 6.1" on page 20 for more information.

4. Run the Identity Server post-upgrade script.

   This script adds Identity Server 6.1 schema to the directory tree, adds new Identity Server services, and updates existing Identity Server services, attributes, and other directory entries as necessary.

   See "To Run the Post-Upgrade Script" on page 26 for detailed information.

5. If you customized your Identity Server 6.0 installation, then you must manually redo the customization in your Identity Server 6.1 installation.

Using the Pre-Upgrade Script

Before you run the pre-upgrade script, be sure that Directory Server is up and running, and that Identity Server 6.0 and Web Server 6.0 are stopped. Once you run the script, be sure to allow it to finish completely. If you stop the script before it has completely finished, the results will be unpredictable.

CAUTION The changes you make when you run the pre-upgrade script are permanent and cannot be undone.

Detailed instructions are included in the following steps.

To Run the Pre-Upgrade Script

1. Make sure that Directory Server is up and running. Restart Identity Server 6.0 by following these steps:

   a. In the directory *IdentityServer_base* / `SUNWam/bin`, execute the `amserver stop` command.

   b. In the directory *IdentityServer_base*/`SUNWam/bin`, execute the `amserver start` command.

2. Make sure that Identity Server and Web Server 6.0 are stopped. In the directory *IdentityServer_base*/`SUNWam/bin`, execute the `amserver stop` command.

3. In the directory where the Java Enterprise System installer exists, run the script pre-upgrade script. Example:

   ```
   # cd JavaEnterpriseSystem_Base/Product/identity_srv/Tools
   ```

   ```
   #./pre60to61upgrade
   ```

4. When prompted, provide the following information:

   Directory Server fully-qualified hostname: Enter the host name of the system where Directory Server is installed. Example: *hostName.domainName*.

   Directory Server port: The default value is 389.

   Top-Level Administrator DN: Enter the DN of the administrator who has unrestricted access to Identity Server entries in the directory.
   Example: `uid=amAdmin,ou=People,dc=iplanet,dc=com`

   Top-Level Administrator password: Enter the password for the Top-Level Administrator specified above.

   Enter directory to store back up files: Enter the path to the directory where you want to store back up Identity Server files. Example: `/opt`

Installing Sun ONE Web Server 6.1

This chapter provides separate instructions for installing Web Server 6.1 and Identity Server 6.1. This is intended to make the instructions more clear when you upgrade an instance of Identity Server for the first time. In practice, you can use the Java Enterprise System installer to install both products during the same session. This will save you time when you need to update multiple instances of Identity Server.

NOTE When upgrading Identity Server for the first time, it's a good practice to install Web Server 6.1, and then confirm that installation was successful, before installing Identity Server 6.1. If the Web Server installation is unsuccessful, then Identity Server installation will fail.

For the most part, you should refer to the documentation that comes with the Java Enterprise System installer for system requirements and detailed installation steps. However, during installation you must provide some parameters that were used when Web Server 6.0 was installed. Be sure to review the following steps before you start the Java Enterprise System installer.

To Install Sun ONE Web Server 6.1

1. In the directory where the Java Enterprise System installer exists, run the `installer` command:

   `./installer`

2. In the Welcome window, click Next.

3. In the Software License Agreement window, to accept the terms, click “Yes, Accept License.”

4. In the Language Support window, select the language applicable to your system, and click Next.

5. In the Component Selection window, select only "Sun ONE Web Server 6.1.” Be sure that all other components are deselected, and then click Next.

6. If a message similar to the following is displayed:

```
JAVA[tm] 2 Standard Edition Software Development Kit (SDK) Update Required.

Sun Java(tm) Enterprise System requires a different version thatn the one
currently installed.

Upgrade the existing J2SE SDK?
```

   then select “Upgrade the exsiting J2SE SDK,” and then click OK. Otherwise, skip to the next step.

7. In the Shared Component Upgrades Required window, click Next.

8. In the Installation Directories window, provide the following information, and then click Next:

   Web Server: Enter the path to the directory where Web Server will be installed. The the default location is `/opt/SUNWwbsvr`.

9. In the Checking System Requirements, after the checking is finished, click Next.

10. In the Configuration Type Panel, select Custom Configuration, and then click Next.

11. In the Common Server Settings window, provide the following information, and then click Next.

    Host Name: Enter the fully-qualified name of the computer system where Web Server will be installed.

    DNS Domain Name: The default value is displayed.

    Host IP Address: The default value is displayed.

    Administrator User ID: Enter the same user ID specified for the web Server when Identity Server 6.0 was installed. The default user ID is `admin`.

    Administrator Password: Enter the same value that was entered when Identity Server 6.0 was installed.

    Retype Password: Enter the same password again to confirm it.

    System User: Enter the name of the user ID to run Web Server as. The default is `root`.

    System Group: Enter the name of the group to run Web Server as. The default is `other`.

12. In the Web Server: Administration window (1 of 2), provide the following information, and then click Next:

    Administrator User ID: Enter the same value you entered when you installed Identity Server 6.0. The default value is carried forward from the Server Settings window.

    Administrator Password: Enter the same value you entered when you installed Identity Server 6.0. The default value is carried forward from the Server Settings window.

    Retype Password: Enter the same password again to confirm it.

    Web Server Domain Name: Enter the fully-qualified name of the computer system where Web Server will be installed.

    Administration Port: The default value is 8888. The Identity Server 6.0 default value was 58888.

    Administration Runtime User ID: The default value is root.

13. In the Web Server: Default Web Server Instance window (2 of 2), provide the following information, and then click Next:

    Runtime User ID: The default value is `root`.

    Runtime Group: The default value is `other`.

    HTTP Port: The default value is 80. The default value for Identity Server 6.0 was `58080`.

    Document Root Directory: Enter the path to the directory where content documents are stored. Example:

    `/opt/SUNWwbsvr/docs`

    Automatically start Web Server when system restarts. By default, this is enabled.

14. In the Ready to Install window, when you're satisfied that all selections are correct, click Next. To change any of your settings, click Back.

15. In the Product Registration window, Next to continue.

16. If a `regtool.sh` message similar to the following is displayed:

```
Dear User,

We have found that <hostName> has not been registered, or the information
has changed since it was last registered. The registration information is
very important for SunIR staff to ensure that your system is secure,
copyright-compliant, and data protective.

...
```

then enter `yes` to continue.

17. If you selected "Open registration window during installation" in step Step 14, then click Getting Started window will be displayed. In the browser window, click File>Close to dismiss the Getting Started window and to end the installation program. Skip to Step 19.

18. If the Installation Complete window is displayed, then click click Close.

19. (Optional) Check to make sure that installation succeeded.

a. An instance of Web Server should exist in the appropriate installation directory. Example:

   `/opt/SUNWwbsvr/https-`*hostName.domainName*

b. You should be able to point a browser to the new Web Server instance.

   First, start the Web Server. In the directory *WebServer_base*`/SUNWam/servers/https-`*hostName.domainName*, execute the `start` command.

   Then point a browser to the following URL:

   `http://`*hostName.domainName*`:58080`

   where *hostName.domainName* is the name of the host computer where you installed Web Server 6.1, and 58080 is the Web Server port number you specified. The Web Server home page should display.

Installing Identity Server 6.1

To install Identity Server 6.1, you must run the Java Enterprise System installer. For the most part, you should refer to the documentation that comes with the Java Enterprise System installer for system requirements and detailed installation steps. However, during installation you must provide some parameters that were used when Identity Server 6.0 was installed. Be sure to review the following steps before you start the Java Enterprise System installer.

To Install Identity Server 6.1

1. In the directory where the Java Enterprise System installer exists, run the `installer` command:

```
./installer
```

2. In the Welcome window, click Next.
3. In the Software License Agreement window, to accept the terms, click “Yes, Accept License.”
4. In the Language Support window, select the language applicable to your system, and click Next.

5. In the Component Selection window, under “Sun ONE Identity Server 6.1,” expand the Identity Server 6.1 listing, and then check the following option:

   `Identity Management and Policy Services Core`

   If the Sun ONE Message Queue3.0.1 SP2 component is automatically selected, leave it selected. The Sun ONE Identity Server 6.1 component will be automatically be selected; leave it selected.

   Then make sure that all other components are deselected, and click Next.

6. If a message similar to this one displays, ignore it and click Continue:

```
"Product Dependency Checks"
    You have not selected to install any Point product to satisfy the
    dependency required by Identity Server. Instead you
    could have selected: SunONEWebServer, appserv, External Containers You
    haven’t selected to install a local Directory Server required by these
    Products: Identity and Service Management Services. You
    can point to a remote installation of Directory Server when you configure
    the products mentioned in the list. However if you wish, you could
    also select to install Directory Server locally and use it for
    configuring Identity and Service Management Services.
```

7. In the Installation Directories window, enter the path to the directory where you want to install Identity Server 6.1, and then click Next.

   Identity Server: Enter the path to the directory where you want to install Identity Server 6.1. The default is `/opt`.

8. In the Checking System Requirements window, when checking is completed, click Next.

9. In the Configuration Type Panel, select Custom Configuration, and then click Next.

10. In the Common Server Settings window, provide the following information, and then click Next:

    Host Name: Enter the name of the computer system where Identity Server will be installed.

    DNS Domain Name: The default value is displayed.

    Host IP Address: The default value is displayed.

    Administrator User ID: The default value is `admin`.

    Administrator Password: Enter the same value that was entered for `amadmin` when Identity Server 6.0 was installed.This value is carried forward to the Identity Server: Administration window.

    Retype Password: Enter the same password again to confirm it.

    System User: Enter the name of the user ID to run Identity Server as. The default is `root`.

    System Group: Enter the name of the group to run Identity Server as. The default is `other`.

11. In the Identity Server: Administration window (1 of 6), provide the following information, and then click Next:

    Administrator User ID: The default is `amadmin`. You cannot change this value.

    Administrator Password: Enter the password that was used for Identity Server 6.0 `amadmin`. The default value is carried forward from the Server Settings window.

    Retype Password: Enter the password again to confirm it if necessary.

    LDAP User ID: This is the user ID that was used in Identity Server 6.0 to bind to the directory for authentication. The default is `amldapuser`. This value cannot be changed.

    LDAP Password: Enter the password specified for the Identity Server 6.0 `amldapuser`.

    Retype Password: Enter the password again to confirm it.

    Password Encryption Key: Change the default value to the Identity Server 6.0 encryption key:

    ```
    KmhUnWR1MYWDYW4xuqdF5nbm+CXIyOVt
    ```

12. In the Identity Server: Web Container window (2 of 6), provide the following information, and then click Next:

    Web Container: For this example, select “Sun ONE Web Server.”

13. In the Identity Server: Web Container window (3 of 6), provide the following information, and then click Next:

    Host Name: Enter the fully-qualified name of the computer system where Identity Server 6.1 is to be installed. Example: *hostName.domainName*

    Web Server Port: The default is 80. The default value for Identity Server 6.0 is `58080`.

    Web Server Instance Directory: Enter the path to the directory where the Web Server instance exists. Example: `/opt/SUNWwbsvr/https-`*hostName*`.`*domainName*

    Document Root Directory: Enter the path to the directory were the server’s content documents exist. Example: `/opt/SUNWwbsvr/docs`

    Is server instance port secure? Mark this box if **t**o indicate that the web server is Secure Sockets Layer (SSL)-enabled. Leave the box unmarked if the web server is not SSL-enabled.

14. In the window titled “Identity Server: Web Container for running Sun ONE Identity Server Services (4 of 6),” provide the following information, and then click Next:

    Host: Enter the fully-qualified hostname of the computer system where the Web Container is installed. Example: *hostName.domainName*

    Services Deployment URI: The Universal Resource Identifier (URI) prefix is a string that tells the Web Server where to look for HTML pages associated with a service and also for web application-specific information such as classes and jars. Enter the string that was entered when Identity Server 6.0 was installed. The default URI prefix is `amserver`.

    Common Domain Deployment URI: This URI determines the mapping between a string you specify and a deployed application.Enter the string that was entered when Identity Server 6.0 was installed. The default is `amcommon`.

    Cookie Domain: Enter the name of the domain for which Identity Server sets cookies. Example: *.domainName*

    Deploy Console with this service? Select Yes.

    Console Host (for existing console): If an administration console was installed for Identity Server 6.0, this is the name of the computer on which the console was installed. It cannot be changed. Example: *hostName.domainName*

Console Port (for existing console): If an administration console was installed for Identity Server 6.0, this value is taken from the previous window. It cannot be changed.

Console Deployment URI: This URI prefix tells the Web Server where to look for HTML pages associated with the Identity Server console and also for other web application-specific information such as classes and jars. The default URI prefix is `amconsole`. Enter the string that was entered when the Identity Server 6.0 console was installed.

Password Deployment URI: Enter the URI to be used for the new Password Reset service. The default is `ampassword`.

15. In the window Identity Server: Directory Server Information window (5 of 6), provide the following information, and then click Next:

    Directory Server Host: Enter the fully-qualified name of the computer system on which Directory Server is installed.

    Directory Server Port: Enter the port number for the Directory Server you specified.The default value for Identity Server 6.0 is `389`.

    Identity Server Directory Root Suffix: Enter the base DN that indicates the part of the directory tree that is managed by Identity Server.

    Example: `dc=iplanet,dc=com`

    Refer to the `com.iplanet.am.rootsuffix` property in the Identity Server 6.0 `AmConfig.properties` file.

    Directory Manager DN: Enter the base DN of the who has unrestricted access to the Directory Server. Enter the Example: `cn=Directory Manager`

    Directory Manager Password: Enter the same Directory Manager password that was entered when Identity Server 6.0 was installed.

16. In the Identity Server: Existing Provisioned Directory window (6 of 6), provide the following information, and then click Next:

    Is the Directory Server provisioned with user data? Click Yes.

    Organization Marker Object Class: Enter the object class used for organizations in the existing directory tree. The default is `sunManagedOrganization`.

    Organization Naming Attribute: Enter the naming attribute used for organizations in the existing directory tree. The default is `o`.

    User Marker Object Class: Enter object class used for users in the existing directory tree. The default is `iplanet-am-user-service`.

    User Naming Attribute: Enter the naming attribute used for users in the existing directory tree. The default is `uid`.

17. In the Ready to Install window, when you're satisfied that the settings are correct, click Next. To make changes to your settings, click Back.

18. In the Product Registration window, Next to continue.

19. If you selected "Open registration window during installation" in step Step 14, then click Getting Started window will be displayed. In the browser window, click File>Close to dismiss the Getting Started window and to end the installation program. Skip to Step 19.

20. If the Installation Complete window is displayed, click Close.

Using the Post-Upgrade Script

The post-upgrade script updates schema in your existing directory tree from Identity Server 6.0 to Identity Server 6.1.

NOTE	Before you run the post-upgrade script, be sure that Directory Server is up and running, and that Web Server is *not* running. In the middle of the script, you'll be asked to restart the Directory Server before the script can continue. You will also be instructed to restart both the Directory Server and Web Server in order for the changes to take effect.

To Run the Post-Upgrade Script

1. Run the script `Upgrade60DitTo61`. You'll find this script in the following location:

   IdentityServer_Base`/SUNWam/migration/60to61/scripts`

2. When prompted, provide the following information:

   Directory Server fully-qualified hostname: Enter the host name of the system where Directory Server is installed. Example: *hostName.domainName*

   Directory Server port: The default value is 389.

   Directory Manager DN: Enter the DN of the user who has unrestricted access to Directory Server. Example: `cn=Directory Manager`.

   Directory Manager password: Enter the password for the Directory Manager specified above.

   Top-Level Administrator DN: Enter the DN of the administrator who has unrestricted access to Identity Server entries in the directory.
   Example: `uid=amAdmin,ou=People,dc=iplanet,dc=com`

   Top-Level Administrator password: Enter the password for the Top-Level Administrator specified above.

3. When prompted by the script, restart Directory Server.

4. When Directory Server has been restarted, return to the script and press Enter to continue.

5. After the script has completed, the following message is displayed:

```
YOU MUST RESTART THE DIRECTORY AND WEB SERVERS FOR THE UPGRADE CHANGES TO
TAKE EFFECT.
```

First restart Directory Server, and then restart Web Server. After both servers have been restarted, you should verify that the upgrade was successful. To start Identity Server, use a browser to go to the following URL:

`http://`*hostName*`.`*domainName*`:58080/amconsole`

Chapter 2

Upgrading from DSAME 5.1 to Identity Server 6.0

This appendix provides steps for upgrading a iPlanet DSAME 5.1 installation to Sun ONE Identity Server 6.0. It includes reference sections that describe specific changes between DSAME 5.1 and Identity 6.0.

The following topics are included in this appendix:

Overview of Procedures and Migration Scripts

It is expected that the person conducting upgrade procedures is familiar with Directory Server commands, schema semantics, directory information trees (DITs) Identity Server schema and Identity Server DIT structures. In addition, familiarity with XML and Identity Server installation procedure are required.

Roadmap of Upgrading Tasks

You can think of the upgrade process as having three major parts:

1. Before you can begin migrating data, you must uninstall DSAME 5.1, and install Identity Server 6.0. For detailed instructions, see "Pre-Data Migration Tasks" on page 34.
2. Data Migration Tasks

   During data migration, first you migrate DSAME schema, policies, authentication entries, and services. Then in Identity Server 6.0, you update authentication entries, console service entries, and policies. For detailed instructions, see "Data Migration Tasks" on page 38.
3. Post-Data Migration Tasks

   After you've migrated data, you can enable Federation Management. Then you must re-do any customization you may have done in the DSAME 5.1 console or in any pre-6.0 agents. There are also a small number of miscellaneous modifications you must make to Identity Server 6.0 configuration. For detailed instructions, see "Post-Data Migration Tasks" on page 46.

The Figure 2-1 on page 32 provides a detailed summary or roadmap of the tasks required for upgrading a DSAME 5.1 installation. Use the roadmap as a checklist as you proceed through the upgrade tasks.

Migration Scripts

Identity Server 6.0 provides a set of Perl scripts to migrate DSAME 5.1 data to Identity Server 6.0. The migration procedure is complex, but these scripts handle many of those complexities. Typically, scripts generate an input file and an output file. Both these files are retained after the scripts are run. This helps in checking the entries in output files. The input files will contain the entries in 5.1 format, while the output files will have the entries in 6.0 format.

When using these migration scripts, keep the following in mind:

- The output file needs to be loaded using `ldapmodify` command. It is not done automatically so that entries in the file can be checked before loading them.
- If you are running a script more than once, make sure to remove the old input and output files generated by the script. Some scripts append the output to an existing file because of which you may get errors while running `ldapmodify` commands.
- Each script contains additional information in them, which you must read before running the script. Additionally, in each script, you may have to set some variables or check the values of the variables before running the script.
- The migration scripts are case sensitive. The scripts will look for Identity Server attributes, object classes, and values *that are in lower case*. If you've customized your Identity Server 5.1 deployment with attribute names or object class names that contain upper case letters, then before running the scripts you must change those names to lower case letters.

NOTE The steps mentioned above represents the generic migration procedure. The scripts can be used in different ways to do the migration. For example, the existing Directory Server available with DSAME 5.1 can exported and loaded into a new Directory Server. The migration scripts can be run on this new Directory Server.

Figure 2-1 Roadmap of upgrade tasks.

1. **Back up Existing Installation**	a. Use DS tools to back up DS. b. Use copy & tar to manually back up WS.
2. **Uninstall Dsame 5.1**	Run DSAME Uninstall program.
3. **Install Identity Server Schema**	Run IS6.0 Installer.
4. **Install Identity Server**	Run IS6.0 Installer.
5. **Migrate Schema changes**	a. Run **update-schema.pl** to generate 51entries.ldif and 60entries.ldif. b. Run ldapmodify on 60entries.ldif.
6. **Migrate DSAME 5.1 Policies**	a. Run **update-policies.pl** to generate XML files for organizations that have policies. b. If domain URL policies defined, manually back them up now. c. For each domain URL policy, you may need to manually create a referral policy.
7. **Migrate Authentication Entries**	Run **update-auth.pl** to generate 51to60auth-entries.ldif and 51auth-entries.dn.
8. **Migrate Services**	a. If top-level entry is not an org, run **update-toporg-services.pl** to generate 51to60toporg-template.ldif and 51toporg-template.dn. Or If top-level entry is an org, use DS console to remove DSAME services. b. Run **load-services.pl** to load IS 6.0 services. c. If top-level org is an IS org, load 51to60toporg-template.ldif.

9. **Update Authentication Entries to 6.0**	a. Load 51to60-auth-entries.ldif. b. Change any customized 5.1 HTML templates to 6.0 JSP-based templates. c. Use AMLoginModule.java to rewrite any customized auth modules. d. Change screen properties to use XML-based Auth module properties. e. Manually migrate User's auth module configuration. f. Migrate and remove user's default login URL attribute.
10. **Update IS Console Service Entries**	a. Run **update-services.pl** to generate 60services.ldif. b. Run ldapmodify on 60services.ldif .
11. **Update Policies to 6.0**	a. Run **delete-policies.pl** to generate delete-policies.ldif. b. Run ldapdelete on delete-policies.ldif to delete all 5.1 policies. c. For each org, register Policy Config service. d. Load IS. 60 policies. f. If any domain URL services, create referral policies.. e. If any sub-orgs, create referral policies from top-level org. g. For each org, create policy admin role.
12. **Enable Federation Management**	Run ldapmodify on liberty-services.ldif to register iPlanetAMAuthenticationDomainConfigService and iPlanetAMProviderConfigService.
13. **Miscellaneous Upgrade Tasks**	a. If DS is on same host as IS, edit amserver script to point to DS instance. b. Restart IS and log in to 6.0 Console. c. Enter amldapuser password for appropriate auth services. d. Create amldapuser. e. Change date format in amUser.properties. f. Any changes in AMConfig.properties for DSAME 5.1 should be reflected in AMConfig.properties in IS6.0.
14. **Migrate Console Changes**	In 6.0, re-do any console customization configuartion.
15. **Migrate Agents**	a. Back up any configuration changes made in Agents 1.0 or 1.1. b. Install 2.0 agents. c. Re-do any agent customization configuration. d. Restart the agents.

Pre-Data Migration Tasks

Before you can begin data migration, you must uninstall DSAME 5.1 and install Identity Server 6.0 schema. It's always a good practice to back up your existing installation before making changes of this scope.

Backing up the Existing Installation

Be sure that the DSAME 5.1 installation is completely backed up.

- Back up Directory Server data in DSAME 5.1 using Directory Server backup tools. The backup should include all 5.1 data including configuration and schema.
- Back up Web Server data by copying any data modified after the DSAME 5.1 installation.

The Web Server data must be backed up manually. You can use the `copy` and `tar` commands for this. Any changes done to the Web Server files after the 5.1 installation must be backed up. Table 2-1 lists directories must also be backed up, and provides a brief descript of what is contained in each directory.

Table 2-1 Directories to be backed up

Directory to be backed up	**Contents**
`<IS install dir>/SUNWam/web-apps/applications`	IS Console files
`<IS install dir>/SUNWam/web-apps/services`	IS services files
`<IS install dir>/SUNWam/servers/alias`	certificates
`<IS install dir>/SUNWam/config`	various XML files
`<IS install dir>/SUNWam/lib/`	property files
`<IS install dir>/SUNWam/locale`	locale files

You can refer to these backups, to make corresponding changes once Identity Server 6.0 is installed. These changes need to be done manually after the Identity Server 6.0 is installed.

You should also backup logs, debug and install files. Table 2-2 lists the files that must be backed up and their location.

Table 2-2 Files to be backed up

Files	Location
log files	`/var/<IS 5.1 install dir>/SUNWam/logs`
debug files	`/var/<IS 5.1 install dir>/SUNWam/debug`
install files	`/var/<IS 5.1 install dir>/SUNWam/install`

Back up any other data that needs to be updated after the 6.0 migration.

Uninstalling DSAME 5.1

Use the DSAME 5.1 uninstallation program to remove components of DSAME. But you must *not* remove Directory Server 5.1.

On Solaris

To remove DSAME components on Solaris:

1. Run `aminstall` script from DSAME 5.1.
2. Choose the following option:

   1) Remove existing components, then continue installation

3. At the next prompt, choose the following option to remove DSAME Management and Policy services.

   1) DSAME Management and Policy Services

4. Run `aminstall` script again and choose option 1.
5. At the next prompt, choose the following option to remove Directory Server Configuration.

   3) iPlanet Directory Server Configuration for DSAME

   This will remove the schema configuration for the Directory Server.

6. If Directory Server and Identity Server exist on different computer systems, then after installation is complete, you must manually remove the DSAME 5.1 schema file `95ns-amschema.ldif` from the Directory Server schema directory.

7. Check for the `SUNWamjdk` package after the uninstallation is complete using the following command:

```
pkginfo |grep SUNWamjdk
```

8. If the `SUNWamjdk` package is present, remove it using the following command:

```
pkgrm SUNWamjdk
```

9. Restart Directory Server after you have uninstalled the DSAME components.

On Windows

To uninstall DSAME 5.1 components, follow these steps:

1. Run the DSAME 5.1 uninstallation program. Refer to DSAME 5.1 Installation and Configuration Guide for detailed steps. You can find this guide online at: `http://docs.sun.com/source/816-5626-10/`.

2. Select partial uninstallation.

3. Choose DSAME Management and Policy services.

The above steps do not remove Directory Server configuration for DSAME on Windows. In order to configure the Directory Server of DSAME 5.1 to Identity Server 6.0 schema in the next step, you must uninstall DSAME 5.1 schema configuration. Remove DSAME 5.1 schema file, `95ns-amschema.ldif` from the Directory Server schema directory. In addition, the `productregistry` file must be updated to remove the directory server configuration component. You may remove the `productregistry` file itself. Please be sure to keep a backup of the `productregistry` file. If you remove the `productregistry` file, you can choose Add/Remove programs to remove Directory Server installation later.

4. Restart Directory Server after you have uninstalled the DSAME components.

Installing Identity Server Schema

Using the Identity Server 6.0 installation program, configure Directory Server to work with Identity Server 6.0. For detailed instructions, see the section “Installing Identity Server Schema” in the *Identity Server 6.0 Installation and Configuration Guide*.

Installing Identity Server 6.0 on Directory Server 5.1

You may need to make some changes to the Directory Server that exists with DSAME. This Directory Server supports a DIT like this:

```
o=isp
|
o=siroe.com
```

Here `isp` is not really an organization. In such cases, the entry *isp* must be updated before you install Identity Server 6.0. If DSAME 5.1 Directory Server has an organization (flat DIT) as top level entry, then this change is not necessary before installing Identity Server 6.0. If the top level entry has iplanet-am-service-status attribute set, Identity Server 6.0 installation does not modify the Directory Server DIT. To retain the 5.1 DIT structure, add this attribute to the top level entry.

1. Run the following command to update the top level entry, if it is not an organization:

```
<5.1 Directory Server install dir>/shared/bin/ldapmodify -D "cn=directory
manager" -w <password>
dn: o=isp
changetype: modify
delete: objectClass
objectClass: iplanet-am-managed-org
-
add: objectClass
objectClass: sunManagedOrganization
-
add: iplanet-am-service-staus
iplanet-am-service-status:iPlanetAMAuthService
```

2. Use the right `dn` in the above command for your installation. If the top level entry in your DSAME 5.1 DIT is already an organization, then you shouldn't run the above command. You may run `ldapsearch` on the top level entry to check if this attribute is set.

3. Now, install Identity Server 6.0 on this Directory Server. During installation, choose the option of installing with an existing DIT.

4. While installing Identity Server 6.0, be sure the following entries have the same values as in DSAME 5.1 installation:

   - directory root suffix
   - directory manager password
   - admin user
   - admin password
   - directory server host
   - directory server port
   - console deployment description
   - services deployment descriptor

   Refer to the `AMConfig.properties` file from the DSAME 5.1 installation backup for any values you are not sure. Also, retain the DSAME 5.1 values for organization object class, organization naming attribute, user object class and user naming attribute.

This step does not modify Directory Server data and schema. It only installs Identity Server 6.0 packages, libraries, configuration files, jar files, etc.

Data Migration Tasks

Once Identity Server 6.0 is installed and the Directory Server schema is updated, the Directory Server data must be modified to Identity Server 6.0 format. All the migration scripts needed for this are located under the directory `<install dir>/SUNWam/migration/51to60`. The scripts contain additional information, which you must read before running the scripts. It will help you set some variables in each script or check the values of the variables.

In Identity Server 6.0, policy, authentication and console components have changed significantly from DSAME 5.1 and hence need to be migrated. However, Identity Server entries such as roles, groups, users, organizations, organizationalUnits and ACIs remain as they were in DSAME 5.1. They need not be migrated. The following entries of DSAME 5.1 need to be updated to Identity Server 6.0: Services branch, Organization, OrganizationalUnit, Policies, Roles, and Users.

Migrating DSAME 5.1 data to Identity Server 6.0 involves the following tasks:

- Migrating Schema Changes
- Migrating DSAME 5.1 Policies
- Migrating Authentication Entries
- Migrating Services
- Updating Authentication Entries to Identity Server 6.0
- Updating Policies to Identity Server 6.0
- Migrating Agents

CAUTION Migration scripts are designed to be run in a particular order. It is important to perform migration tasks in the exact sequence presented here and in the following instructions.

Migrating Schema Changes

Identity Server 6.0 has seen some schema changes from DSAME 5.1. For example, objectClass iplanet-am-managed-org for the organization has been changed to sunManagedOrganization. The attribute iplanet-am-domain-name is changed to sunPreferredDomain. Similarly, there are other schema changes. A script, `update-schema.pl` is provided to migrate schema changes. Run this script to migrate schema changes. Refer to the script for additional information. It generates an input file, `51entries.ldif` and an output file, `60entries.ldif`. Run `ldapmodify` on this output `ldif` file. The last line of the script specifies the syntax for running the `ldapmodify` command on this output file. DSAME 5.1 entries are updated to Identity Server 6.0 schema when you run this script.

Migrating DSAME 5.1 Policies

DSAME 5.1 uses Class of Service (CoS) templates for policy implementation. The policy definition does not contain subjects to which the policy is assigned. Instead, policy must be explicitly assigned to a role or an organization. When a policy is assigned to an organization or a role, a CoS template is created. There is one CoS template for each organization or role that has a URL policy assigned.

In Identity Server 6.0, policy implementation is not done using CoS templates. The policy definition itself contains the subjects like organization or roles. Using the policy CoS templates, the DSAME 5.1 policies must be converted to Identity Server 6.0 policies to contain the subjects in the policy definition itself.

To convert the DSAME 5.1 policies to Identity Server 6.0 policies, the script `update-polices.pl` is provided. This script can be run for each organization or top level organization. If only one organization is specified to the script, it generates one output file containing 5.1 policies converted to 6.0 format for that organization. If the top level organization is specified, one XML file for each organization that has policies under the top level organization is generated. The output file name is of the format `<rdn>-<rdn>.xml`. For example, if o=iplanet.com,o=isp has some policies, the output file is `o=iplanet.com-o=isp.xml`. Run `update-policies.pl` script to generate XML files for the organization that have policies in DSAME 5.1. Refer to the script for additional details.

DSAME 5.1 has domain URL service. This service lets you do policy delegation control. This is somewhat similar to referral policies in Identity Server 6.0. If there are any domain URL policies defined, they must be backed up manually in this migration step.

For each domain URL policy, a referral policy may need to be created in Identity Server 6.0. Based on the domain URL policies, corresponding referral policies must be created in Identity Server 6.0. This needs to be done manually. No script is provided for this purpose. Refer to the section "Updating Policies to Identity Server 6.0" on page 44 for more details.

This step must be run before migrating services. The output generated in this step will be used after the step "Updating Policies to Identity Server 6.0" on page 44.

The URL policy DTD for DSAME 5.1 is located under `DSAME_root/SUNWam/dtd`.

The URL policy DTD for Identity Server 6.0 is installed under `IS_root/SunWam/dtd`.

Migrating Authentication Entries

Authentication services have changed significantly in Identity Server 6.0. These include changes in attribute names, attribute values, and default values and removal of attributes. The authentication information is present for each organization. The migration must update authentication entries for all organizations.

Run the authentication migration script, `update-auth.pl`. The script generates an output file, `51to60auth-entries.ldif`. It also generates an input file `51auth-entries.dn`.

The output files generated in this step will be used in the step "Updating Authentication Entries to Identity Server 6.0" on page 43.

Refer to the section "Changes in Authentication Services" on page 49 for the details on changes in authentication services from DSAME 5.1 to Identity Server 6.0.

Migrating Services

Each of the DSAME services needs to be removed and the corresponding Identity Server 6.0 service loaded. In addition, all the new Identity Server 6.0 services must also be loaded. The services branch has global schema information for each of the services and may contain entries specific to an organization. If the top level DSAME entry is an organization itself, like `o=sun.com`, then the services branch under `sun.com` will contain global schema and organization specific entries as well. The organization-specific entries depend on what services have been registered for this organization. Follow this procedure to update the services branch.

1. If the top level entry is not an organization (such as o=isp, not an Identity Server organization), go to step 3. If the top level is an organization, go to step 2.

2. Run `update-toporg-services.pl` script. This script backs up organization entries for various authentication services and Identity Server Console service registered for the top level organization. The organization entries for the services reside under the global services entry for the top level organization. In order to update global services to 6.0, those services must be removed and loaded from 6.0. This step keeps a backup of the organization entries that reside under global services entry. Refer to this script for details. Check that it covers all the services which have organization specific entries (refer to step 3 as well).

   This script generates `51to60toporg-template.ldif` output file. It also generates, `51toporg-template.dn` input file.

3. Use Directory Server console to remove the DSAME services. If you added other services, don't remove them. The following services must be removed.

   `iPlanetAMAdminConsoleService`

   `iPlanetAMAuthService`

   `iPlanetAMAuthAnonymousService`

   `iPlanetAMAuthCertService`

   `iPlanetAMAuthLDAPService`

```
iPlanetAMAuthMembershipService
iPlanetAMAuthNTService
iPlanetAMAuthRadiusService
iPlanetAMAuthSafewordService
iPlanetAMAuthUnixService
iPlanetAMClientDetectionService
iPlanetAMDomainURLAccessService
iPlanetAMEntrySpecificService
iPlanetAMLoggingService
iPlanetAMNamingService
iPlanetAMPlatformService
iPlanetAMPolicyService
iPlanetAMSessionService
iPlanetAMUserService
iPlanetAMWebAgentService
DAI
```

If you have not added any additional services, you can remove the entire services branch under the top level entry.

4. Run `load-services.pl` to load Identity Server 6.0 services. This script loads all Identity Server 6.0 services. It uses the services XML `<install dir>/SUNWam/config/ums/ums.xml` and the XML files under `<install dir>/SUNWam/config/xml`.

5. Load the output file generated (`51to60toporg-template.ldif`) in step 2 above. This is required only if top level organization is an Identity Server organization. Use the `ldapmodify` command to load the output file. Refer to the last line in the output file for the syntax ("system" is not required, as that is used to run the script from the Perl script itself).

For details on Identity Server services changes in 6.0, refer to "Services in Identity Server 6.0" on page 54.

Updating Authentication Entries to Identity Server 6.0

Load the output file generated in the step "Migrating Authentication Entries." Use the `ldapmodify` command to load this. You can refer to the last line in the script for the syntax. This step migrates the DSAME 5.1 authentication entries to Identity Server 6.0. In addition, the following procedures may be necessary.

Customized 5.1 HTML templates, if any, need to be changed to 6.0 JSP-based templates.

Any customized authentication module need to be rewritten using `AMLoginModule.java`. Screen properties need to be changed to use the XML- based Authentication module properties. Refer to Identity Server 6.0 documentation for more details on writing custom authentication modules.

User's authentication module configuration will not be migrated automatically. If any authentication module is selected for any user in DSAME 5.1, it will not be available for that user after migration. The required authentication modules need to be configured manually for that user in Identity Server 6.0.

The user's default login URL attribute (iplanet-am-user-default-url) is no longer available in 6.0. This attribute is not migrated to 6.0 automatically. The value of this attribute can be set to iplanet-am-auth-login-success-url in the core authentication service or the planet-am-auth-login-success-url in the authentication configuration service or to a custom attribute depending on the deployment needs. This attribute must be migrated and removed from the user entries. Otherwise user entries that have this attribute can't be modified (you will get object class violation error).

Updating Identity Server Console Service Entries to 6.0

There are changes in the Identity Server 6.0 Console service that affect the Console display. The Domain URL Service is no longer available in 6.0. Because of the policy changes in 6.0, Web Agent service and Domain URL service need not be registered to organization, role and user entries. A script is provided to update entries to reflect these changes.

1. Run `update-services.pl` script to update the Console service if it is registered to any organization. The script generates input files, `51console.ldif` and `51services.ldif` and an output file, `60services.ldif`.

2. Run the `ldapmodify` command on the output file `60services.ldif`. This command migrates console service entries registered for the organization. It also migrates organization, roles and user entries for Domain URL and Web Agent services.

Updating Policies to Identity Server 6.0

Before loading the updated policies in the Directory Server, the DSAME 5.1 policies must be deleted. Run the `delete-policies` script to delete all policies. It generates an input file `delete-policies.dn` and an output file, `delete-policies.ldif`. Run `ldapdelete` command on `delete-policies.ldif` to delete all 5.1 policies. Make sure all the entries specified in `delete-policies.ldif` are deleted. If you get errors for the entries that don't exist in the Directory Server, remove those entries from the ldif file and continue deleting other entries of the file. The `delete-policies.ldif` file may have some duplicate entries. It may give errors while deleting already deleted entries (for duplicate entries). You can ignore such errors. You can run `ldapdelete` in continuous mode to ignore such errors.

Identity Server 6.0 has a new Policy Configuration service. This service specifies various configuration attributes used by policy components like subjects, referrals and conditions. In order to create policies in an organization, policy configuration service must be registered. For each organization, before loading the policies to that organization, this service must be registered. You can login to Identity Server 6.0 Console and register these services to each organization. You can also use the `amadmin` command to register this service to each organization.

Starting with the top level organization, run the following command to load Identity Server 6.0 policies:

IS_root`/bin/amadmin -u` *amadmin id* `-w` *password* `-t` *output migrated policy file for the organization*

In Identity Server 6.0, the policies can have a description along with the policy name. Also individual elements of policy such as rules, subjects, conditions and referrals have names. When importing DSAME 5.1 policies, the names for these elements and description are automatically generated. The names can be modified after the policies are imported.

Identity Server 6.0 has the concept of referral policies. Refer to the Identity Server 6.0 documentation for more details on this. In order to create policies in sub organizations, there must be referral policies from the top level organization. Referral policies do policy delegation based on the resource referrals.

Consider the following DIT:

```
        o=isp
         /\
o=siroe.com    o=iplanet.com
```

In order to create policies at `siroe.com` or `iplanet.com`, there must be a referral policy at `o=isp.com` to `siroe.com` and `iplanet.com`.

The referral policies at o=isp.com must contain the resource or resource prefix being managed at `o=siroe.com` or `o=iplanet.com`. If `siroe.com` manages `http://www.siroe.com/`, the referral policy at `o=isp.com` must contain the resource `http://www.siroe.com/` in its rule and it must refer to `siroe.com` organization. For other resources being managed at o=siroe.com, other referral policies must be created at o=isp. Only after creating the top level referral policies, the policies at the sub-organization must be updated by running the command specified above. If there are no referral policies at the parent level for the resource specified at the sub org level policies, policy creation fails. So it is important to create referral policies at the parent level to the sub organization level before running the above command. Refer to the policy output file for each sub organization and check the resources contained in the XML files. For each of those resources, a referral policy must exist at the top level before loading the policy output XML file for that sub organization.

DSAME 5.1 has domain URL service. This service lets you do policy delegation control. This is somewhat similar to referral policies in 6.0. The key difference is that the domain URL service is enforced during policy evaluation, referral policies are enforced during policy creation as well as policy evaluation. Only the top level administrator can create domain URL policies in DSAME 5.1 by default.

In DSAME 5.1, using the above DIT, one could create policies in `siroe.com` giving access to resources in `iplanet.com` and vice versa. However the top level admin at `o=isp`, can create domain URL policies at `o=siroe.com,o=isp`. This policy specifies what is allowed in this organization. If the domain URL policy says, allow `http://www.siroe.com/*`, only those resources allowed in URL policies that match `http://www.siroe.com/*` would be returned during policy evaluation. This is enforced using referral policies in 6.0. For each domain URL policy created in DSAME 5.1, a corresponding referral policy must be created in 6.0. This step must be done manually.

Identity Server 6.0 has policy administrator role for policy management. The policy administrator has privileges to create, delete or modify policies and to assign services to organizations. For each organization in 6.0, policy administrator role must be created. Run `update-policy-roles.pl` script. It generates an output file, `add-policy-roles.ldif`. It also generates an input file, `51org-entries.dn`. Load the output file generated by this script using `ldapmodify`. Refer to the syntax at the end of the script. This step creates various policy admin roles in Directory Server.

Post-Data Migration Tasks

(Optional) Enabling Federation Management

Identity Server 6.0 implements Liberty Alliance Phase 1 specifications. When you migrate services (see "“Migrating Services” on page 41"), two services are loaded for Federation Management. They are `iPlanetAMAuthenticationDomainConfigService` and `iPlanetAMProviderConfigService`. These services must be registered before you can use the Federation management features. An ldif file named `liberty-services.ldif` is provided to register these services.

Substitute the value of ROOT_SUFFIX in this file with the top-level organization. Run `ldamodify` on this ldif file. If the entries specified in this file are present in Identity Server 6.0, remove those entries from this ldif file and load the remaining entries. After this step, Federation management is enabled in Identity Server 6.0.

Miscellaneous 5.1-to-6.0 Modifications

Once the services and authentication entries are migrated, users should be able to successfully log in to Identity Server 6.0. If the Directory Server is on the same machine as Identity Server 6.0, edit `<installdir>/bin/amserver` script and modify the `NDS_SERVER` variable to point to the correct Directory Server instance.

Restart the Identity Server and login to the Identity Server 6.0 Console. If the default login URL of 5.1 in the core authentication service is not modified (`<protocol>://<host>:<port>/amserver/login`), you can use default login URL of Identity Server 6.0 (`<protocol>://<host>:<port>/amserver/UI/Login`) to login to Identity Server 6.0 Console. There is a known issue in 5.1, where the default login url is set to `/amserver/login` sometimes instead of `<protocol>://<host>/amserver/login` in core authentication service. In such cases, you can't use 6.0 default login URL to login. You need to modify the associated domain attribute of the default org to the 6.0 default login URL(`<protocol>://<host>/amserver/UI/Login`) to access the console using 6.0 default login URL. Use the fully qualified domain name for host and use the correct deployment descriptor in the URL. Note that the associated domain attribute value does not have port number in it but while accessing the console, need to specify the port. You can also use the URL of the form `<protocol>://<host>:<port>/amserver/UI/Login?org=<org RDN>` . It is a good idea to check for login, before migrating other entries. The migration is a step-by-step process; the steps should be validated as and when possible.

User management is not enabled by default. After the login to Identity Server Console, go to Service Configuration tab. Edit Administration service. Click on "Enable User Management" check box and save the service. Now you can go to Identity Management tab for user management functions.

IS 6.0 has introduced a new user, `amldapuser`. This user is used to bind and search the directory for LDAP, Membership authentication modules. This user is also used in the policy config service. Once the LDAP, Membership or Policy Config Service is registered to an organization, password for this user must be explicitly entered in those services. The password is the `amldapuser` password entered during Identity Server 6.0 installation. In addition, this user must also be created. Run the following two commands to create this user and to set access rights to this user.

```
<path to ldapmodify>/ldapmodify -D "cn=directory manager" -w <password>
dn: cn=amldapuser,ou=DSAME Users,ROOT_SUFFIX
changetype: add
objectclass: inetuser
objectclass: organizationalperson
objectclass: person
objectclass: top
cn: amldapuser
sn: amldapuser
userPassword: <password>
<path to ldapmodify>/ldapmodify -D "cn=directory manager" -w <password>
dn: ROOT_SUFFIX
changetype: modify
add: aci
aci: (target="ldap:///ROOT_SUFFIX")(targetattr="*")(version 3.0; acl
"special ldap auth user rights"; allow (read,search) userdn =
"ldap:///cn=amldapuser,ou=DSAME Users,ROOT_SUFFIX";)
```

In the above commands, specify the Directory Manager password for the -w option. Replace `ROOT_SUFFIX` with the your install root entry. Specify the `amldapuser` password for the `userPassword` attribute in the first command.

Note that if the LDAP and Membership services are already registered to an organization in IS 5.1, the bind DN used for the user search is `"dsameuser"`. Change that user to the `"amldapuser"` created above and the password to the `amldapuser` password. This should be done in all the organizations where these two services are registered. If they are kept as `"dsameuser"`, it will continue to work, but for Identity Server 6.0, it is recommended that you use `"amldapuser"`.

The Identity Server user's profile has an account expiration date attribute. The account expiration date format in DSAME 5.1 is mm/dd/yy hh:mm, while the account expiration date format in 6.0 is mm/dd/yyyy hh:mm. The attribute format is present in the file `amUser.properties` present in the directory named locale. If the account expiration attribute is set for the users in DSAME 5.1, the format of the date should be changed to 6.0 format, when modifying the users profile from Identity Server Console, to save the users profile changes. Alternatively, the date format can be changed to use the DSAME 5.1 format in `amUser.properties`. The ldapmodify command can also be used to modify the account expiration attribute value.

If there are any specific changes made to `AMConfig.properties` in DSAME 5.1, those changes must also be made in `AMConfig.properties` after Identity Server 6.0 is installed.

Migrating Console Changes

If any console customization has been done in DSAME 5.1, those changes must be migrated to console files in Identity Server 6.0.

This step must be done manually. No scripts are provided for this purpose.

The above steps migrate Directory Server Data, Schema and any customized data to 6.0. Once the steps are complete, you should restart Identity Server 6.0.

Migrating Agents

Agents 1.0 or 1.1 do not work with Identity Server 6.0. You must have Agents 2.0 to work with Identity Server 6.0. In order to migrate agents, you must uninstall Agents 1.0 or 1.1 and then install Agents 2.0.

1. Backup any configuration changes made in 1.0 or 1.1 agents. For example changes done to AMAgent.properties.
2. Install 2.0 agents.
3. Make changes to 2.0 configuration files.

4. Restart the agents.

NOTE

1. Any changes done to DSAME 5.1 XML files are not migrated in this procedure. Once Identity Server 6.0 is installed, those changes need to be manually updated in the 6.0 XML files. One way to do this is to update the 6.0 XML files before loading them.
2. The script update-rootsuffix.pl is not used in the migration procedure. This script can be used in step 3.4, if this script is available from another Identity Server 6.0 install. This script updates the top level entry to have iplanet-am-service-staus attribute.

Changes in Authentication Services

This section describes in detail the changes in authentication services.

Authentication Service (Core) [amAuth.xml]

Attribute Changes

Global

1. Removed "iplanet-am-auth-login-worker-classes"
2. Added "iplanet-am-auth-sleep-interval"

Organization

1. Removed "iplanet-am-auth-chaining-modules"
2. Removed "iplanet-am-auth-chaining-enabled"
3. Removed "iplanet-am-auth-non-interactive-modules"
4. Removed "iplanet-am-auth-default-url"
5. Removed "iplanet-am-auth-user-based"
6. Removed "iplanet-am-auth-login-worker-class"
7. Added "iplanet-am-auth-org-config"
8. Added "iplanet-am-auth-login-success-url"
9. Added "iplanet-am-auth-login-failure-url"

10. Added "iplanet-am-auth-post-login-process-class
11. Added "iplanet-am-auth-username-generator-enabled"
12. Added "iplanet-am-auth-username-generator-class
13. Changed "iplanet-am-auth-menu" to "iplanet-am-auth-allowed-modules"
14. In "iplanet-am-auth-admin-auth-module",
    - Changed ’type’ from "single_choice" to "single"
    - Changed ’syntax’ from "string" to "xml"
    - Added attribute ’propertiesViewBeanURL’ set to "/amconsole/auth/ACModuleList"
    - Added attribute ’uitype’ set to "link"
    - Removed sub-element ChoiceValues
    - Changed Default Value from plain string to xml string
15. In "iplanet-am-auth-login-failure-count", Changed Default Value from 3 to 5
16. In "iplanet-am-auth-login-failure-duration", Changed Default Value from 15 to 300
17. In "iplanet-am-auth-lockout-warn-user", Changed Default Value from 1 to 4
18. In "iplanet-am-auth-default-auth-level", Changed ’syntax’ from "string" to "number"

Authentication related attribute changes in User Service [amUser.xml]

Attribute Changes

Dynamic

1. Removed "iplanet-am-user-auth-modules"

User

1. Removed "iplanet-am-user-auth-modules"
2. Removed "iplanet-am-user-default-url"
3. Added "iplanet-am-user-auth-config"

4. Added "iplanet-am-user-alias-list"
5. Added "iplanet-am-user-success-url"
6. Added "iplanet-am-user-failure-url"
7. In "iplanet-am-user-account-life", changed ’syntax’ from "date" to "string"

All "Organization" attributes in the following xml files

- amAuthLDAP.xml
    - In "iplanet-am-auth-ldap-search-filter", changed ’syntax’ from "string" to "xml"
    - In "iplanet-am-auth-ldap-auth-level", changed ’syntax’ from "string" to "number"
- amAuthAnonymous.xml
    - In "iplanet-am-auth-anonymous-auth-level", changed ’syntax’ from "string" to "number"
- amAuthMembership.xml
    - In "iplanet-am-auth-membership-search-filter", changed ’syntax’ from "string" to "xml"
    - In "iplanet-am-auth-membership-auth-level", changed ’syntax’ from "string" to "number"
- amAuthRadius.xml
    - In "iplanet-am-auth-radius-auth-level", changed ’syntax’ from "string" to "number"
- amAuthUnix.xml
    - In "iplanet-am-auth-unix-auth-level", changed ’syntax’ from "string" to "number"
- amAuthCert.xml
    - In "iplanet-am-auth-cert-auth-level", changed ’syntax’ from "string" to "number"
- amAuthSafeWord.xml
    - In "iplanet-am-auth-safeword-auth-level", changed ’syntax’ from "string" to "number"

Table 2-3 summarizes the UI changes for authentication interface. The left column lists the filenames used in DSAME 5.1.1, the center column provides a brief description of each file, and the right column lists the filenames used in Identity Server 6.0.

Table 2-3 GUI Changes in the Authentication Interface

	DSAME 5.1.1 Filename	Description	IS 6.0 Filename
1.	account_expired.html	Your account has expired. Contact your system administrator.	account_expired.jsp
2.	configuration.html	Configuration error.	configuration.jsp
3.	disclaimer.html	This is a sample disclaimer template.	disclaimer.jsp
4.	invalidPCookieUserid.html	Persistent Cookie Username does not exist in the Persistent Cookie Domain.	invalidPCookieUserid.jsp
5.	invalidPassword.html	The password entered does not contain enough characters.	invalidPassword.jsp
6.	invalid_domain.html	No such domain.	invalid_domain.jsp
7.	login_denied.html	User has no profile in this organization.	login_denied.jsp
8.	login_fail_template.html	Authentication failed.	login_failed_template.jsp
9.	login_menu.html	Authentication Menu the tag rows will be replaced with login_menu_modules.html.	removed
10.	login_menu_modules.html	Authentication Menu will loop and display (replace the tag inside) this file with all the available modules	removed
11.	login_prompt.html	User based login page.	Login.jsp
12.	login_success.html	You have logged in successfully, but your system has no default login page.	Login.jsp
13.	login_template.html	Login/Password page	Login.jsp
14.	login_timeout_template.html	Your login session has timed out.	session_timeout.jsp
15.	logout.html	You have logged out.	Logout.jsp

Table 2-3 GUI Changes in the Authentication Interface

	DSAME 5.1.1 Filename	Description	IS 6.0 Filename
16.	max_sessions.html	Maximum Sessions Limit Reached.	Message.jsp
17.	membership.html	Self Registration Module	membership.jsp
18.	membershipSkeleton.html		removed
19.	missingReqField.html	One of the required fields was not completed.	missingReqField.jsp
20.	module_denied.html	Your authentication module is denied.	module_denied.jsp
21.	noConfirmation.html	Missing the confirmation password field.	noConfirmation.jsp
22.	noLoginWorker.html	Authentication Page Generator not found.	removed
23.	noPassword.html	There was no password entered.	noPassword.jsp
24.	noUserName.html	There was no user name entered.	noUserName.jsp
25.	noUserProfile.html	No user profile was found matching the user name.	noUserProfile.jsp
26.	org_inactive.html	This organization is not active.	org_inactive.jsp
27.	passwordMismatch.html	The password and the confirm password do not match.	passwordMismatch.jsp
28.	privilege_failure.html	User does not have access to this operation.	Message.jsp
29.	profileException.html	An error occurred while storing the user profile.	profileException.jsp
30.	radius_patch.html	RADIUS authentication requires i-Planet patch 1.	Message.jsp
31.	register.html	Self Registration	register.jsp
32.	session_invalid.html	Your session is invalid.	removed
33.	session_timeout.html	Your session is timed out.	session_timeout.jsp
34.	userExists.html	A user already exists with this name.	userExists.jsp

Table 2-3 GUI Changes in the Authentication Interface

	DSAME 5.1.1 Filename	Description	IS 6.0 Filename
35.	`userPasswordSame.html`	The User Name and Password fields cannot have the same value.	`userPasswordSame.jsp`
36.	`user_inactive.html`	This user is not active.	`user_inactive.jsp`
37.	`wrongPassword.html`	The password entered is invalid.	`wrongPassword.jsp`
38.	add	Displays internal authentication framework errors.	`auth_error_template.jsp`
39.	add	No configuration found/defined for a user for an organization.	`noConfig.jsp`
40.	add	User is not in a Role. (for ’role’ based authentication.)	`userDenied.jsp`

Services in Identity Server 6.0

The following are the new services in Identity Server 6.0:

- SAML service (amSAML.xml)
- Security Service (amDSS.xml)
- Policy Config Service (amPolicyConfig.xml)
- Auth Config service (amAuthConfig.xml)

The following services have been removed in 6.0:

- Domain URL service (amDomainURLAccess.xml)

The following services remain quite similar in DSAME 5.1 and Identity Server 6.0:

- Identity Server Console Service (amAdminConsole.xml)
- Auth Anonymous Service (amAuthAnonymous.xml)
- Auth Membership Service (amAuthMembership.xml)
- Auth Cert Service (amAuthCert.xml)
- Auth LDAP Service (amAuthLDAP.xml)
- Auth NT Service (amAuthNT.xml)
- Auth Radius Service (amAuthRadius.xml)

- Auth SafeWord Service (amAuthSafeWord.xml)
- Auth Unix Service (amAuthUnix.xml)
- Client Detection Service (amClientDetection.xml)
- Naming Service (amNaming.xml)
- Platform Service (amPlatform.xml)
- Session Service (amSession.xml)
- URL Agent Service (amWebAgent.xml)
- Entry Specific Service (amEntrySpecific.xml)
- User (amUser.xml)

The following services have changed significantly in 6.0:

- Auth Service (amAuth.xml)
- Logging Service (amLogging.xml)
- Policy Service (amPolicy.xml)
- User Service (amUser.xml)

Name Changes to Attributes and Object Classes

Table 2-4 lists the names of attributes that have been renamed in Identity Server 6, and provides the new name for each attribute.

Table 2-4 Renamed Attributes

Old Attribute Name	New Attribute Name
iplanetserviceschema	sunserviceschema
iplanetserviceid	sunserviceid
iplanetsmspriority	sunservicepriority
iplanetpluginschema	sunpluginschema
iplanetkeyvalue	sunkeyvalue
iplanetpluginid	sunpluginid
iplanetxmlkeyvalue	sunxmlkeyvalue
iplanet-am-domain-name	sunPrefferedDomain

Table 2-5 lists the names of object classes that have been renamed in Identity Server 6.0, and provides the new name for each object class.

Table 2-5 Renamed Object Classes

Old Object Class	New Object Class
`iplanetservice`	`sunservice`
`iplanetservicecomponent`	`sunservicecomponent`
`iplanetorgservice`	`sunorgservice`
`iplanetserviceplugin`	`sunserviceplugin`
`iplanet-am-managed-org`	`sunManagedOrganization`

In addition, **iplanet-am-unique-attribute-list** and `iplanet-am-attribute-uniqueness-enabled` attributes are removed from Identity Server Console Service. A new attribute `sunNameSapceUniqueAttrs` in the new object class `sunNameSpace` is added to the organization entries to accommodate unique attribute list removed from Identity Server Console Service.

Chapter 3

Configuring Identity Server with a Provisioned Directory

This chapter provides instructions for installing Sun ONE Identity Server against a directory tree that is already provisioned with user data. It also explains how to configure Identity Server to work with your directory information tree (DIT), and how to make the necessary changes to your existing Sun ONE Directory Server and directory entries. The number and scope of changes you must make will depend upon how your directory tree is structured, and how you plan to use Identity Server.

Topics in this chapter include:

Overview of Installation & Configuration Tasks

There are a number of steps you must take before you will see your existing directory data in Identity Server. This chapter groups steps into two distinct phases: installation and post-installation configuration.

Installing and Starting Identity Server 6.1

1. Install Identity Server 6.1.

   In this step you run the Java Enterprise System installer. For detailed installation instructions, see the *Java Enterprise System Installation Guide*.

2. Start Identity Server.

   For detailed instructions, see "Starting Identity Server and Logging In" on page 62 in this chapter.

Post-Installation Configuration

Detailed instructions for each of the following steps are provided in this chapter.

1. Manually Configure Directory Server to work with Identity Server.

   In this step you enable the Directory Server referential integrity plug-in, and create new database indexes. See "Configuring the Directory Server" on page 66 in this chapter.

2. Enable User Management services.

   a. Add Marker Object Classes to Existing Directory Entries.

      Before Identity Server can recognize the data in an existing directory, you must add special object classes to entries for all organizations, groups and users that will be managed by Identity Server. Sample scripts are bundled in the product to help you automatically add these object classes to your directory. Detailed steps and examples are provided in this chapter. See "Adding Identity Server Object Classes to Existing Directory Entries" on page 71.

b. Add Custom Object Classes to Identity Server Schema.

If your existing directory tree contains custom object classes, Identity Server won't recognize them until you manually configure both Identity Server and Directory Server. The types of changes you need to make are illustrated in this chapter using the directory tree for MadisonParc, a fictitious company. Detailed steps are in "Adding Custom Object Classes to Identity Server Schema" on page 89.

c. Load Identity Server LDIF into your directory.

In this step, you commit all of your LDIF modifications to make actual changes in the Directory Server. Detailed instructions are in "Loading Identity Server LDIF into Your Directory" on page 101.

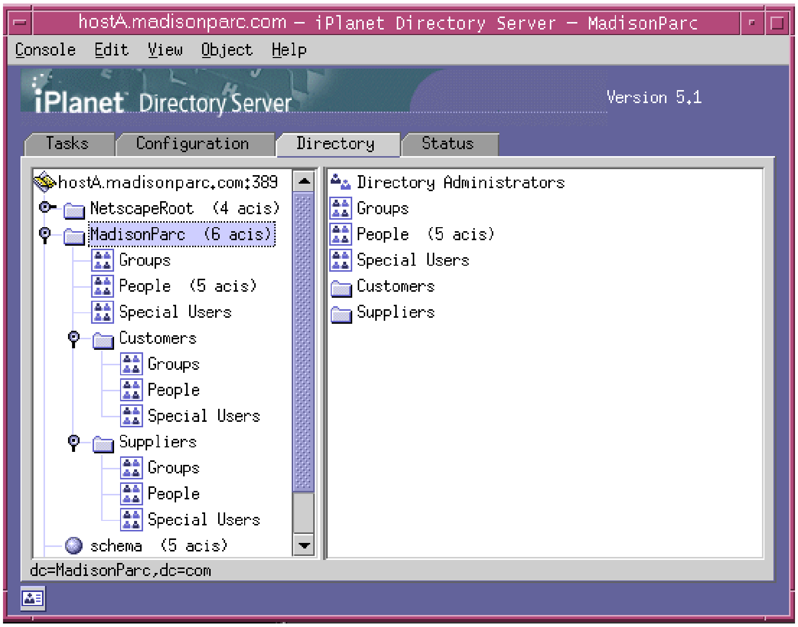
d. In the Identity Server console, enable User Management.

This is the last step in the instructions for "Enabling User Management Services" on page 69.

MadisonParc Examples Used in This Chapter

The MadisonParc examples in this chapter help to illustrate the kinds of manual modifications you must make in both Directory Server and Identity Server. Figure 3-1 illustrates the Directory Server console view of the directory tree for MadisonParc. The tree includes three organizational units (ou) at the top level of the tree: Groups, People, and Special Users. These organizational units contain entries for MadisonParc employees. Two organizations (dc), Customers and Suppliers, were created under the root level to contain entries for non-employees.

Figure 3-1 The existing directory tree for MadisonParc and Identity Server 6.0.



In the MadisonParc example, there are two custom object classes and three custom attributes. These object classes and attributes are not included in the Identity Server schema nor in the Directory Server 5.1 SP1 schema.Table 3-1 summarizes the custom objects and their uses in the MadisonParc directory tree.

Table 3-1 User-defined objects used in the MadisonParc directory tree.

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 Identity Server to manage these extensions, the following modifications would have to be made in the Identity Server schema:

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

If your existing directory tree contains custom object classes or attributes, you'll need to make similar changes in your directory and in your Identity Server XML files.

Detailed steps and examples are provided in this chapter. See "Adding Custom Object Classes to Identity Server Schema" on page 89.

Before You Begin

You resolve the following Directory Server issues before you can install Identity Server against an existing Directory Server that is provisioned with users. Refer to the documentation for Sun ONE Directory Server for detailed information about the following issues.

Migrating Pre-5.2 Versions of Directory Server

If you are using a pre-5.2 version of Directory Server, you can upgrade your existing Directory Server to version 5.2, and then migrate your existing data to the upgraded directory. For detailed instructions, see the *Sun ONE Directory Server Installation and Tuning Guide* at the following URL:

```
http://docs.sun.com/source/816-6697-10/
```

Backing Up Directory Server

The installation and post-installation tasks described in this chapter require numerous changes to your existing directory. Be sure to back up the Directory Server before you install running the Identity Server installation program. To back up Directory Server, use `db2ldif` or `db2bak`,or use the Directory Server console. For detailed information on backing up your directory, see the *Sun ONE Directory Server Administration Guide* at the following URL:

```
http://docs.sun.com/doc/816-6698-10
```

Directory Server Access

Before you run the Identity Server Installation program, be sure that Directory Server is running, and that Identity Server can access it. You must also have appropriate administrator privileges in the Directory Server to modify user entries.

Installing Identity Server User and Policy Management Services

See the *Java Enterprise System Installation Guide* for detailed instructions on installing Identity Server User and Policy Management Services. The Java Enterprise System installer will guide you to provide information about your existing provisioned directory. Once the installation program is finished, you can start Identity Server, and log in to its administration console.

Starting Identity Server and Logging In

After you've installed Identity Server and configured Directory Server appropriately, you can check the installation. Start Identity Server and log in to the Identity Server Console as the user `amAdmin`. Once you successfully log in, you'll see the Identity Server web interface.

To Start Identity Server

1. Start Directory Server.

   Execute the start or restart command in the directory where the Directory Server instance is installed. Examples:

UNIX	*DirectoryServer_base*/slapd-*hostName*/slapd-start *DirectoryServer_base*/slapd-hostName/slapd-restart
Windows	C: *DirectoryServer_base*\slapd-*hostName*\slapd-start C: *DirectoryServer_base*\slapd-hostName\slapd-restart

2. Start the web container that runs Identity Server.

   If the web container is Sun ONE Web Server, then execute the start command in directory were the Web Server instance is installed. Examples:

UNIX	*WebServer_base*/https-*hostName*/start
Windows	C: *WebServer_base*\https-*hostName*\start

   If the web container is Sun ONE Application server, then execute the start command in the directory where the Application Server instance is installed. Examples, where *server1* is the instance name by default:

UNIX	*ApplicationServer_base*/bin/asadmin start-*server1* --user *admin_user*--password *your_admin_passwd server1*
Windows	C: *ApplicationServer_base*\bin\asadmin start-*server1* --user *admin_user*--password *your_admin_passwd server1*

   If the web container is BEA WebLogic or IBM WebSphere, then see the instructions for starting the server in the documentation that comes with the product.

To Log into the Identity Server Console

1. Go to the login URL using the form:

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

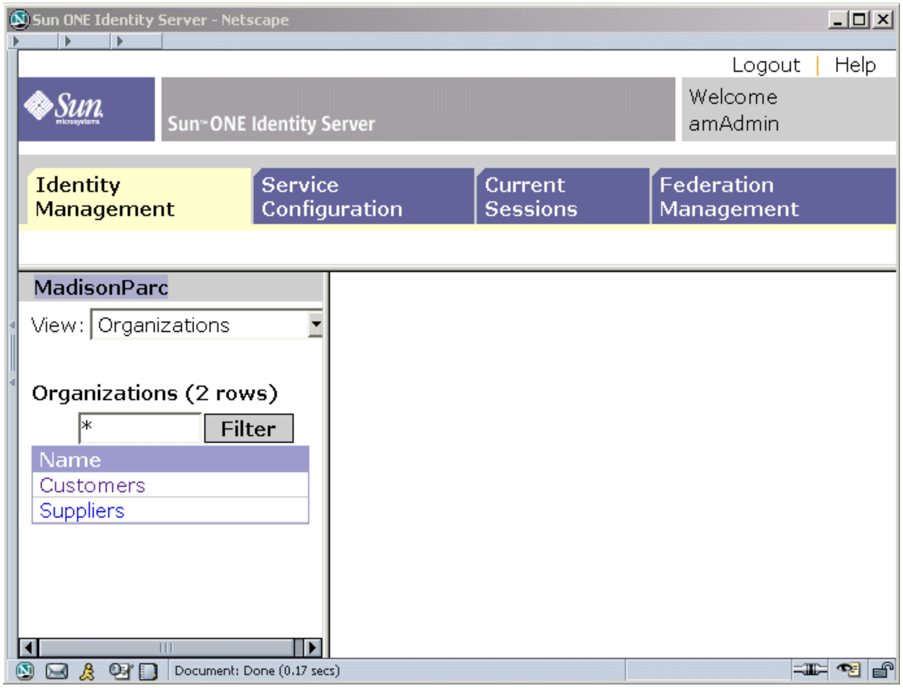
   where *host* is the host name of the system, *domain* is the domain name of the server that runs Identity Server services, and *port* is the Identity Server services port number.

   Example: `http://ginac.sun.com:58080/amserver/UI/Login`

2. In the Login page, enter the Top-Level Administrator user name `amAdmin`, and then enter the password you specified at installation.

The Policy Management services are automatically installed against your existing provisioned directory. When you see the Identity Server interface (see Figure 3-2), you can immediately begin creating policies. See the Identity Server Administration Guide for more information. You will see the root suffix and organizations you specified during installation. For the MadisonParc example used at the beginning of this chapter, you would see the root suffix MadisonParc, and the two organizations `Customers` and `Suppliers`.

Figure 3-2 First-time login for MadisonParc.



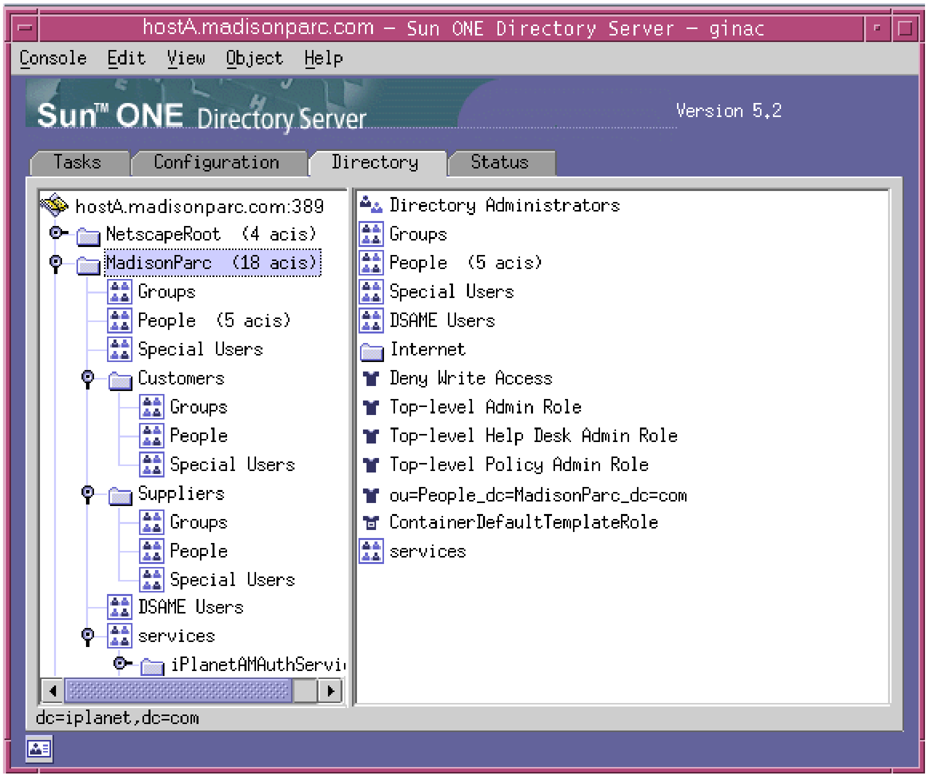
.

NOTE You will not see directory entries below the organization level until you complete the post-configuration tasks described in the following sections:

- "Configuring the Directory Server" on page 66
- "Enabling User Management Services" on page 69

Also immediately after Installation, if you look in the Directory Server console view, you'll see the Identity Server object classes, roles, users, and services Figure 3-3. illustrates the Identity Server objects and existing directory tree for the MadisonParc examples used in this chapter.

Figure 3-3 Policy Management services installed against the existing MadisonParc directory tree.



Configuring the Directory Server

After you've installed the Identity Server schema, you must configure the Directory Server to work with Identity Server. Perform the steps in the following procedures:

- Enable the Directory Server referential integrity plug-in
- Add Identity Server indexes

When the referential integrity plug-in is enabled, it performs integrity updates on specified attributes immediately after a delete or rename operation. This ensures that relationships between related entries are maintained throughout the database. Database indexes enhance the search performance in Directory Server.

NOTE Before continuing with these procedures, be sure that Directory Server is running.

To Enable the Referential Integrity Plug-In

1. In Directory Server console, click Configuration.
2. In the navigation tree, double-click Plug-ins to expand the list of Plug-ins.
3. In the Plug-ins list, click “referential integrity postoperation.”
4. In the properties area, check the “Enable plug-in” box.
5. Click Save.

The plug-in is not enabled until you restart Directory Server.

To Add Identity Server Indexes

1. In Directory Server console, click Configuration.
2. Add the `nsroledn` index.
    - e. In the navigation tree, double-click the Data icon, and then click the root suffix that contains the directory entries you want to use in Identity Server.
    - f. Click the Indexes tab.
    - g. Under “Additional Indexes,” for the `nsroledn` attribute, check the following checkboxes: Equality, Presence, and Substring.
    - h. Click Save.
    - i. In the “Indexes” window, after the index is successfully created, click Close.
3. Add the `memberof` index.
    - a. In the Indexes tab, click "Add attribute..."
    - b. In the “Select Attributes” window, select the attribute `memberof`, and then click OK.
    - c. In the Indexes tab, for the `memberof` attribute, check the following checkboxes: Equality and Presence.
    - d. Click Save.

   e. In the “Indexes” window, after the index is successfully created, click Close.

4. Add the `iplanet-am-static-group` index.

   a. In the Indexes tab, click "Add attribute..."

   b. In the “Select Attributes” window, select the attribute `iplanet-am-static-group`, and then click OK.

   c. In the Indexes tab, for the `iplanet-am-static-group` attribute, check the following checkbox: Equality.

   d. Click Save.

   e. In the “Indexes” window, after the index is successfully created, click Close.

5. Add the `iplanet-am-modifiable-by` index.

   a. In the Indexes tab, click "Add attribute..."

   b. In the “Select Attributes” window, select the attribute `iplanet-am-modifiable-by`, and then click OK.

   c. In the Indexes tab, for the `iplanet-am-modifiable-by` attribute, check the following checkbox: Equality.

   d. Click Save.

   e. In the “Indexes” window, after the index is successfully created, click Close.

6. Add the `iplanet-am-user-federation-info-key` index.

   a. In the Indexes tab, click "Add attribute..."

   b. In the “Select Attributes” window, select the attribute `iplanet-am-user-federation-info-key`, and then click OK.

   c. In the Indexes tab, for the `iplanet-am-user-federation-info-key` attribute, check the following checkbox: Equality.

   d. Click Save.

   e. In the “Indexes” window, after the index is successfully created, click Close.

7. Restart Directory Server.

Enabling User Management Services

This section provides an overview of the configuration tasks you'll need to perform in order to see your existing user data displayed in the Identity Server interface. If you want to enable user management before proceeding with the configuration tasks, first log in to Identity Server, and then skip to step Step 8 this section. Note that if you skip to step Step 8 without performing the necessary configuration, although you will be able to create and manage new user entries, you will not be able to access your existing data through Identity Server.

To enable User Management Services

1. Add Identity Server object classes and attributes to your existing DIT.

   For detailed instructions, see "Adding Identity Server Object Classes to Existing Directory Entries" on page 71 in this chapter.

2. Modify the Identity Server schema.

   See "Adding Custom Object Classes to Identity Server Schema" on page 89 in this chapter for more information. The section provides detailed instructions on performing the following steps:

   a. Modifying the Creation Templates

   b. Adding Attributes to the Organization Schema

   c. Adding Attributes to the User Schema

3. To load all XML files, enter the following command:

   Identity_Server_root`/config/ums/amserveradmin`
   `"`*user_naming_attibute*`=amadmin,ou=people,`*root_suffix*`"` *password*

4. Load `installExisting.ldif` file into your Directory Server.

   For detailed instructions, see "Loading Identity Server LDIF into Your Directory" on page 101 in this chapter.

5. Load `install.ldif` file into your Directory Server.

   For detailed instructions, see "install.ldif" on page 103 in this chapter.

6. Restart Identity Server. To start Identity Server services, you must start both Directory Server and the web container that runs Identity Server.

   a. Start Directory Server.

      Execute the `start` or `restart` command in the directory where the Directory Server instance is installed. Examples:

UNIX	*DirectoryServer_base*`/slapd-`*instanceName*`/slapd-start`
Windows	`C:` *DirectoryServer_base*`\slapd-`*instanceName*`\slapd-start.bat`

b. Start the web container that runs Identity Server.

- If the web container is Sun ONE Web Server, then execute the `start` command in directory were the Web Server instance is installed. Examples:

UNIX	*WebServer_base*`/https-`*instanceName*`/start`
Windows	`C:` *WebServer_base*`\https-`*instanceName*`\start`

- If the web container is Sun ONE Application server, then execute the `start` command in the directory where the Application Server instance is installed. Examples, where *server1* is the instance name by default:

UNIX	*ApplicationServer_base*`/bin/asadmin start-`*server1* `--user` *admin_user*`--password` *your_admin_passwd server1*
Windows	`C:` *ApplicationServer_base*`\bin\asadmin start-`*server1* `--user` *admin_user*`--password` *your_admin_passwd server1*

- If the web container is BEA WebLogic or IBM WebSphere, then see the instructions for starting the server in the documentation that comes with the product.

7. Log in to Identity Server console as `amAdmin`.

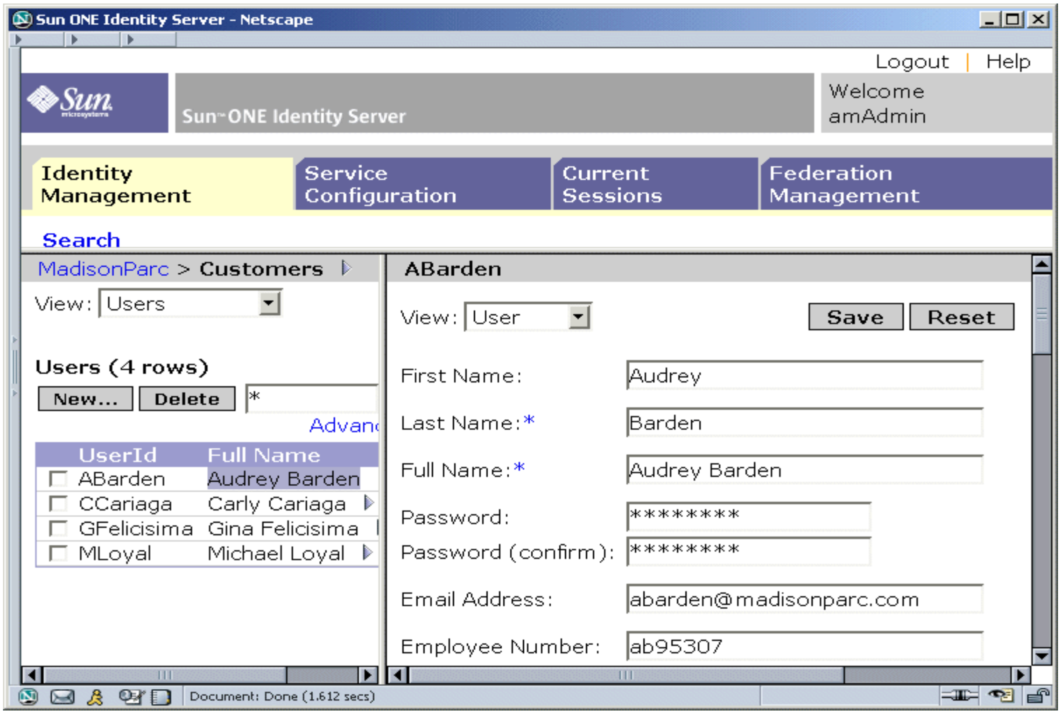
You will not see your existing groups and users from your existing directory tree until you complete the following two steps.

8. In the Identity Server console, click Service Management > Administration.

9. In the “Administration“ window, click the "Enable User Management" box, and then click Save.

Once you’ve checked the “Enable User Management” box, you should see all entries beneath the organization level in Identity Server. For instructions on using Identity Server to create or manage users and policies, see the *Administration Guide*.

Figure 3-4 Identity Server with data from existing MadisonParc directory tree



View: Users

	UserId	Full Name
□	ABarden	Audrey Barden
□	CCariaga	Carly Cariaga
□	GFelicisima	Gina Felicisima
□	MLoyal	Michael Loyal

View: User

First Name: Audrey

Last Name:* Barden

Full Name:* Audrey Barden

Password: ********

Password (confirm): ********

Email Address: abarden@madisonparc.com

Employee Number: ab95307

Adding Identity Server Object Classes to Existing Directory Entries

After you've installed configured Identity Server, you must modify your existing directory entries to include the necessary Identity Server object classes and attributes. You can think of the Identity Server object classes as *markers* that indicate the directory entries you want to manage through Identity Server. These markers enable Identity Server to recognize the entries in your directory. The object classes contain special attributes that are necessary to achieve delegated administration.

Before You Begin

There are a number of resources you can use to facilitate the remaining steps for using an existing directory.

Examples Used in This Section

The examples used in this chapter are based on the directory tree for a fictitious company named MadisonParc. Figure 3-5 shows two organizations, `Customers` and `Suppliers`, under the root.

Utilities and Scripts You Can Use

You can make these modifications by using Sun ONE Directory Server Console, or by using the `ldapmodify` or `db2ldif` utilities that come with Directory Server. You can also use the sample scripts that come with Identity Server.

Directory Server Utilities

Make sure that you're using the appropriate version of `ldapmodify`. Set your path to use the `ldapmodify` command that is shipped with Sun ONE Directory Server. (Do not use the version shipped with Solaris, which is found in `/bin` or `/usr/bin`.) Follow these procedures:

- On Solaris, add *IdentityServer_base*`/SUNWam/ldaplib/solaris/sparc/ldapsdk` to your `LD_LIBRARY_PATH` to pick up the appropriate Directory Server libraries. At the command line, enter:

  ```
  which ldapmodify
  ```

  The following should be displayed:

  IdentityServer_base`/SUNWam/bin/ldapmodify`

- If you are on Windows, you'll find ldapmodify in this directory of the Identity Server installation:

  IdentityServer_base`\tools`

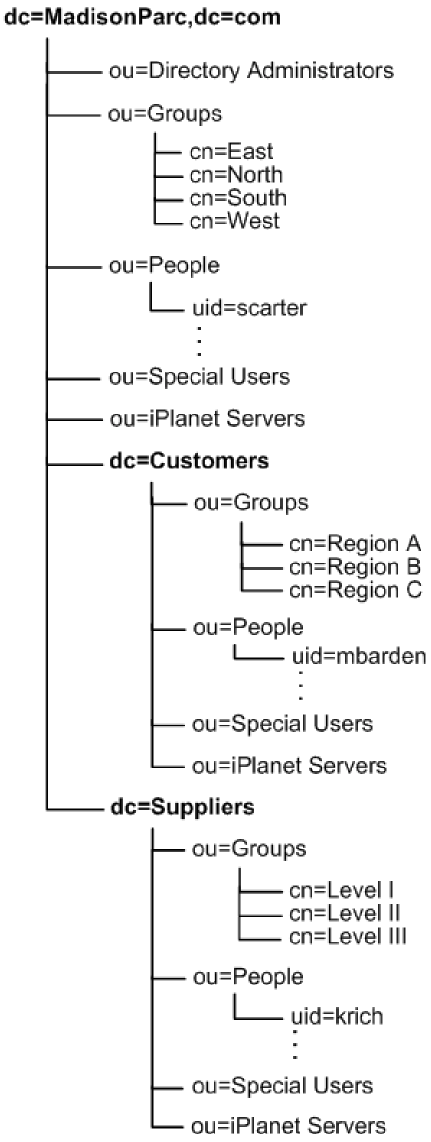
  Open a DOS prompt window and set the path to the ldapmodify tool.

  Example:

  `set PATH=`*IdentityServer_base*`\tools;%PATH%`

  For detailed information on how to make changes to the directory using these utilities or by using SunONE Console, see the documentation for Sun ONE Directory Server `http://docs.sun.com/prod/s1dirsrv`.

Figure 3-5 The existing MadisonParc directory tree.



Sample Migration Scripts

The sample scripts included with Identity Server require Perl 5.x or later. Table Table 3-2 provides descriptions of what each script adds to existing directory entries. You'll find the sample scripts in the following location:

UNIX	*IdentityServer_base*`/SUNWam/migration`
Windows	*IdentityServer_base*`\migration`

Table 3-2 Descriptions of scripts for adding Identity Server marker object classes.

Script	What it Does
`update-o.pl`	Adds the following to each organization entry: • `sunManagedOrganization` • `sunISManagedOrganization` • `sunNameSpace` • `inetDomain` • `inetDomainStatus`
`update-people.pl`	Adds `iplanet-am-managed-people-container` to each people container.
`update-ou.pl`	Adds `iplanet-am-managed-org-unit` to each organizational unit.
`update-users.pl`	Adds the following to each user entry: • `inetadmin` • `iplanet-am-managed-person` • `iplanet-am-user-service` • `inetuser` • `iPlanetPreferences` • `inetOrgPerson`
`udpate-static-groups.pl`	Adds the following to each static group: • `iplanet-am-managed-static-group` • `iplanet-am-managed-group`
`update-filtered-groups`	Adds the following to each dynamic, or *filtered*, group: • `iplanet-am-managed-group` • `iplanet-am-managed-filtered-group`

Table 3-2 Descriptions of scripts for adding Identity Server marker object classes.

Script	What it Does
`update-assignable-dynamic-groups`	Adds the following to each assignable dynamic group: • `iplanet-am-managed-group` • `iplanet-am-managed-assignable group`
`update-groups.pl`	Adds `iplanet-am-managed-group-container` to each organizational unit that contains groups.

While these samples should prove useful, keep in mind that they are only tools to assist you in properly formatting the directory tree and other data. Each script generates an LDIF file that you can inspect before making actual changes in your directory. You run each script a second time with the last line uncommented to make the actual changes. Steps for using each sample script are included in this chapter under the following headings:

- Marking Organizations
- Marking People Containers
- Marking Organizational Units
- Marking Users
- Marking Static Groups
- Marking Dynamic (Filtered) Groups
- Marking Assignable Dynamic Groups
- Marking Group Containers

Important: The changes made by using these scripts cannot be automatically undone. Be sure to back up your data before running each script.

Two Approaches to Modifying the Existing Directory Tree

You can use one of two approaches for modifying the directory tree. One option is to make all the necessary modifications to your directory tree before loading the Identity Server LDIF and XML configuration files. This procedure is more error-prone, but may be faster if you have experience using LDAP.

The other option is to make a few modifications in your LDIF and XML files, and then start Identity Server to make sure those modifications were done correctly. This second approach is the recommended approach. For example, you may want to add the Identity Server object classes for each of your organizations, restart Identity Server, and verify that your organizations appear in the Identity Server Administration Console. Then add marker object classes for groups, check them and so forth.

Marking Organizations

If you used an existing organization as your default organization during installation, you do not have to make these changes. The installation program automatically added these object classes and attributes. Skip to "Marking People Containers" on page 78.

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

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

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

To Mark Organizations Using the Sample Script

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

   DirectoryServer_base`/shared/bin`

2. Set the `$base` variable to the base suffix of the directory tree to be managed by Identity Server. Example: `dc=MadisonParc,dc=com`

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 username 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 Identity Server, 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");
   ```

In Code Example 3-1, 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
```

Code Example 3-1 Organization entries with marker object classes. *(Continued)*

```
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

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 Mark People Containers Using the Sample Script

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

   DirectoryServer_base`/shared/bin`

2. Be sure the `$base` variable is set to the base suffix of the directory tree to be managed by Identity Server. Example: `dc=MadisonParc,dc=com`

   In the MadisonParc example, the script was also modified to include people containers located under the organizations. In Code Example 3-2, bold indicates the change in the search scope.

   Code Example 3-2 The 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 username 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`

   Enter People Container: Enter the name of the people container that contains the uids you want to modify. 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 Identity Server, 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 3-3, 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

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 Mark Organizational Units Using the Sample Script

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

   DirectoryServer_base`/shared/bin`

2. Set the `$base` variable to the base suffix of the directory tree to be managed by Identity Server. Example: `dc=MadisonParc,dc=com`.

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 username 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 `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 Identity Server, 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 3-4, 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

To each user entry, add the following object classes:

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

To Mark Users Using the Sample Script

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

   DirectoryServer_base`/shared/bin`

2. Be sure the `$base` variable is set to the base suffix of the directory tree to be managed by Identity Server. Example: `dc=MadisonParc,dc=com`

3. Be sure the `$base-component` variable is set to the base suffix of the directory tree.Example: `dc=MadisonParc,dc=com`

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

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

5. 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 Identity Server, 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
   ```

6. 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 3-5, 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
```

Code Example 3-5 User entry with user marker object class. *(Continued)*

```
objectClass: iplanet-am-managed-person
objectClass: iplanetPreferences
objectClass: iplanet-am-user-service
uid: scarter
cn: Sam Carter
userPassword: {SSHA}3XwjhBgbt6ae5syCndDeANoossEGRJ1NdnLyZw==
employeeType: Manager
departmentNumber: 1000
businessCategory: East
inetUserStatus: Active
```

Marking Static Groups

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 Mark Static Groups Using the Sample Script

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

   DirectoryServer_base`/shared/bin`

2. Set the `$base` variable to the base suffix of the directory tree to be managed by Identity Server. Example: `dc=MadisonParc,dc=com`.

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 username 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`

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 Identity Server, 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 3-6, 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

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 Mark Filtered Groups Using the Sample Script

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

   ```
   DirectoryServer_base/shared/bin
   ```

2. Set the $base variable to the base suffix of the directory tree to be managed by Identity Server.

   Example: dc=MadisonParc,dc=com

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 username 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 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 Identity Server, 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 3-7, 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

The *assignable* dynamic group is an Identity Server concept. In Identity Server, 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 Identity Server:

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

To Mark Assignable Dynamic Groups Using the Sample Script

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

DirectoryServer_base`/shared/bin`

2. Set the `$base` variable to the base suffix of the directory tree to be managed by Identity Server. Example: `dc=MadisonParc,dc=com`

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 username 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 `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 Identity Server, 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

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 Mark Group Containers Using the Sample Script

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

   DirectoryServer_base/shared/bin

2. Be sure the $base variable is set to the base suffix of the directory tree to be managed by Identity Server.

   Example: dc=MadisonParc,dc=com.

   In the MadisonParc example, the script was also modified to include all group containers located under organizations. In Code Example 3-8, the script changes are indicated in bold.

   Code Example 3-8 The 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 username 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 `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 Identity Server, 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 3-9, 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 to Identity Server Schema

If your existing directory tree contains object classes you've created that do not come with Directory Server, then you'll have to add those object classes and attributes to the Identity Server schema. In the examples in this section, the MadisonParc directory tree uses two object classes and two user attributes that do not come with the Directory Server schema or with Identity Server schema. These object classes and attributes help to distinguish MadisonParc employees at the top level of the directory tree from non-employees in the Customers and Suppliers organizations. Before Identity Server can manage these extensions, changes must be made in the following three Identity Server files:

- `umExisting.xml`
- `amEntrySpecific.xml` (for organization data)
- `amUser.xml` (for user data)

This chapter contains detailed instructions for making these modifications. The instructions are provided here to help you see your existing data in Identity Server after you run the Installation program.

For background information on the Identity Server schema and detailed information about customizing Identity Server, see "Understanding Identity Server XMLs and DTDs" in the *Programmer's Guide*.

Modifying the Creation Templates

The creation templates configure Identity Server to add or allow specific object classes and attributes when these entries are created. To expose custom object classes in the UI, 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.

The DAI Service

When you install Identity Server services, the `ums.xml file` is stored in Directory Server as the Directory Access Instructions (DAI) service. Identity Server will 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. Once you're finished modifying the files, you must reload the DAI service into Directory Server.

To Remove the DAI Service

In the following directory:

IdentityServer_base`/bin`

execute the following command:

```
./amadmin -u "user_naming_attibute=amadmin,ou=people,root_suffix" -w
  password -r   DAI
```

To Load the DAI Service

In the following directory:

IdentityServer_base/bin

execute the following command:

```
./amadmin -u "user_naming_attibute=amadmin,ou=people,root_suffix" -w
  password -s umsExisting.xml
```

To Modify the Creation Templates

1. If during installation, you chose to load the Identity Server schema, or if you have already run the `amserveradmin` command for any reason, skip this step and go on to step 2. Otherwise, remove the DAI service. In the following directory:

   IdentityServer_base/bin

   execute the following command:

   ```
   ./amadmin -u "user_naming_attibute=amadmin,ou=people,root_suffix" -w
     password -r DAI
   ```

2. Locate the following file:

UNIX	*IdentityServer_base*/SUNWam/config/ums/umsExisting.xml
Windows	*IdentityServer_base*\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.

   Under the following SubConfiguration:

   ```
   "BasicOrganization" id="CreationUmsObjects
   ```

   change

   ```
   <Value>objectClass=organization</Value>
   ```

   to

   ```
   <Value>objectClass=domain</Value>
   ```

In Code Example 3-10, bold indicates the changed value. Note that three lines down, the naming attribute dc was changed by Identity Server 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>
                                <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>
   ```

Example:

```
<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>
                        </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>
   ```

   Example:

```
<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>
            <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 (the `ums.xml` file or `umsExisting.xml` file).

   In the following directory:

   IdentityServer_base/`bin`

   execute the following command:

```
  ./amadmin -u "user_naming_attibute=amadmin,ou=people,root_suffix" -w password
-s
    IdentityServer_base/SUNWam/config/ums/umsExisting.xml
```

Adding Attributes to the Organization Schema

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

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

The Identity Server console uses the information in amEntrySpecific.xml for display purposes. Each Identity Server 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 98.) 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.

The "any" attribute

The any attribute in the XML descriptions may have five possible values: filter, display, adminDisplay, userReadOnly, required, or optional. The 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 UserId 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 Identity Server Console, as indicated by the use of the `filter` value.

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

The "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 Identity Server 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:

UNIX	*IdentityServer_base*`/SUNWam/config/xml/amEntrySpecific.xml`
Windows	*IdentityServer_base*`\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 Identity Server 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

:

UNIX	*IdentityServer_base*/SUNWam/locale/amEntrySpecific.properties
Windows	*IdentityServer_base*\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.Example:

```
iplanet-am-entry-specific-service-description=Identity Server 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 following directory:

   IdentityServer_base/bin

   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:

UNIX	*IdentityServer_base*/SUNWam/config/xml
Windows	*IdentityServer_base*\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 Identity Server. 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 Identity Server 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 95 and "The "type" attribute" on page 96). 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 Identity Server 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 ”Understanding Identity Server XMLs and DTDs” in the *Programmer’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:

UNIX	*IdentityServer_base*`/SUNWam/config/xml/amUser.xml`
Windows	*IdentityServer_base*`\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 Identity Server 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:

UNIX	*IdentityServer_base*`/SUNWam/locale/amUser.properties`
Windows	*IdentityServer_base*`\locale\amUser.properties`

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
```

4. Load all XML files.

In the following directory:

IdentityServer_base/bin

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 `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:

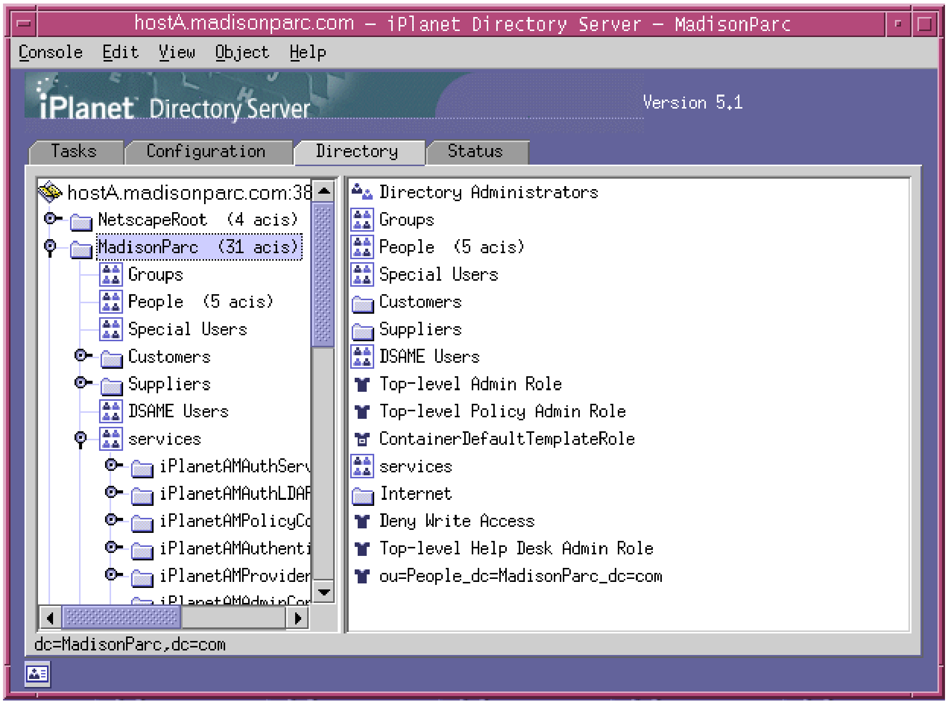
UNIX	*IdentityServer_base*/SUNWam/config/xml
Windows	C: *IdentityServer_base*\config\xml

Loading Identity Server LDIF into Your Directory

Identity Server provides two different LDIF files to help you make the necessary modifications in your directory when you are enabling User Management services. You'll need to follow instructions for both loading `installExisting.ldif` and `install.ldif`.

Figure illustrates the MadisonParc directory tree after enabling both User Management and Policy Management services. Both `installExisting.ldif` and `install.ldif` files were loaded into an existing directory.

Figure 3-6 The MadisonParc directory tree with both `installExisting.ldif` and `install.ldif` added.



```
hostA.madisonparc.com:38
  NetscapeRoot  (4 acis)
  MadisonParc  (31 acis)
    Groups
    People  (5 acis)
    Special Users
    Customers
    Suppliers
    DSAME Users
    services
      iPlanetAMAuthServ
      iPlanetAMAuthLDAP
      iPlanetAMPolicyCo
      iPlanetAMAuthenti
      iPlanetAMProvider
      iPlanetAMAdminCor
```

```
Directory Administrators
Groups
People  (5 acis)
Special Users
Customers
Suppliers
DSAME Users
Top-level Admin Role
Top-level Policy Admin Role
ContainerDefaultTemplateRole
services
Internet
Deny Write Access
Top-level Help Desk Admin Role
ou=People_dc=MadisonParc_dc=com
```

dc=MadisonParc,dc=com

installExisting.ldif

The `installExisting.ldif` file contains Identity Server-specific entries that are loaded into Directory Server during installation. Typically, you will not need to modify this file before it gets loaded during the installation process.

You can use the `ldapmodify` utility that comes with Directory Server to load `installExisting.ldif`. In the MadisonParc example, when you load the LDIF, the following occurs:

- Users and marker object classes required for Identity Server 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.

To Load the installExisting.ldif File

1. Go to the following directory:

UNIX	*IdentityServer_base*`/SUNWam/config/ldif`
Windows	*IdentityServer_base*`\config\ldif`

2. At the command line, enter the following:

```
ldapmodify -v -c -D "cn=Directory manager" -w password -a
  -f installExisting.ldif
```

NOTE You must specify the `-c` option. Be sure you install only `installExisting.ldif`, and no other files in the same directory.

The Identity Server administration user `amAdmin` will be created under the `ou=People,dc=MadisonParc,dc=com` people container. This is the top level administrator for Identity Server. 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 Identity Server console is started.

install.ldif

To Load the install.ldif File

1. Go to the following directory:

UNIX	*IdentityServer_base*`/SUNWam/config/ldif`
Windows	*IdentityServer_base*`\config\ldif`

2. Enter the following command:

```
  ldapmodify -v -c -D "cn=Directory manager" -w password -a -f
    install.ldif
```

NOTE You must specify the `-c` option. Be sure you install only `install.ldif`, and none of the other files in the same directory.

Results of Identity Server and Directory Modifications

After making the modifications in the previous steps, all entries in your existing directory will be manageable by Identity Server. The existing ACIs for the organization administrators do not have to be modified. Even though Identity Server uses roles and ACIs by default, your existing groups and ACIs will still work.

You can convert a groups-based directory tree to one that leverages roles and ACIs. If you choose to do this, you can use the Identity Server organization administrator roles and assign them to your existing `organizationList` administrators. For more information, see the *Administrator's Guide.*

Index