Oracle® Fusion Middleware

Administrator's Guide for Oracle Authentication Services for Operating Systems

11*g* Release 1 (11.1.1)

E16454-02

July 2010

ORACLE®

Oracle Fusion Middleware Administrator's Guide for Oracle Authentication Services for Operating Systems, 11*g* Release 1 (11.1.1)

E16454-02

Primary Author: Ellen Desmond

Contributing Author: Buddhika Kottahachchi

Contributors: Olfat Aly, Vasuki Ashok, Quan Dinh, Prathima Nagesh, Loganathan Ramasamy, Daniel Shih, Olaf Stullich, Arun Theebaprakasam, Dai Vu

Contents

4 Migrating Entries to Oracle Internet Directory

5 Configuring Active Directory Integration

6 Managing Oracle Authentication Services for Operating Systems

7 Restricting User Logins

A Troubleshooting

List of Figures

Preface

This is the Administrator's Guide for Oracle Authentication Services for Operating Systems, Release 11*g* Release 1 (11.1.1). It explains how to install, configure, and manage Oracle Authentication Services for Operating Systems on server and client systems.

Audience

This document is intended for Linux and UNIX system administrators. You need to be familiar with Oracle Internet Directory before you attempt to install or configure Oracle Authentication Services for Operating Systems.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/support/contact.html or visit http://www.oracle.com/accessibility/support.html if you are hearing impaired.

Related Documents

For more information about Oracle Authentication Services for Operating Systems 11*g* Release 1 (11.1.1), see:

- The README document accompanying this release
- Note 1064891.1: Oracle Authentication Services for Operating Systems Documentation Addendum (11.1.1.3). This document is available on My Oracle Support at `https://support.oracle.com/`

Also see the following documents in the Oracle Application Server 11*g* Release 1 (11.1.1) documentation set, at `http://www.oracle.com/technology/documentation/`:

- *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*
- *Oracle Fusion Middleware Installation Planning Guide*
- *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Integration Platform*
- *Oracle Fusion Middleware Reference for Oracle Identity Management*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Product Overview

Oracle Authentication Services for Operating Systems enables you to centralize storage, authentication, and management of user identities using Oracle Internet Directory.

This chapter contains the following topics:

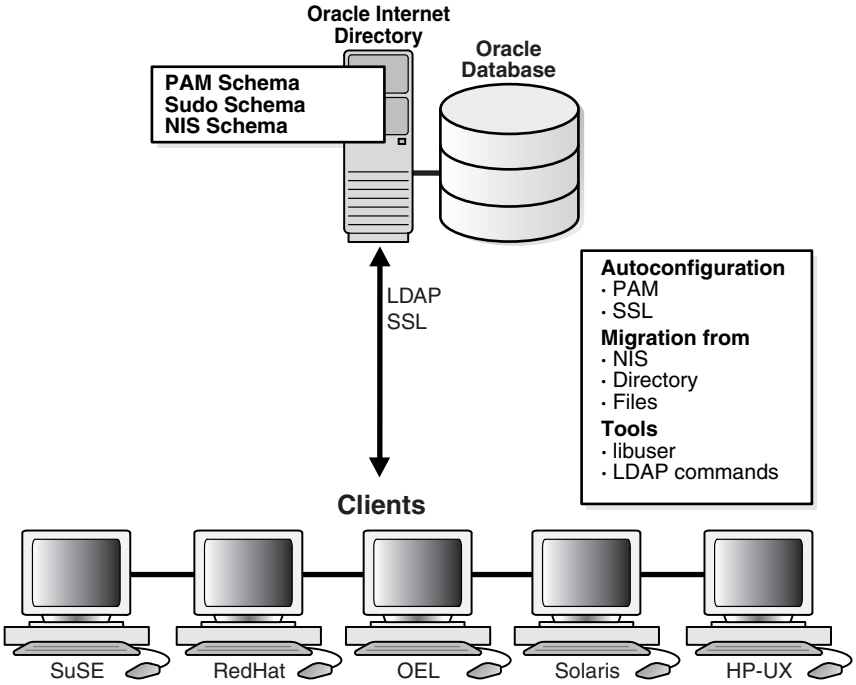
Introduction to Oracle Internet Directory

Oracle Internet Directory is a standards-based directory server that leverages the security, scalability, and reliability of Oracle Database to store users, groups, and other types of entries. Oracle Internet Directory supports password policy enforcement. Oracle Internet Directory can be synchronized with third-party directory servers, such as Active Directory.

Features of Oracle Authentication Services for Operating Systems

Oracle Authentication Services for Operating Systems enables you to use Oracle Internet Directory for authentication on Linux- and UNIX-based operating systems. Configuration scripts automate the configuration of Pluggable Authentication Modules (PAM) and Secure Sockets Layer (SSL). You can then migrate existing entries from NIS, files, or another LDAP-compliant directory, and optionally configure features such as password policy enforcement, `sudo`, and automount. Oracle Internet Directory tools are available for entry management, and `libuser` tools can be used for many operations. These features are summarized in Figure 1–1.

Figure 1–1 Features of Oracle Authentication Services for Operating Systems



Components of Oracle Authentication Services for Operating Systems

In Oracle Fusion Middleware 11*g* R1 Patch Set 2 (11.1.1.3.0), the Oracle Internet Directory installation contains the following components, which are used by Oracle Authentication Services for Operating Systems:

- SSL and non-SSL server configuration scripts
- SSL and non-SSL client configuration scripts
- Support for migration from NIS as well as from flat file-based authentication
- Support for migration from a third party LDAP directory to Oracle Internet Directory.
- Support for migration of sudo policy from a `sudoers` file to Oracle Internet Directory
- Support for migration of automounts to Oracle Internet Directory

How User Authentication Works With Oracle Internet Directory

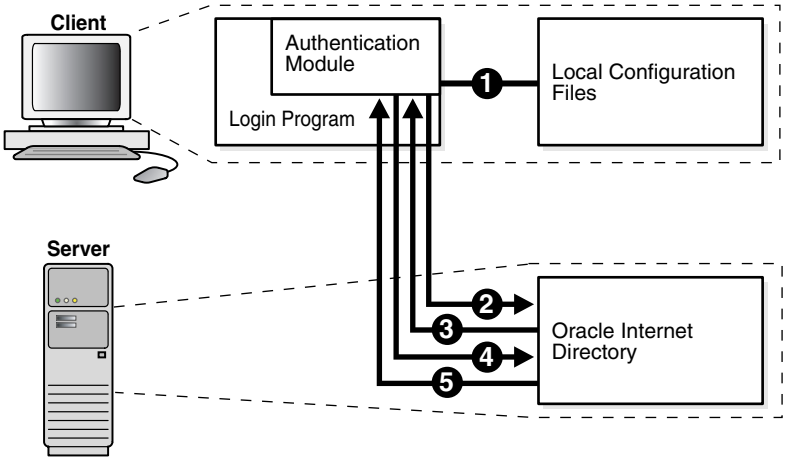
When a user provides credentials (a username and password) to `login`, `xdm`, `ssh`, `su`, or some other client login program, the following events occur.

1. An authentication module in the login program examines local configuration files to determine how to authenticate the user. The files contain information such as the method to use (LDAP), the location of the server, and, if SSL is configured, the certificate to use.
2. The authentication module attempts authenticate the user against the Oracle Internet Directory server with the user's credentials. If SSL is configured, the module first establishes the SSL communications channel using the certificate.

3. If Oracle Internet Directory determines that the credentials are correct and the account is active, the user's login attempt succeeds. Otherwise, the user's login attempt fails.
4. If the user login attempt succeeds, the module queries Oracle Internet Directory again for the user's group membership information.
5. Oracle Internet Directory returns the group membership information.

These events are shown in Figure 1–2.

Figure 1–2 Authentication Using Oracle Internet Directory



Configuration Overview

To configure Oracle Authentication Services for Operating Systems, you perform the following steps:

1. Install Oracle Internet Directory. See the *Oracle Fusion Middleware Installation Guide for Oracle Identity Management* for your platform.
2. Apply 11*g* R1 Patch Set 2 (11.1.1.3.0).
3. Execute the configuration scripts to configure the server and clients for user authentication.
4. Configure password policies.
5. Migrate entries from NIS, local files, or another LDAP-compliant directory to Oracle Internet Directory.
6. Configure `sudo` and migrate `sudo` entries to Oracle Internet Directory.
7. Optionally, you can configure integration with Active Directory so that you can use credentials stored in Active Directory for authentication on a Linux or UNIX-based operating system.
8. Optionally, you can restrict user logins on individual machines.

Management Overview

After you configure Oracle Authentication Services for Operating Systems and migrate your data to Oracle Internet Directory, you must use specific tools to manage users, passwords, and other data. Specifically, you must use:

- Oracle Directory Services Manager
- The LDAP tools and bulk tools in `$ORACLE_HOME/bin`
- The `passwd` command
- Certain platform specific tools:
  - The `libuser` tools on Linux distributions that support it, with some limitations. See libuser Tools.
  - The command `mkuser` and similar AIX tools with the option `-R LDAP`. See AIX-Specific Tools.

Additional Documentation

For more information about Oracle Authentication Services for Operating Systems 11*g* Release 1 (11.1.1), see:

- The README document accompanying this release
- Note 1064891.1: Oracle Authentication Services for Operating Systems Documentation Addendum (11.1.1.3). This document is available on My Oracle Support at `https://support.oracle.com`.

2

Before You Configure

Before configuring Oracle Authentication Services for Operating Systems, ensure that you are using a supported operating system and the supported version of Oracle Internet Directory. Then, before you start the install, determine which of the optional product features you will use and locate the scripts you will use for migration.

This chapter contains the following topics:

Verify Your Client and Server Operating Systems

Oracle Authentication Services for Operating Systems has both server and client components. The server is the computer that runs Oracle Internet Directory. The client is a computer that uses the services of Oracle Internet Directory for authentication.

For up-to-date information about supported server and client operating systems, please consult the following documents:

- The README document accompanying this release
- Note 1064891.1: Oracle Authentication Services for Operating Systems Documentation Addendum (11.1.1.3). This document is available on My Oracle Support at `https://support.oracle.com`.

Install Oracle Internet Directory and Oracle Directory Integration Platform

Before you can configure Oracle Authentication Services for Operating Systems, you must install Oracle Internet Directory. If you plan to migrate entries from an existing LDAP-compliant directory, or to synchronize Oracle Internet Directory with another directory, such as Active Directory, you must install Oracle Directory Integration Platform along with Oracle Internet Directory.

> **See Also:** The *Oracle Fusion Middleware Installation Guide for Oracle Identity Management* for your platform for information about installing Identity Management components.

Upgrade from Oracle Authentication Services for Operating Systems 10*g*

If you have already installed Oracle Authentication Services for Operating Systems 10*g*, you do not need to reconfigure your server or client machines unless you are changing some configuration features, such as ports or SSL certificate.

Upgrade to Oracle Internet Directory 11*g* as described in *Oracle Fusion Middleware Upgrade Guide for Oracle Identity Management*. Apply 11*g* R1 Patch Set 2 (11.1.1.3.0).

If you need to change the configuration, use the 11*g* scripts, as described in Chapter 3, "Configuring Oracle Authentication Services for Operating Systems."

In Oracle Internet Directory 11g Release 1 (11.1.1) and later, anonymous binds are allowed by default, but anonymous users can only perform search operations on the root DSE entry. When you upgrade, however, Oracle Internet Directory enables anonymous binds. If, for some reason, anonymous binds have been disabled, you can enable them by using the `ldapmodify` command, as described in the Troubleshooting section "Users Cannot Log In".

Determine Which Product Features You Will Use

Before you begin the installation, consider which features of the product you are likely to use. For basic functionality, you must run the server script on the system where you are running the Oracle Internet Directory server, then run the client script on each client. These scripts configure the server and clients for LDAP authentication. In addition to configuring basic LDAP authentication, you can choose from the following options:

- Secure Socket Layer (SSL)–Unless your server and clients are isolated from the internet, you should enable SSL. To do so, use the SSL versions of the server and client configuration scripts. The `libuser` tool `system-config-users` requires SSL when you use it with Oracle Authentication Services for Operating Systems on Red Hat or Oracle Enterprise Linux.
- Certificate and wallet to use with SSL–The SSL server configuration script can use an existing certificate or generate a self-signed certificate, which is not designed for production mode. If you plan to use an existing certificate, you must have already configured Oracle Internet Directory in SSL mode with this certificate. You can also choose to use a customized wallet instead of the default wallet.

  See Also:

  - The "Configuring Secure Sockets Layer (SSL)" chapter in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for information on configuring Oracle Internet Directory in SSL mode.
  - *Oracle Fusion Middleware Administrator's Guide* for information on Oracle wallets.

- Current authentication source to migrate from–If you are using files, NIS, or another LDAP server, you can migrate to Oracle Internet Directory.
- Whether to configure the `libuser` tools to use LDAP–The GUI tool `system-config-users` and the command-line utilities (`luseradd`, `luserdelete`, etc.) exist, by default, on Red Hat Enterprise Linux and Oracle Enterprise Linux. You can configure the `libuser` tools to work with LDAP, so that adding a user with `luseradd`, for example, adds the user entry to Oracle Internet Directory. If you do not use the `libuser` tools, you must use Oracle

Directory Manager, Oracle Internet Directory bulk tools, or Oracle Internet Directory LDAP tools to configure entries directly in Oracle Internet Directory. If your client is Red Hat Enterprise Linux or Oracle Enterprise Linux, the client script will prompt you as to whether you want to configure `libuser`.

Note:

- To use `libuser` tools, you must configure your client and server for SSL.
- If you plan to use Oracle Internet Directory to enforce password policies, you cannot use tools in the `libuser` package to add passwords or entries containing passwords.
- You cannot use the non-`libuser` commands `useradd`, `userdel`, `groupadd`, or `groupdel` for user or group administrative tasks.

- Data to migrate–Open Source scripts such as those described in the next section support migration of users and groups and other configuration data from NIS or from files. Oracle Authentication Services for Operating Systems includes tools for migrating from a third-party LDAP directory server.
- Whether to migrate `sudo`–You can use Oracle Internet Directory instead of a `sudoers` configuration file to authenticate `sudo` commands.
- How to enforce password policies–You can continue to use the operating system for password enforcement. Alternatively, you can use Oracle Internet Directory for centralized password policies.
- Whether to integrate with Active Directory–You can use credentials stored in Active Directory for user authentication on Linux or UNIX-based operating systems.

Get NIS Migration Tools

If you have user, group, and other entries maintained in the local file system or in NIS/NIS+, you can move to LDAP as your storage mechanism for these entries. There are tools available to extract the existing information and produce output files in the LDAP Data Interchange Format (LDIF). Once you have your information in LDIF files, you can use the `ldapadd` tool to load the information into Oracle Internet Directory.

AIX 5.3

You must use the `sectoldif` and `nistoldif` tools on AIX for user and group migrations. Do not use the migration tools from `http://www.padl.com/`.

Other Platforms

A number of free tools are available. We have validated the process of migrating information using the LDAP migration tools available at:

`http://www.padl.com/`

If you have the `openldap` packages installed on your host, you will find the same migration tools at: `/usr/share/openldap/migration`.

Download SUDO Package

If you want to migrate the contents of the `sudoers` file to LDAP, you must run a migration script and build `sudo` with LDAP enabled. You can obtain the `sudo` package from:

`http://www.gratisoft.us/sudo`

Create and Index New Custom Attributes (Optional)

You cannot successfully search for an attribute in Oracle Internet Directory unless the attribute is indexed. If you plan to add custom attributes, you can index them at the time you create them by using Oracle Directory Manager. You can also use `ldapmodify` to create an indexed attribute. You would use an LDIF file such as this:

```
dn: cn=catalogs
changetype: modify
add: orclindexedattribute
orclindexedattribute: attribute_name
```

Alternatively, you can index attributes after they have been created in Oracle Internet Directory by using `catalog`, as explained in "Using Custom Attributes in Oracle Internet Directory" on page 4-7.

Note: If you attempt to perform a search with a non-indexed attribute specified as a required attribute, the server will return the error:

```
Function not implemented. DSA unwilling to perform.
```

See `https://support.oracle.com`.

Platform-Specific Tasks

The following pre-installation tasks are platform-specific.

HP-UX

If a computer that you plan to use as a client is running HP-UX, you must download and install: LDAP-UX Integration J4269AA, HP-UX 11iv2 for Workstations and Servers B.04.00.03, as `root`. You can download the software from:
`http://h20293.www2.hp.com/portal/swdepot/try.do?productNumber=J4269AA`

Solaris 5.9 and 5.10

If you plan to run the SSL version of the server configuration script on Solaris 5.9 or 5.10, you must ensure that Oracle Internet Directory is using the standard LDAP ports, `389` and `636`, for non-SSL and SSL, respectively.

If necessary, start a new Oracle Internet Directory instance using the standard LDAP ports. Proceed as follows:

1. Stop all Oracle Internet Directory instances by using the `opmnctl` command. Type:

   ```
   opmnctl stopproc process-type=OID
   ```

2. As `root`, execute the command:

```
$ORACLE_HOME/oidRoot.sh
```

3. Create a new component of type `OID`. For example, to create a component with component name `oid2` and namespace `dc=us,dc=example,dc=com`, type:

```
$ORACLE_INSTANCE/bin/opmnctl createcomponent -componentType OID \
   -componentName oid2 -Db_info \
   "myhost.us.example.com:1521:dbservice.us.example.com" \
   -Port 389 -Sport 636 -Namespace "dc=us,dc=example,dc=com"
```

4. Start the Oracle Internet Directory instances. For example, to start component `oid2`, type:

```
$ORACLE_INSTANCE/bin/opmnctl startproc ias-component=oid2
```

See Also: The chapter "Managing Oracle Internet Directory Instances," in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.

3

Configuring Oracle Authentication Services for Operating Systems

This chapter contains the following topics:

Before you begin the procedures described in this chapter, you must perform the prerequisite procedures described in Chapter 2.

Introduction

This introduction contains the following sections:

SSL Support

Oracle Internet Directory can be configured for SSL-no authentication, SSL-server authentication and SSL-mutual authentication modes. In all three modes, the data is encrypted during transmission. Oracle Internet Directory comes pre-configured with the SSL-no authentication mode. However, some clients such as the PAM_LDAP clients used for Linux user authentication do not support this mode and only support SSL-server authentication mode.

The initial server configuration process enables you to configure Oracle Internet Directory for SSL-server authentication mode. You can use an existing certificate or let the SSL configuration script generate a self-signed certificate for you. To use an

existing certificate, you must have already configured Oracle Internet Directory in SSL mode with this certificate. See the "Configuring Secure Sockets Layer (SSL)" chapter in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for information on configuring Oracle Internet Directory in SSL mode.

Only Privacy Enhanced Mail (`PEM`) format is supported. This is a base64 encoded DER certificate, enclosed between these two lines:

```
-----BEGIN CERTIFICATE-----

-----END CERTIFICATE-----.
```

> **Note:** Self-signed certificates are not intended for production use.

Self Signed Certificates

If you do not specify an existing certificate, the SSL server configuration script generates two Oracle wallets:

1. Test Certificate Authority (CA) Wallet–used to sign the Oracle Internet Directory SSL Server Certificate. This consists of the following files in `$ORACLE_INSTANCE/wallet/root`:
   - `cakey.txt`–a 1024 bit RSA private key
   - `cacert.txt`–based64 encoded certificate
2. Oracle Internet Directory SSL Server Certificate. This consists of the following files in `$ORACLE_INSTANCE/wallet/server`:
   - `creq.txt`–Oracle Internet Directory SSL Server Certificate Request
   - `cert.txt`–Oracle Internet Directory SSL Server Certificate signed by Test CA Wallet
   - `cwallet.sso`–Oracle Internet Directory SSL Server Wallet for auto-login
   - `ewallet.p12`–PKCS12 encoded Oracle Internet Directory SSL wallet

> **Note:** The PKCS12-encoded wallets contain the private keys for the relevant entities and are protected by a wallet password that you set when running the SSL server configuration script.

For a client to trust the Oracle Internet Directory SSL Server Certificate (2) it must trust the Test CA Wallet (1). Since most Linux clients work with the PEM format, a copy of the Test CA Wallet (1) in PEM format is available at: `$ORACLE_INSTANCE/OID/admin/wallet/pem.cert`.

Password Policy Enforcement

Oracle Internet Directory ships with a rich set of password policies that can be leveraged for centralized password policy management. See the chapter on Password Policies in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* 11*g* Release 1 (11.1.1) to understand the concepts governing these features.

Oracle Internet Directory supports two types of password policies: value policies and state policies. Value policies govern password construction requirements, such as minimum length. State policies govern things like password expiration and lockout. On Linux and UNIX-based operating systems, state policies are traditionally handled

in the shadow password file using the password aging feature. These policies can be applied in a fine-grained manner down to the level of a single user entry.

You can use Oracle Internet Directory to enforce both value and state policies. Value policy violations result in visible error message on the Linux client, but state policy violations simply result in login failures. This is because the `pam_ldap` client does not display the messages that Oracle Internet Directory sends as additional information with the LDAP bind failure.

To use Oracle Internet Directory for centralized password policies, you must disable value and state policies local to the operating system. The procedure for doing this is described in "Configuring Oracle Internet Directory for Centralized Password Policies" on page 3-10.

If you do not want to use Oracle Internet Directory for password policy enforcement, you must disable password policies in Oracle Internet Directory by setting `orclpwdpolicyenable` to `0`. To avoid messages about password syntax, you must also disable the password syntax check by setting `pwdCheckSyntax` to `0`.

> **See Also:** The Password Policies chapter in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.

Active Directory Integration

If you have users in Active Directory, and you want to use the credentials stored in Active Directory for Linux authentication, you can configure Oracle Directory Integration Platform to integrate with Active Directory. The configuration process is described in Chapter 5, "Configuring Active Directory Integration."

Directory Plug-ins

A directory server plug-in is a customized program that extends the capabilities of the Oracle Internet Directory server. The procedures for augmenting Active Directory entries and for setting up external authentication with Active Directory both include setting up plug-ins. These procedures are described in Chapter 5, "Configuring Active Directory Integration."

> **See Also:** *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for more information about directory server plug-ins.

Language Support

Before you run the configuration scripts, you must set your locale by setting the `NLS_LANG` environment variable. After you set `NLS_LANG`, the scripts will work correctly when you provide input in your local language.

Tools Used During Configuration

Some of the tasks described in this chapter require you to use Oracle Internet Directory or Oracle Directory Integration Platform tools. These tools include:

- The Oracle Internet Directory LDAP command-line tools–These are located in the `$ORACLE_HOME/bin` directory. These tools are `ldapsearch`, `ldapbind`, `ldapmodify`, `ldapdelete`, `ldapcompare`, `ldapmoddn`, `ldapaddmt` and `ldapmodifymt`. For interaction with the Oracle Internet Directory server, you must use the LDAP tools in `$ORACLE_HOME/bin` and not those shipped in the operating system base image.

- The Oracle Internet Directory bulk tools–These are also located in the `$ORACLE_HOME/bin` directory. These tools are `bulkload`, `bulkmodify`, `catalog`, `bulkdelete` and `ldifwrite`. The bulk tools allow you to perform bulk operations, such as adding or deleting a large number of entries.

  One important bulk tool is the `catalog` tool. This tool enables you to add indexes to attributes in Oracle Internet Directory. Attributes must be indexed in order to be searchable. This example adds an index to the attribute `uid`:

  ```
  catalog connect="connect_str" add="TRUE" attribute="uid"
  ```

- The `opmnctl` command–You use this to stop and start the Oracle Internet Directory server.
- The `Oracle Directory Integration Platform` command `syncProfileBootstrap`–You use `syncProfileBootstrap` when configuring SSL for communication between Oracle Directory Integration Platform and Active Directory and when migrating data from another LDAP-compliant directory to Oracle Internet Directory.

See Also:

- *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* and the *Oracle Fusion Middleware Reference for Oracle Identity Management* for information about the Oracle Internet Directory LDAP tools, bulk tools, and `opmnctl`.
- The chapter entitled "Oracle Directory Integration Platform Tools" in the *Oracle Fusion Middleware Reference for Oracle Identity Management* and the chapter entitled "Configuration of Directory Synchronization Profiles" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Integration Platform* for more information on `syncProfileBootstrap`.

Configuring Oracle Authentication Services for Operating Systems on the Server

Use the server configuration script to configure the server for UNIX or Linux authentication, as follows.

Note:

- Look for error messages printed to the screen while the configuration script is running. An example of a successful run is provided for comparison in Appendix E, "Sample Script Output."
- You can switch between SSL and non-SSL configurations. See "Switching Between SSL Authentication and Non-SSL Configurations" on page 3-10.
- You can disable either the SSL port or the non-SSL port if you are not using it. You do this by changing the value of the configuration attribute `orclSSLEnable`. See the entry for `orclSSLEnable` in the Attribute Reference chapter of the *Oracle Fusion Middleware Reference for Oracle Identity Management*.

1. Execute the server script on the server as the same user who installed Oracle Internet Directory. Change directory to `$ORACLE_HOME/oas4os/bin`, then type:

```
./sslConfig_OIDserver.sh
```

or

```
./config_OIDserver.sh
```

2. You will be prompted for `ORACLE_HOME`, `ORACLE_INSTANCE`, realm (naming context), SSL- and non-SSL port, OID component name (for example, `oid1`), and password for `cn=orcladmin`. Supply the appropriate values in response to the prompts.

   If you have set `ORACLE_HOME` or `ORACLE_INSTANCE` as environment variables, you will not be prompted for them.

3. You will be asked if you want the client machines to connect to Oracle Internet Directory anonymously or by using a specific user DN and password. If you answer `y`, the script will enable anonymous binds in Oracle Internet Directory server and clients will connect to the server by using anonymous binds. If you choose `n`, you will be prompted for the DN and password for connecting to Oracle Internet Directory.

   See Also:

   - The "Managing Accounts and Passwords" chapter in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*
   - The "Managing Authentication" chapter in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*

4. If you are using the SSL configuration script, the script will print:

   ```
   You can provide an SSL Certificate or use the script to create and update OID
   SSL configuration with a test certificate.
   Do you have an SSL Certificate [y/n]:
   ```

   - If you type `y` in response to that prompt, you will be prompted to supply the path to the certificate. Specify the full path, including the filename, in response to the prompt, for example: `/home/jdoe/sslcert.pem`. Only `PEM` format is supported.
   - If you type `n` in response to the prompt, you will be prompted for the wallet password. The script configures Oracle Internet Directory for SSL server side authentication mode with a self-signed certificate.

The SSL version of the script configures the non-SSL port for StartTLS, which allows SSL and non-ssl connections to use the same port. if the self-signed certificate option was chosen, the script also configures the SSL port for connections from clients that do not support StartTLS. (If the self-signed certificate option was not chosen, you are expected to have already configured OID's SSL port with your custom certificate.)

The server script creates the client script, `sslConfig_OIDclient.sh` or `config_OIDclient.sh`, in the location `$ORACLE_INSTANCE/OID/oas4os/`*`component_name`*`/scripts_`*`timestamp`*, customizing it for your environment. The server script prints the client script location on the screen at the end of the script as follows:

```
OAS4OS Client Config Script: client_script_path
```

The script updates several Oracle Internet Directory server parameters with the information it has gathered. The SSL version of the script restarts the Oracle Internet Directory server. The non-SSL version does not.

See Also:

- Appendix E, "Sample Script Output"
- Appendix F, "LDAP Containers Added by Configuration Script"

Configuring Oracle Authentication Services for Operating Systems on the Client

You configure each client for UNIX or Linux authentication by running a client configuration script. Follow these steps.

Solaris 9

The following steps are specific to Solaris 9.

1. On Solaris 9 only, download the Sun Java System Directory Server Resource Kit SDRK52 and install it as `root`.
2. After installing the Sun Java System Directory Server Resource Kit, before you run the client configuration script, modify the environment variables `PATH` and `LD_LIBRARY_PATH` so that `PATH` includes *`installroot`*`/lib/nss/bin` and LD_LIBRARY_PATH includes *`installroot`*`/lib`, where *`installroot`* is the directory where you installed the Sun Java System Directory Server Resource Kit For example, if you installed the software in `/usr`, add `/usr/lib/nss/bin` to `PATH` and add `/usr/lib` to `LD_LIBRARY_PATH`.
3. Perform the tasks described under "All Client Platforms" on page 3-8.

AIX 5.3

The following steps are specific to AIX 5.3.

Install the LDAP Client on AIX

The base AIX 5L LDAP client is packaged in the `ldap.client` file sets located on the AIX 5L product media.

If you plan to use SSL to connect to the LDAP server, you must install the `gskta.rte` and `ldap.max_crypto_client` file sets located on the AIX 5L Expansion Pack. The installation procedure is described in "Install SSL-Related Client Packages on AIX" on page 3-8.

1. Install the base AIX LDAP client package. You can find it in the `ldap.client` file sets located on the AIX 5L product media. Execute the following command to install the basic package:

   ```
   installp -acgXd LPPSOURCE ldap.client
   ```

   where *`LPPSOURCE`* is the location of your Licensed Product Packages (LPPs).

   Note: You can also use SMIT or the Web-based System Manager to install the LPPs.

2. Verify the installation by typing the following command:

   ```
   lslpp -l "ldap"
   ```

The output from the `lslpp` command should include `ldap.client.adt` and `ldap.client.rte`.

Add At Least One User and One Group to Oracle Internet Directory on AIX

Before you execute the client script on AIX, you must add at least one user and group to LDAP. Otherwise, the `mkseccldap` command executed by the configuration script on AIX might fail with one of these error messages:

```
Cannot find users from all base DN
client setup failed."

Cannot find the group base DN from the LDAP server.
Client setup failed."
```

To prevent this problem, you can simply add one user and one group, or you can migrate all your users and groups to Oracle Internet Directory now, rather than waiting until you have run the configuration script.

> **See Also:** "LDAP configuration management and troubleshooting on AIX" at `http://www.ibm.com/developerworks/` for more information and an alternative solution.

To migrate all your users and groups, proceed as follows:

1. Convert local system entries to LDAP entries by using the `sectoldif` command. Type:

   ```
   sectoldif -d "realm" -S "RFC2307" > users.ldif
   ```

2. Ensure that all users to be migrated are associated with a system group or net group. That is, edit `user.ldif` so that each user has a `gidnumber`. For example:

   ```
   dn: uid=test,ou=People,dc=us,dc=example,dc=com
   uid: test
   objectClass: posixaccount
   objectClass: shadowaccount
   objectClass: account
   cn: test3
   uidnumber: 209
   gidnumber: 502
   homedirectory: /home/test
   loginshell: /usr/bin/ksh
   userpassword: passwordhash
   shadowlastchange: 13182

   cn=testgroup,ou=Group,dc=us,dc=example,dc=com
   gidnumber=502
   cn=testgroup
   objectclass=posixGroup
   objectclass=groupOfUniqueNames
   objectclass=top
   ```

3. Add the user entries in `users.ldif` to Oracle Internet Directory:

   ```
   ldapadd -h host -p port -D "cn=orcladmin" -q -c -f users.ldif
   ```

4. If you are using the non-SSL script, perform the tasks described under "All Client Platforms" on page 3-8. Otherwise, proceed as described in the next section.

Install SSL-Related Client Packages on AIX

If you plan to use SSL to connect to the LDAP server, you must install the `gskta.rte` and `ldap.max_crypto_client` file sets located on the AIX 5L Expansion Pack.

1. The following packages are required for SSL Configuration on an AIX 5L Version 5.3 client:

   - `gskta.rte`
   - `ldap.max_crypto.client`

   If these packages are not already installed, install them from the AIX 5L Version 5.3 Expansion Package CD (5705-603) or from the equivalent package in Tivoli Directory Server, which is available at the IBM web site. Type:

   ```
   installp -acgXd LPPSOURCE gskta ldap.max_crypto_client
   ```

2. Verify the installed packages by typing:

   ```
   lslpp -l | grep "gskta*" "*ldap*"
   ```

   The output of the `lslpp` command should include `gskta.rte`, `ldap.client.adt`, `ldap.client.rte`, `ldap.max_crypto_client.adt`, and `ldap.max_crypto_client.rte`.

3. If necessary, create a symbolic link in `/usr/lib` to the new LDAP client library. For example:

   ```
   ln -s  /opt/IBM/ldap/release/lib/libidsldap.a /usr/lib/libibmldap.a
   ```

4. Proceed as described for all client platforms.

5. Verify that LDAP SSL is enabled by using `ldapsearch`, for example:

   ```
   ldapsearch -h myserver.example.com -Z -K /etc/security/ldap/key.kdb \
              -Q -b "" -s base objectclass=*
   ```

6. Verify that authentication is working correctly by logging into your client machine using `telnet`, `rlogin`, `ssh`, or a similar program.

AIX 6.1

The SSL client configuration script fails on AIX 6.1 due to a problem with the `mkseccldap` tool. You can only configure Oracle Authentication Services for Operating Systems in non-SSL mode, using the non-SSL configuration script, on AIX 6.1.

All Client Platforms

1. Copy the client configuration script from the server to the client after you have run the server configuration script. The server script edits the client script, customizing it for your environment.

   For SSL Server Authentication enabled Linux clients, use the client script `sslConfig_OIDclient.sh`. For non-SSL Linux clients, use `config_OIDclient.sh`. Copy the script from `$ORACLE_HOME/ldap/bin` on the server to each client you want to configure.

2. Execute the client configuration script on the client as the `root` user. Type:

   ```
   ./sslConfig_OIDclient.sh
   ```

   or

```
./config_OIDclient.sh
```

> **Note:** Look for error messages printed to the screen while the configuration script is running. An example of a successful run is provided for comparison in Appendix E, "Sample Script Output."

3. The script prints the host and port, then prompts:

   ```
   Do you want to configure test-host to authenticate users against the above
   OID LDAP server [n]: y
   ```

   If the host and port are correct, confirm that you want to configure the client to authenticate against the LDAP server. If either is incorrect, type `n`, edit the script to correct the problem, and execute the script again.

4. If, while running the server configuration script, you specified that you did not want to use anonymous binds, the client script prints the proxy DN and prompts you for the password to use for connecting to Oracle Internet Directory. Supply the same password that you provided when configuring the server.

5. If the client is Red Hat Enterprise Linux or Oracle Enterprise Linux, the client script prompts you as to whether you want to configure the `libuser` package to work with LDAP. Respond `y` if you want `libuser` to be configured. If you configure `libuser` to work with LDAP, adding a user with `luseradd`, for example, adds the user entry to Oracle Internet Directory.

The script configures Pluggable Authentication Modules (PAM) on the client operating system to use Oracle Internet Directory for user authentication. The exact tasks performed depend on the operating system type. The script performs the following basic tasks:

- Makes configuration changes to `nsswitch.conf` so that `ldap` is an option for `passwd`, `group` and `shadow`.
- Configures `/etc/ldap.conf` and `/etc/openldap/ldap.conf` with the correct URI, Base DN
- Optionally, configures the `libuser` package (via `libuser.conf`) for user management on Red Hat Enterprise Linux and Oracle Enterprise Linux.

> **Note:** The script makes backup copies of the files it touches in subdirectories of the `/etc` directory. These subdirectories have names of the form `oracle_backup_`*`time_stamp`*. For example, a backup directory created 18:54:46 on Jan. 13 2010 would have the name `/etc/oracle_backup_20100113185446`.

In addition, `sslConfig_OIDclient.sh` performs the following steps:

- Writes out `/etc/oracle-certs/oid-test-ca.pem`, the pem format encoded certificate for the Test CA created during configuration on the Oracle Internet Directory Server. This is equivalent to `pem.cert` in "Self Signed Certificates" on page 3-2.
- Adds `oid-test-ca.pem` as a trusted CA in `/etc/ldap.conf` and `/etc/openldap/ldap.conf`
- Configures `/etc/ldap.conf` to use cleartext passwords and enable SSL

On most client operating systems, the script configures the client to use the StartTLS port on the server for SSL communication. The script does not configure StartTLS if the operating system on the client is HP-UX or Solaris. These clients use the standard SSL port, 636, on the server for SSL communication.

After you have successfully executed the client configuration script, your Linux or UNIX-based client can use Oracle Internet Directory to authenticate users.

Configuring Oracle Internet Directory for Centralized Password Policies

To use Oracle Internet Directory for centralized password policies, you must disable value and state policies local to the operating system.

After you do that, users can invoke the `passwd` tool as usual to change their password. Violations of Oracle Internet Directory password value policies produce error messages in the log files beginning with `Password Policy Error`.

Disabling Value Policies Local to the Operating System

Most Linux distributions are configured by default to use the `cracklib` library to perform end-user supplied password quality validations. When using a centralized password policy enforced in Oracle Internet Directory, you might want to disable the local validations in order to avoid conflicts between the two policies.

On Oracle Enterprise Linux and Red Hat Linux, you can do this as follows:

1. Locate the following line in `/etc/pam.d/system-auth` and comment it out:

   ```
   password requisite /lib/security/$ISA/pam_cracklib.so retry=3
   ```

2. Locate all subsequent lines beginning with `password` and remove `use_authtok` from those lines.

Disabling State Policies Local to the Operating System

As mentioned previously, state policies on Linux are enforced through the password aging feature enabled by the shadow password information. The operating system parses the shadow information on each account and enforces state policies locally.

In Red Hat Enterprise Linux or Oracle Enterprise Linux, you can disable password ageing for accounts created under Oracle Internet Directory by modifying `/etc/libuser.conf` to use `-1` as the default value for `LU_SHADOWINACTIVE`, `LU_SHADOWEXPIRE`, `LU_SHADOWWARNING` in the `[userdefaults]` section of the file.

For accounts that already exist in Oracle Internet Directory, or that are to be migrated to Oracle Internet Directory, you must set `shadowmax=99999` and `shadowexpire=-1` to disable password expiration.

Switching Between SSL Authentication and Non-SSL Configurations

If you have configured non-ssl authentication, you can switch to SSL authentication as follows:

1. On the server, run the script `sslConfigure_OIDserver.sh`. Optionally, you can disable the non-ssl port by following the instructions in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.

2. Copy the `sslConfigure_OIDclient.sh` script generated on the server to the client machine and run this script as root.

If you have configured SSL authentication, you can switch to non-ssl authentication as follows:

1. On the server, run the script `config_OIDserver.sh`. Optionally, you can disable the ssl port by following the instructions in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.
2. Copy the `config_OIDclient.sh` generated on the server to the client machine and run this script as `root`.

Rerunning the Configuration Scripts

There are occasions when you might need to rerun the configuration scripts. For example, you might need to change to a different Oracle Internet Directory server. As another example, if you are using a proxy DN, rather than anonymous binds, to connect to Oracle Internet Directory, the password of the proxy user will expire at some point and need to be reset.

To rerun the scripts, proceed as follows:

1. Rerun the configuration script on the server. Execute `config_OIDserver.sh` or `sslConfig_OIDserver.sh` as the user who installed Oracle Internet Directory.
2. Restore each client, as described in "Restoring the Client" on page 3-11.
3. Rerun the script on each client. Execute the generated script `config_OIDclient.sh` or `sslConfig_OIDclient.sh` on each client machine as `root`.

Restoring the Client and Server to Their Pre-Configuration State

You can restore the computers to their original state.

Restoring the Client

If necessary, you can restore your client computers to the state they were in before you ran `config_OIDclient.sh` or `sslConfig_OIDclient.sh`. To do so, locate directories under `/etc` with names of the form `oracle_backup_`*`time_stamp`*. For example, a backup directory created 18:54:46 on Jan. 13 2008 would have the name `/etc/oracle_backup_20080113185446`. If there is more than one backup directory, in most cases, you need to use the backup files in the earliest backup directory.

To restore a client to its pre-configuration state, run the script `resetClient.sh`. You can find this script on the server at `$ORACLE_HOME/oas4os/bin`. Copy it to the client and run it as `root`. The script prompts you for the path to the configuration files that were saved when you ran the configuration script.

Restoring the Server

There is nothing to restore on the server. See the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* if you want to stop the Oracle Internet Directory server or to disable the SSL or non-SSL port.

4

Migrating Entries to Oracle Internet Directory

This chapter contains the following topics:

Migrating Entries

Before migrating entries from NIS, files, or another LDAP directory, perform the following tasks:

- Tune your Oracle Internet Directory database.

  See Also:

  - The basic tuning steps listed in the Oracle Internet Directory chapter of *Oracle Fusion Middleware Performance and Tuning Guide*.
  - *Oracle Internet Directory Tuning and Configuration, a Quick Reference Guide* at `http://www.oracle.com/technology`.

- Take a cold backup of the Oracle Internet Directory database in case you need to restore it.
- Ensure that, in the event that Oracle Internet Directory becomes unavailable, the administrator will still be able to log in as `root`. Specifically:
  - Keep a local `root` account in your `/etc/passwd` and `/etc/shadow` files.
  - Do not modify the `passwd` or `shadow` precedence in `nsswitch.conf`. The configuration script sets them to:

    ```
    passwd: files ldap
    shadow: files ldap
    ```

- Ensure that the default password hashing algorithm in your environment is DES or MD5 crypt. If it is not, change it. Then require all users to modify their passwords, so that passwords are stored in a format supported by Oracle Internet Directory.

Before you load LDIF files into Oracle Internet Directory, you can check the files for schema and data consistency violations using the `check` feature of the `bulkload` tool. The syntax is:

```
$ORACLE_HOME/ldap/bin/bulkload connect=oid-db check=true file=ldif_file
```

> **Note:** Exercise security precautions in your handling of files that contain sensitive information.

This section contains the following topics:

- Migrating from NIS to Oracle Internet Directory
- Migrating from Operating System Files to Oracle Internet Directory
- Migrating from Another LDAP Directory to Oracle Internet Directory

Migrating from NIS to Oracle Internet Directory

Migrate entries as follows.

AIX 5.3

If you did not migrate all your entries to LDAP before running the client configuration script, proceed as described in the AIX 5.3 section of "Configuring Oracle Authentication Services for Operating Systems on the Client" on page 3-6.

Other Platforms

The steps for migrating entries from NIS to Oracle Internet Directory are as follows:

1. Run the LDAP migration scripts, described in "Get NIS Migration Tools" on page 2-3, on your NIS master files. This will generate LDIF files containing the entries.
2. For compatibility with a variety of clients, as well as with the `system-config-users` tool, ensure that the entries include all the required attributes shown in the following example. (Substitute the user's password for *password*.)

```
dn: uid=jueno,ou=People,dc=us,dc=example,dc=com
uid: jueno
homedirectory: /home/jueno
loginshell: /bin/tcsh
uidnumber: 506
gidnumber: 506
cn: juri ueno
objectclass: posixAccount
objectclass: shadowAccount
objectclass: account
objectclass: top
userpassword: password
shadowwarning: -1
shadowmax: 99999
shadowlastchange: 13916
shadowexpire: -1
shadowmin: 0
shadowinactive: -1
gecos: jueno
```

The `shadowAccount` objectclass and attributes are typically missing in user entries migrated from an HP-UX server.

3. Use the `ldapadd` client tool shipped with Oracle Internet Directory to load the LDIF entries into Oracle Internet Directory. Use a command line of the form:

```
ldapadd -p port -h host -D binddn -q -v -f ldif_file
```

Note:

- If you are using the same naming context created during installation, these scripts will generate parts of the DIT (Directory Information Tree) that already exist. This will cause `ldapadd` failures because you are attempting to add an existing entry. You can avoid these failures by specifying the `-c` option to continue upon encountering such errors.
- The `binddn` you use must be the directory administrator DN so that you have the proper privileges when performing these additions.

Migrating from Operating System Files to Oracle Internet Directory

Migrating from operating system files is basically the same as migrating from NIS, except that you might have different versions of your configuration files on different machines. If you have multiple versions, run the migration scripts on each version and combine the LDIF files. You must resolve conflicts manually, using a text editor. Each user must have a unique user name and `UID`, and each group must have a unique group name and `GID`.

Migrating from Another LDAP Directory to Oracle Internet Directory

You can migrate entries from a third-party, LDAP-compliant directory to Oracle Internet Directory.

Note: This section describes how to do a one-time migration of data from an LDAP-compliant source directory to Oracle Internet Directory. If you are planning to set up ongoing synchronization between a source directory and Oracle Internet Directory by using Oracle Directory Integration Platform, refer to the *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Integration Platform*.

Migration of entries from a third-party source directory to Oracle Internet Directory occurs in two phases: schema migration and data migration.

Schema Migration

The steps for migrating schema are as follows:

1. Analyze the schema difference between the directories by running the `schemasync` tool. The syntax is:

```
$ORACLE_HOME/bin/schemasync -srchost srchost -srcport srcport -srcdn binddn \
                            -srcpwd bindpwd -dsthost oidhost -dstport oidport \
```

```
                                -dstdn oiddn -dstpwd oidpwd
```

where *srchost* and *srcport* are the connection details of the source directory and *srcdn* and *srcpwd* are the credentials to connect to the source directory.

> **See Also:** The command reference for `schemasync` in the Oracle Directory Integration Platform Tools chapter of the *Oracle Fusion Middleware Reference for Oracle Identity Management*.

The command produces four output files that list differences between the source directory and Oracle Internet Directory schema. They are:

- `$ORACLE_HOME/ldap/odi/log/attributetypes.log`–difference in the schema definition of the common attributes between the source directory and Oracle Internet Directory.
- `$ORACLE_HOME/ldap/odi/log/objectclasss.log`–difference in the schema definition of the common object classes between the source directory and Oracle Internet Directory
- `$ORACLE_HOME/ldap/odi/data/attributetypes.ldif`–attributes that are available only in the source directory and not in Oracle Internet Directory.
- `$ORACLE_HOME/ldap/odi/data/objectclasses.ldif`–object classes that are available only in the source directory and not in Oracle Internet Directory.

2. If necessary, extend the schema elements in Oracle Internet Directory.

   a. Based on the analyses in Step 1, determine what new schema elements you must load onto Oracle Internet Directory. Modify the files `attributetypes.ldif` and `objectclasses.ldif` (from step) 1 to have only the attributes and object classes that you must load. Name the modified files `modified_attributetypes.ldif` and `modified_objectclasses.ldif`.

      For example, assume that the objectclass of the user entry in the third-party directory is `inetorgperson, organizationalperson, person, srcuser` and the objectclass of user entry in Oracle Internet Directory is `inetorgperson`,`organizationalperson`, `person`, `orcluser`. In Step 1, if the objectclass definitions of `inetorgperson`, `organizationalperson`, and `person` are different between Oracle Internet Directory and the third-party directory, the difference will be written to the `objectclasses.log` files. After looking at the file, you might decide to make the required changes in the objectclass definitions of Oracle Internet Directory. Since `srcuser` is a third-party directory specific objectclass, the objectclass definition will be in the `objectclasses.ldif` file. Modify the `objectclass.ldif` file to contain the objectclass definition and rename it `modified_objectclasses.ldif`. Modify the `attributes.ldif` file to contain the definitions of the attributes required for the objectclasses in `objectclasses.ldif`.

   b. Upload the required schema using the `ldapmodify` command as follows:

```
ldapmodify -h oidhost -p oidport -D 'cn=orcladmin' -q \
           -f modified_attributetypes.ldif
ldapmodify -h oidhost -p oidport -D 'cn=orcladmin' -q \
           -f modified_objectclasses.ldif
```

Data Migration

Migration of data is more complicated because you must include some entries and exclude others. Even in the entries that are included, you might want to include only specific attributes. In general, user and group are migrated. The attributes representing access control definitions, password policy-related attributes, and other operational attributes such as `createtimestamp`, `modifytimestamp`, `creatorsname`, `modifiersname`, `entrydn`, `numsubordinates`, `parentid`, `entryid`, and `nsuniqueid` are excluded. You might want to include `userpassword` as an attribute to be migrated. Do so only if both the directories support the same kind of encryption or hashing techniques.

You can get the exact data to be migrated by filtering the data either while exporting it from the source directory (Step 1) or as a separate step (Step 2).

1. Export the data from the source directory into LDIF file format, using the appropriate LDAP tool on your system, and analyze it. See the documentation for your directory server to determine what command to use. If you filter and export only the required LDAP entries with only the required attributes during the export operation, proceed to Step 3. Otherwise, filter it in Step 2.
2. If you did not filter out the entries and attributes not to be migrated in Step 1, remove them in this step by using `syncProfileBootstrap`.

   The `syncProfileBootstrap` tool filters the entries based on the configuration and also supports mapping and transformation of attributes. You specify the configuration of filtering, mapping and transformation in the mapfile. Sample mapfiles are provided in Appendix C.

   a. If you are migrating entries other than user and group from source directories, update the mapfile accordingly.

   b. Make a copy of the sample file `$ORACLE_HOME/ldap/odi/samples/migrate.properties` and name it `migrate.properties`.

      See Also: The command reference for `syncProfileBootstrap` in the Oracle Directory Integration Platform Tools chapter of the *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Integration Platform* for documentation of the parameters used in the properties file.

   c. In the properties file, you must specify the name of the LDIF file containing the entries to be migrated as source file and a name for the file that is to be generated by `syncProfileBootstrap` as the destination file name. The containers to be included/excluded and the attributes to be included/excluded are specified in the mapfile parameter of the properties file. Note: This mapfile can be used only for migration purposes and is not supported for synchronization. A sample properties file is shown in Appendix B.

   d. Generate a new LDIF file in the format required by Oracle Internet Directory by running the command:

      ```
      syncProfileBootstrap -file testmigrate.properties
      ```

3. Optionally, you can use an Oracle Internet Directory plug-in to augment entries. See "Setting up a Plug-in to Augment Active Directory Entries for Linux Authentication" on page 5-1. This method has been shown to work for iPlanet (Sun Java System Directory Server) 5.2 as well as Active Directory.

4. Get the filtered LDIF file resulting from Step 1 or Step2 and use either `$ORACLE_HOME/bin/ldapadd` or `$ORACLE_HOME/ldap/bin/bulkload` to add the data to Oracle Internet Directory. If you have more than a few thousand entries, use `bulkload` in preference to `ldapadd`.

   a. The syntax for `ldapadd` is:

      ```
      ldapadd -h oidhost -p oidport -d oiddn -q -f ldif_file
      ```

      If you use `ldapadd`, once the data is successfully added, update the Oracle Internet Directory database statistics using `$ORACLE_HOME/ldap/admin/oidstats.sql`. Log in to the Oracle Internet Directory database as the `ODS` database user and execute this SQL script.

      See Also: The `oidstats.sql` command reference in the Oracle Internet Directory Database Tools chapter of the *Oracle Fusion Middleware Reference for Oracle Identity Management*.

   b. If you decide to use `bulkload`, then proceed to Step 5

5. Bulk load LDIF data into Oracle Internet Directory. In the following steps, the file `/home/jdoe/migrationdata.ldif` is the filtered LDIF file.

   a. Stop all Oracle Internet Directory processes by executing the command:

      ```
      $ORACLE_HOME/opmn/bin/opmnctl stopproc ias-component=OID
      ```

   b. Take a cold backup of the database if you have not done so already.

   c. Use `bulkload` to check for schema errors, duplicate entries and other errors and to generate intermediate files for a subsequent data load. The syntax is:

      ```
      $ORACLE_HOME/ldap/bin/bulkload connect=oid-db check=true generate=true
      file=/home/jdoe/migrationdata.ldif
      ```

      When you specify both `check` and `generate` options, `bulkload` checks the entries for schema compliance and duplicates and generates the intermediate files that are used during the load phase in the `$ORACLE_HOME/ldap/load` directory.

      If there are any `check`-related errors, `bulkload` reports them on the screen. The tool logs entries in `$ORACLE_HOME/ldap/log/duplicateDN.log` and logs schema-related violations in `$ORACLE_HOME/ldap/log/bulkload.log`. It writes entries that have errors to `$ORACLE_HOME/ldap/load/badentry.ldif`.

      If `bulkload` detects any errors in the entries, you might have to fix the entries or schema or both in Oracle Internet Directory. After you fix the problems, re-run the `bulkload` command. Repeat this until there are no errors or the errors reported are acceptable. For example, if you encounter some schema check error for a small number of entries, you can choose to `ldapadd` them from `badentry.ldif` later by fixing the entries or schema in Oracle Internet Directory.

      When you use the `check` and `generate` options, `bulkload` generates the intermediate files for entries that had no `check`-related errors. The `generate` occurs even if there are erroneous entries. For example, if the LDIF file has 100 entries and 10 entries have `check` errors, `bulkload` generates the intermediate files for 90 good entries.

d. Use `bulkload` to load the data, recreate all indexes and generate db statistics. Execute the command:

```
$ORACLE_HOME/ldap/bin/bulkload connect=oid-db load=true
file=/home/jdoe/migrationdata.ldif
```

This command accomplishes three things: loading data from `$ORACLE_HOME/ldap/load` directory into the database using SQL*Loader, creating indexes, and generating database statistics.

If it detects an error, `bulkload` indicates the error on the screen. If it reports an error during loading of data, you must restore the database from the backup taken in Step b and then repeat the `bulkload load=true` command. If `bulkload` reports an error during indexing, use the following command to recreate all indexes:

```
bulkload connect=oid-db index=true
```

If `bulkload` reports an error during database statistics generation, you can use the following command to generate the statistics:

```
$ORACLE_HOME/ldap/admin/oidstats.sql
```

e. Start all Oracle Internet Directory processes by executing the command:

```
$ORACLE_HOME/opmn/bin/opmnctl startproc ias-component=OID
```

Setting Access Control on User Entry Attributes

To protect sensitive user attributes from unauthorized modification, set an access control item. Type:

```
ldapmodify -h oidhost -p oidport -D 'cn=orcladmin' -q -f aci.ldif
```

where `aci.ldif` looks like this:

```
dn:
changetype: modify
add: orclaci
orclaci: access to attr=(uidnumber,gidnumber,homedirectory,uid)
 by group="cn=OracleUserSecurityAdmins,cn=Groups,cn=OracleContext"
 (search,read,write,compare) by group="cn=directoryadmingroup,cn=oracle internet
 directory" (search,read,write,compare) by * (search,compare,nowrite,nocompare)
```

Using Custom Attributes in Oracle Internet Directory

You can search for an attribute in Oracle Internet Directory only if the attribute is indexed. By default, standard attributes of the user and group entries are indexed. If you use a custom attribute, you can index it by using the `catalog` command. For example, if you migrate automount data to be used by automount programs such as `amd` or `autofs`, index the automountKey attribute by using the `catalog` command, as follows:

```
catalog connect="connect_str" add="TRUE" attribute="automountKey"
```

Note: If you attempt to perform a search with a non-indexed attribute specified as a required attribute, the server will return a "Function not implemented. DSA unwilling to perform" error. See "Create and Index New Custom Attributes (Optional)" on page 2-4.

Some attributes, such as `uid` and user name, must be unique. Oracle Internet Directory will enforce uniqueness if you create a uniqueness constraint for that attribute. For more information see the chapter "Attribute Uniqueness in the Directory" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.

Note: The attribute uniqueness feature works on indexed attributes only.

Migrating SUDO

You can migrate entries from `/etc/sudoers` into Oracle Internet Directory using the `sudo` package you downloaded from:

`http://www.gratisoft.us/sudo`

This `sudo` package includes `sudo` software and the scripts to convert `sudo` data to LDAP data (LDIF). Read the documentation included in the package before you begin the migration process.

Note: After migrating `sudo`, run some security tests to ensure that your `sudo` policy is being enforced correctly.

Migrating SUDO Entries to Oracle Internet Directory on the Server

To move the contents of your `sudoers` file to Oracle Internet Directory, perform these steps:

1. Add a `Sudoers` container to Oracle Internet Directory using the command:

   ```
   ldapadd -h oid_hostname -p port -D cn=orcladmin \
           -q -f sudocontainer.ldif
   ```

   where `sudocontainer.ldif` looks like this:

   ```
   dn:ou=Sudoers,dc=us,dc=example,dc=com
   objectclass:top
   objectclass:organizationUnit
   ou=sudoers
   ```

2. Using the `/etc/sudoers` file from your existing `sudo` client, generate an LDIF file by running the conversion script supplied with the `sudo` package you downloaded. Follow the instructions at the download site. Please see the `sudo` package documentation for known limitations.
3. View the resulting LDIF file in a text editor and correct any obvious errors.
4. Add the contents of the ldif file to Oracle Internet Directory by using the command:

   ```
   ldapadd -h oid_hostname -p port -D cn=orcladmin \
           -q -f sudoers.ldif
   ```

where `sudoers.ldif` is the file generated from your `/etc/sudoers` file.

If `ldapadd` encounters an error, it will stop and report the error. Correct the error and repeat the command until it runs successfully and adds all the entries.

Once you have migrated your `sudo` entries to Oracle Internet Directory, you must use LDAP tools to modify them. See the documentation in the downloaded `sudo` package for information about LDAP browsers you can use for editing `sudo` entries.

Configuring a Client to Use LDAP for SUDO Information

On most client operating systems, you can configure `sudo` with the native LDAP and SSL libraries for that operating system. On a few operating systems, you must use OpenLDAP and OpenSSL.

When you configure `sudo`, the `make install` step will install a new copy of `/etc/ldap.conf`. If you already have an `ldap.conf` file, you must make a copy before you configure `sudo` or the file will be overwritten. Once you have performed a `make install`, copy that file back to `/etc/ldap.conf`.

SuSE 10 Client

1. Download, build, and install the OpenLDAP and OpenSSL packages.
2. If you already have the file `/etc/ldap.conf`, make a copy. For example

   ```
   cp /etc/ldap.conf /etc/ldap.conf.save
   ```

3. In the directory where you downloaded the `sudo` package, build `sudo` by typing the following commands:

   ```
   ./configure --with-ldap-type=openldap --with-pam --enable-ssl
   make all
   make compile
   make install
   ```

4. If you made a copy of your `ldap.conf` file, copy it back to its original name. For example:

   ```
   cp /etc/ldap.conf.save /etc/ldap.conf
   ```

5. If there is no `libpam.so` link, make one by typing:

   ```
   cd /usr/lib
   ln -s libpam.so.0 libpam.so
   ```

6. Edit `/etc/pam.d/sudo`. Add the following line above the first `auth` line:

   ```
   auth     sufficient     /lib/security/pam_ldap.so debug
   ```

7. Modify `/etc/ldap.conf` so that `sudoers_base` points to the base sudoers container. For example:

   ```
   sudoers_base  ou=Sudoers,dc=us,dc=example,dc=com
   ```

   If you want to configure `ssl` for `sudo` you must specify `startTLS` in `ldap.conf` since the current `sudo` implement does not support SSL only. For example:

   ```
   ssl startTLS
   ```

Solaris 9, Solaris 10, HP-UX 11.23 or AIX 5.3 Client

On these operating systems, the native LDAP client does not support StartTLS. If you plan to use `sudo` with SSL, download, build, and install the OpenLDAP and OpenSSL packages and build `sudo` as described for SuSE 10 clients. Once you have completed those steps, add the following lines to `/etc/ldap.conf` to specify the target LDAP host and port and the SSL certificate authority certificate path and certificate filename:

```
host ldap_host
port ldap_port
tls_cacertdir  /etc/ca_certs_dir
tls_cacertfile  /etc/ca_cert_file
```

If you plan to use `sudo` in non-SSL mode only, build it using the native LDAP client libraries, as described for other clients.

AIX 5.3 Client

To build sudo with LDAP enabled on AIX 5.3, proceed as follows, where `base_dir` is the directory where you install `openssl`, `openldap`, and `sudo`.

1. Set environment variables.

   ```
   export CFLAGS="-I$base_dir/include -I/usr/include" \
   export CPPFLAGS="-I$base_dir/include -I/usr/include" \
   export LDFLAGS="-L$base_dir/lib -L/usr/lib"
   export CC=/usr/local/bin/gcc
   ```

2. Build `openssl`.

   ```
   cd $build_dir/openssl-0.9.8g
   ./Configure aix-cc shared threads -D_REENTRANT --prefix=$base_dir
   make
   make install
   ```

3. Build `openldap`

   ```
   cd $build_dir/openldap-2.3.39
   ./configure \
   --prefix=$base_dir \
   --enable-slapd=no \
   --enable-bdb=no \
   --enable-static=no \
   --enable-shared=yes
   ```

4. Edit the `Makefile`, adding the TLS_LIBS parameter, as follows:

   ```
   TLS_LIBS = -lssl -lcrypto -lldap
   ```

5. Run `make`.

   ```
   make depend MKDEP=$PWD/build/mkdep.aix
   make
   ```

   If you encounter errors when you run `make`, update this parameter in the `Makefile` and try again:

   ```
   LUTIL_LIBS=$(LDAP_LIBDIR)/libldap/.libs/libldap.a
   ```

6. After make completes successfully, type:

   ```
   make install
   ```

7. Build `sudo`

```
cd $build_dir/sudo1.6.9p15
./configure [--with-pam | --with-aixauth] --with-ldap=$base_dir/lib
--with-prefix=$base_dir
make
make install
```

If you encounter linking errors, determine which objects are missing from the link lines in your Makefile, add those objects, then try again.

Other Clients

1. If the `sudo` binary you are using was not built using the `--with-ldap` option, then rebuild the `sudo` command using the `--with-ldap` option, as described in the documentation in the downloaded `sudo` package. Before rebuilding `sudo`, save a copy of `/etc/ldap.conf` to a different name. Be sure to check the documentation and the README files for other options you might need to use. For example, you might have to specify your library and header location or a different configuration file if they are non-standard. You might also have to modify the Makefile by adding an `-lldif` flag to `SUDO_LIBS` if you are using an SDK other than OpenLDAP. Once you have rebuilt `sudo`, copy your saved `ldap.conf` file back to its original name.
2. Modify `/etc/ldap.conf` so that `sudoers_base` points to the base sudoers container you created in Server Step 1. For example:

   ```
   sudoers_base  ou=Sudoers,dc=us,dc=example,dc=com
   ```

   If you want to configure SSL for `sudo` you must specify `startTLS` in `ldap.conf` because the current `sudo` implementation does not support SSL only. For example:

   ```
   ssl startTLS
   ```

   Optionally, enable `sudo` debugging by adding the following line to `/etc/ldap.conf`:

   ```
   Sudoers_debug 2
   ```

3. Prevent `sudo` from using the `/etc/sudoers` file by adding the `ignore_local_sudoers` suboption to the `sudoers` defaults. You do this by running this command:

   ```
   ldapmodify -h oid_hostname -p port -D cn=orcladmin \
          -q -f ignore_local_sudoers.ldif
   ```

   where `ignore_local_sudoers.ldif` looks like this:

   ```
   dn:cn=defaults, ou=Sudoers, dc=us, dc=example, dc=com
   changetype:modify
   add: sudooption
   sudooption: ignore_local_sudoers
   ```

Reconfiguring a Client to Use /etc/sudoers

If you have configured a client computer to use LDAP for `sudo`, you can reconfigure it to use the `sudoers` file again by commenting out the line that begins with `sudoers_base` in `/etc/ldap.conf`.

Setting Access Control on SUDO Attributes

To protect sensitive `sudo` attributes from unauthorized modification, set an access control item. Type:

```
ldapmodify -h oidhost -p oidport -D 'cn=orcladmin' -q -f aci.ldif
```

where `aci.ldif` looks like this:

```
dn:
changetype: modify
add: orclaci
orclaci: access to
 attr=(sudoUser,sudoHost,sudoCommand,sudoRunAs,sudoOption,sudoRole)
 by group="cn=OracleUserSecurityAdmins,cn=Groups,cn=OracleContext"
 (search,read,write,compare) by group="cn=directoryadmingroup,cn=oracle
 internet directory" (search,read,write,compare) by * (none)
```

5

Configuring Active Directory Integration

If you have users in Active Directory, and you want to use the credentials stored in Active Directory for Linux or UNIX authentication, you can configure integration with Active Directory. Setting up integration with Active Directory requires several steps:

- You use the Oracle Directory Integration Platform to synchronize user and group entries to Oracle Internet Directory when they are added to or changed in Active Directory.
- You use an Oracle Internet Directory plug-in to add required attributes to the user and group entries in Oracle Internet Directory after they are synchronized from Active Directory to Oracle Internet Directory.
- You use another Oracle Internet Directory plug-in to enable Active Directory authentication of Linux or UNIX users.
- To secure communication, you configure SSL between Oracle Directory Integration Platform and Active Directory and between Oracle Directory Integration Platform and Oracle Internet Directory.

> **Note:** After you have synchronized users from Active Directory into Oracle Internet Directory, you can only change passwords through Active Directory. You must change the password in the Active Directory user entry, not the Oracle Internet Directory entry. If you change the password in Oracle Internet Directory or by using the `passwd` command, the change will appear to be successful but will not be propagated to the Active Directory entry. The password in the Active Directory user entry will remain in effect.

This chapter contains the following sections:

Setting up a Plug-in to Augment Active Directory Entries for Linux Authentication

User entries in Active Directory do not include key information required for Linux authentication. Therefore, when you synchronize users from Active Directory into Oracle Internet Directory by using the Active Directory connector of Oracle Directory Integration Platform, you must augment those user entries with the required

information. To facilitate this, the product includes a PL/SQL plug-in that can be enabled on Oracle Internet Directory.

Enable the plug-in as follows:

1. Use a text editor to make the following changes to `$ORACLE_HOME/ldap/admin/posixattr_when_add.pls`:
   - In line 71, replace the value of `v_homeDirectory` with the desired home directory.
   - In line 72, replace the value of `v_loginShell` with the desired login shell.
   - In line 73, replace the value of `v_gidNumber` with the GID number of the users
2. Load the plug-in package into the database by typing:

   ```
   sqlplus ods/odspwd@$ORACLE_HOME/ldap/admin/posixattr_when_add.pls
   ```

   where *odspwd* is the password of the `ODS` user.
3. Use a text editor to make the following change in `$ORACLE_HOME/ldap/admin/posixattr_when_add.ldif`: Replace the value of `orclpluginsubscriberdnlist` with your realm's DN.
4. Add the plug-in to Oracle Internet Directory by running the following command:

   ```
   ldapadd  -h host -p port -D cn=orcladmin -q \
            -f $ORACLE_HOME/ldap/admin/posixattr_when_add.ldif
   ```

Configuring Oracle Directory Integration Platform

Oracle Directory Integration Platform is documented in the *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Integration Platform*. The following procedure refers to that document in several places.

To enable Oracle Directory Integration Platform for Active Directory integration with Oracle Authentication Services for Operating Systems, perform these steps:

1. Verify the synchronization requirements, as described in "Verifying Synchronization Requirements," under "Configuring Synchronization with a Third-Party Directory," in Chapter 18 of the *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Integration Platform*.
2. Create a synchronization profile by running `expressSyncSetup`, as described in the section "Creating Import and Export Synchronization Profiles Using expressSyncSetup" in the chapter entitled "Creating Synchronization Profiles with Express Configuration" in*Oracle Fusion Middleware Administrator's Guide for Oracle Directory Integration Platform*.
3. Edit the profiles resulting from the express configuration. To understand mapping rules, see: "Configuring Mapping Rules," in Chapter 6 of the *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Integration Platform*.

   Make the following changes:

   a. Change the domain rules to point to `ou=People` under the realm `DN: ou=People,<realm DN>` in Oracle Internet Directory.
   b. Provide a DN mapping rule: `uid=%,ou=People,<realm DN>`
   c. Comment out this line:

      ```
      userPrincipalName: : :user:uid: :inetorgperson:userPrincipalName
      ```

d. Uncomment this line:

```
#sAMAccountName: : :user:uid: :inetorgperson
```

See the sample synchronization profile in Appendix D. The customizations are shown in **boldface**.

4. Continue with Steps 2-5 of "Creating Synchronization Profiles with Express Configuration," under "Configuring Synchronization with a Third-Party Directory," in Chapter 18 of the *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Integration Platform*.

Configuring External Authentication Plug-ins

You must configure external authentication plug-ins for authenticating users synchronized from AD. The procedure for doing this is documented in the "Configuring External Authentication Plug-ins" section of *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Integration Platform*.

6

Managing Oracle Authentication Services for Operating Systems

This chapter contains the following topics:

Creating Home Directories

On Linux systems, you do not have to create each user's home directory when you migrate or add that user to Oracle Internet Directory. The client configuration script that you ran on each client computer enabled the creation of each user's home directory on first login. On operating systems other than Linux, however, you must manually create user home directories.

Managing Users and Groups with Platform-Specific Tools

In addition to the Oracle Internet Directory tools ODSM, LDAP commands, and bulk tools, you can use some platform-specific tools to manage users and groups after you have configured Oracle Authentication Services for Operating Systems.

libuser Tools

If your client has the `libuser` library and you have configured it to use Oracle Internet Directory, you can use `system-config-users` or `luseradd` to add users. When you invoke one of the `libuser` commands, it will prompt you for the password for logging into Oracle Internet Directory. See your operating system documentation for more information about `system-config-users`.

Note:

- If you use `system-config-users` or other tools in the `libuser` package to add passwords or entries containing passwords, Oracle Internet Directory cannot enforce its password policies on those passwords. The reason is that the `libuser` tools generate a hashed password before sending it to Oracle Internet Directory, so Oracle Internet Directory cannot determine whether the password meets policy criteria or not.
- The `system-config-users` tool requires that you configure your client and server for SSL.
- Before using `system-config-users`, ensure that the user entries have all the required attributes shown in "Migrating from NIS to Oracle Internet Directory" on page 4-2.The tool may report errors if fields are missing.
- You cannot use the non-`libuser` commands `useradd`, `userdel`, `groupadd`, or `groupdel` for user or group administrative tasks.

AIX-Specific Tools

On AIX, you can use the following tools to manager users after you have configured Oracle Authentication Services for Operating Systems.

Table 6–1 AIX User and Group Management Tools

Action	Command
Add User	`mkuser -R LDAP`
Add Group	`mkgroup -R LDAP`
Delete User	`rmuser -R LDAP`
Delete Group	`rmgroup -R LDAP`
Change Password	`passwd -R LDAP`
List User	`luser -R LDAP`

Managing Oracle Internet Directory with Oracle Directory Services Manager and Command-Line Utilities

The *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* contains information about managing Oracle Internet Directory. See the "Directory Administration and Monitoring Tools" chapter for information on Oracle Directory Services Manager. See the "Process Management" chapter for information on starting and stopping Oracle Internet Directory. See the Using Bulk Tools chapter for information on the bulk tools.

The *Oracle Fusion Middleware Reference for Oracle Identity Management* provides the syntax for Oracle Internet Directory command-line tools, including the bulk tools and LDAP tools.

Please see the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for information about modifying or deleting users and groups.

Testing Whether a User Has Been Added

You can test whether a user has been added by using the following command:

```
ldapsearch -D cn=orcladmin -q -b 'searchbase' -s -sub '(uid=username)'
```

where *searchbase* is the realm, for example, dc=us,dc=example,dc=com.You can also test the account by logging in as the user. For example, you can log in to one client from another by using ssh. For example:

```
ssh -l username hostname
```

Once you are logged in, type:

```
id
```

to confirm that you are logged in as the correct user.

Changing a User's Password by Using ldapmodify

To change a user's password, you use the command:

```
ldapmodify -p port -h host -D binddn -q -v -f passwd_file
```

where passwd_file looks like this:

```
dn: userDN
changetype: modify
replace: userpassword
userpassword: new_password
```

Note:

- After you have used passwd_file, delete it or remove the cleartext password.
- Users can change their own passwords by using the passwd command.

Adding a User by Using ldapadd

To add users and groups from the command line you use a command line such as:

```
ldapadd -p port -h host -D binddn -q -v -f ldif_file
```

where ldif_file contains the information about the entry you are adding in LDIF format.

In the following ldif_file example, we create a user called jueno. The user is created in the user container ou=People,dc=us,dc=example,dc=com under the realm dc=us,dc=example,dc=com. To create a user, you must provide the following attributes: uid, homedirectory, loginshell, uidnumber, gidnumber, cn, objectclass, and userpassword (in cleartext). For compatibility with a variety of clients and with the system-config-users management tool, use all the object classes shown in the example.

```
dn: uid=jueno,ou=People,dc=us,dc=example,dc=com
uid: jueno
homedirectory: /home/jueno
loginshell: /bin/tcsh
uidnumber: 506
gidnumber: 506
```

```
cn: juri ueno
objectclass: posixAccount
objectclass: shadowAccount
objectclass: account
objectclass: top
userpassword: password
shadowwarning: -1
shadowmax: 99999
shadowlastchange: 13916
shadowexpire: -1
shadowmin: 0
shadowinactive: -1
gecos: jueno
```

After you have used the LDIF file, delete it or remove the cleartext password.

Adding a Group by Using ldapadd

To add groups from the command line, you use the same command line you use to add users. That is:

```
ldapadd -p port -h host -D binddn -q -v -f ldif_file
```

In the following example, we create a group called `kobukuro` with group ID 505. The group is created in the group container `ou=Group,dc=us,dc=example,dc=com` in the realm `dc=us,dc=example,dc=com`. We also add a member, `juero`, at the same time, by specifying the memberuid and the value. The LDIF file looks like this:

```
dn: cn=kobukuro,ou=Group,dc=us,dc=example,dc=com
cn: kobukuro
gidnumber: 505
objectclass: posixGroup
objectclass: groupOfUniqueNames
objectclass: top
memberuid: jueno
```

Adding a member to the group at the same time is optional.

Managing Password Policies

See the Managing Password Policies chapter in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.

Note:

- If you use `system-config-users` or other tools in the `libuser` package to add passwords or entries containing passwords, Oracle Internet Directory cannot enforce its password policies on those passwords. The reason is that the `libuser` tools generate a hashed password before sending it to Oracle Internet Directory, so Oracle Internet Directory cannot determine whether the password meets policy criteria or not.
- On AIX, the `passwd` utility does not display password policy errors. Instead, it displays:

  ```
  3004-604 Your entry does not match the old password.
  ```

7

Restricting User Logins

You can use Oracle Authentication Services for Operating Systems to restrict which users can log into each host. For example, you can enforce rules like these:

- `user1` can only log into `hostA`.
- `user2` can only log into `hostB`.
- `user3` can log into `hostA`, `hostB`, and `hostC`.

To enforce rules like these, you must perform some configure tasks on both the Oracle Internet Directory server and all the client hosts where you want to restrict access. The setup procedure on the Oracle Internet Directory server is the same, regardless of the operating system. The setup instructions on the client host are operating system-specific.

This chapter includes the following topics:

Oracle Internet Directory Server Setup

Before you begin, ensure that Oracle Internet Directory is running and that Oracle Authentication Services for Operating Systems is working correctly. To configure the rules example at the beginning of this chapter, perform the following steps:

1. Index the `host` attribute so that it is searchable, by using the `catalog` command. Type:

   ```
   catalog connect=connect string add=true attribute=host
   ```

2. Restart the Oracle Internet Directory server:

   ```
   $ORACLE_HOME/opmn/bin/opmnctl restartproc ias-component=OID
   ```

3. Modify the entry for `user1`, adding the `host` attribute with value `hostA`:

   ```
   $ORACLE_HOME/bin/ldapmodify -D cn=orcladmin -q -h OID_host -p OID_port <<E
   dn: uid=User1,ou=people,dc=us,dc=example,dc=com
   changetype: modify
   add: host
   host: hostA
   E
   ```

4. Modify the entry for `user2`, adding the `host` attribute with value `hostB`:

```
$ORACLE_HOME/bin/ldapmodify -D cn=orcladmin -q -h OID_host -p OID_port <<E
dn: uid=user2,ou=people,dc=us,dc=example,dc=com
changetype: modify
add: host
host: hostB
E
```

5. Modify the entry for `user3`, adding the `host` attribute with value `ALL`:

```
$ORACLE_HOME/bin/ldapmodify -D cn=orcladmin -q -h OID_host -p OID_port <<E
dn: uid=user3,ou=people,dc=us,dc=example,dc=com
changetype: modify
add: host
host: ALL
E
```

Solaris 9 and 10 Client Setup

To configure the rules example at the beginning of this chapter on Solaris 9 and 10 clients, perform the following steps.

1. On Solaris 9 clients, install operating system patch `112960-61` or later.
2. Configure SSL authentication between Oracle Internet Directory and the Solaris clients and verity that it is working correct.
3. On each client, make a backup copy of `sslConfig_OIDclient.sh`.
4. On each client, open `sslConfig_OIDclient.sh` in an editor and locate the following section:

```
/usr/sbin/ldapclient manual \
 -a defaultServerList=${oidServerHost} \
 -a defaultSearchBase=${realm} \
 -a authenticationMethod=none \
 -a credentialLevel=anonymous \
 -a serviceAuthenticationMethod=pam_ldap:tls:simple \
 -a serviceSearchDescriptor=passwd:ou=people,${realm}?one \
 -a serviceAuthenticationMethod=passwd-cmd:tls:simple \
 -a serviceSearchDescriptor=group:ou=group,${realm}?one
```

   Locate the two instances of `ldapclient` commands like this, one for Solaris 10 and the other for Solaris 9. Identify the appropriate instance for your operating system version and edit that instance.

5. Make the following changes on `hostA`:

```
/usr/sbin/ldapclient manual \
 -a defaultServerList=${oidServerHost} \
 -a defaultSearchBase=${realm} \
 -a authenticationMethod=none \
 -a credentialLevel=anonymous \
 -a serviceAuthenticationMethod=pam_ldap:tls:simple \
 -a
serviceSearchDescriptor=passwd:ou=people,${realm}?one?(|(host=hostA)(host=ALL))
\
 -a serviceAuthenticationMethod=passwd-cmd:tls:simple \
 -a serviceSearchDescriptor=shadow:ou=people,${realm}?sub \
 -a serviceSearchDescriptor=group:ou=group,${realm}?one
```

6. Make the following changes on `hostB`:

```
/usr/sbin/ldapclient manual \
 -a defaultServerList=${oidServerHost} \
 -a defaultSearchBase=${realm} \
 -a authenticationMethod=none \
 -a credentialLevel=anonymous \
 -a serviceAuthenticationMethod=pam_ldap:tls:simple \
 -a
serviceSearchDescriptor=passwd:ou=people,${realm}?one?(|(host=hostB)(host=ALL))
\
 -a serviceAuthenticationMethod=passwd-cmd:tls:simple \
 -a serviceSearchDescriptor=shadow:ou=people,${realm}?sub \
 -a serviceSearchDescriptor=group:ou=group,${realm}?one
```

7. Make the following changes on `hostC`:

```
/usr/sbin/ldapclient manual \
 -a defaultServerList=${oidServerHost} \
 -a defaultSearchBase=${realm} \
 -a authenticationMethod=none \
 -a credentialLevel=anonymous \
 -a serviceAuthenticationMethod=pam_ldap:tls:simple \
 -a
serviceSearchDescriptor=passwd:ou=people,${realm}?one?(|(host=hostC)(host=ALL))
\
 -a serviceAuthenticationMethod=passwd-cmd:tls:simple \
 -a serviceSearchDescriptor=shadow:ou=people,${realm}?sub \
 -a serviceSearchDescriptor=group:ou=group,${realm}?one
```

8. Re-run `sslConfig_OIDclient.sh` on the client as `root`.

These changes to the `ldapclient` command restrict operating system login to those users who either have `host=ALL` or the `host` attribute value that matches the host name.

Linux Client Setup

These procedures have been tested and certified with Red Hat Enterprise Linux 4.6 and 5.1, Oracle Enterprise Linux 5.0, and SuSE Linux Enterprise 9 and 10.

To configure the rules example at the beginning of this chapter, perform the following steps.

1. Configure SSL authentication between Oracle Internet Directory and the Linux clients and verity that it is working correctly.
2. On each client, make a copy of the file `/etc/ldap.conf`.
3. On each client, open `/etc/ldap.conf` in an editor and locate the `pam_filter` entry near the end of the file. It looks like this:

```
pam_filter objectclass=posixaccount
```

4. On `hostA`, change the entry to this:

```
pam_filter &(objectclass=posixaccount)(|(host=ALL)(host=hostA))
```

5. On `hostB`, change the entry to this:

```
pam_filter &(objectclass=posixaccount)(|(host=ALL)(host=hostB))
```

6. On `hostC`, change the entry to this:

```
pam_filter &(objectclass=posixaccount)(|(host=ALL)(host=hostC))
```

The above pam_filter changes restrict operating system login to those users who either have host=ALL or the host attribute value matching the host name.

Optionally, you can use additional attributes in the filter condition specified in pam_filter. For example, most of the operating system user entries have a gidnumber attribute indicating which operating system group the user is in. You can add gidnumber to pam_filter so that you can open operating system access to certain groups. For example, you can open access to users who are in the group507 by specifying the following:

```
pam_filter &(objectclass=posixaccount)(|(host=ALL)(host=hostC)(gidnumber=507))
```

HP-UX 11.23 Client Setup

To configure the rules example at the beginning of this chapter, perform the following steps.

1. Configure SSL authentication between Oracle Internet Directory and the HP-UX clients and verity that it is working correctly.
2. Open sslConfig_OIDclient.sh in an editor and locate the following section:

```
version: 1
dn: cn=ldapuxprofile,ou=ldapuxprofile,${realm}
defaultserverlist: ${oidServerHost}:636
authenticationmethod: tls:simple
serviceauthenticationmethod: pam_ldap:tls:simple
serviceauthenticationmethod: passwd-cmd:tls:simple
cn: ldapuxprofile
defaultsearchbase: ${realm}
credentiallevel: anonymous
servicesearchdescriptor: passwd:ou=people,${realm}?one
servicesearchdescriptor: group:ou=group,${realm}?one
objectclass: top
objectclass: duaconfigprofile
```

3. On hostA, make the following changes, keeping the order of the lines in the file exactly as shown:

```
version: 1
dn: cn=ldapuxprofile,ou=ldapuxprofile,${realm}
defaultserverlist: ${oidServerHost}:636
authenticationmethod: tls:simple
serviceauthenticationmethod: pam_ldap:tls:simple
serviceauthenticationmethod: passwd-cmd:tls:simple
cn: ldapuxprofile
defaultsearchbase: ${realm}
credentiallevel: anonymous
servicesearchdescriptor:
passwd:ou=people,${realm}?one?(|(host=hostA)(host=ALL))
serviceSearchDescriptor: shadow:ou=people,${realm}?sub
servicesearchdescriptor: group:ou=group,${realm}?one
objectclass: top
objectclass: duaconfigprofile
```

4. On hostB, make the following changes, keeping the order of the lines in the file exactly as shown:

```
version: 1
```

```
dn: cn=ldapuxprofile,ou=ldapuxprofile,${realm}
defaultserverlist: ${oidServerHost}:636
authenticationmethod: tls:simple
serviceauthenticationmethod: pam_ldap:tls:simple
serviceauthenticationmethod: passwd-cmd:tls:simple
cn: ldapuxprofile
defaultsearchbase: ${realm}
credentiallevel: anonymous
servicesearchdescriptor:
 passwd:ou=people,${realm}?one?(|(host=hostB)(host=ALL))
serviceSearchDescriptor: shadow:ou=people,${realm}?sub
servicesearchdescriptor: group:ou=group,${realm}?one
objectclass: top
objectclass: duaconfigprofile
```

5. On hostC, make the following changes, keeping the order of the lines in the file exactly as shown:

```
version: 1
dn: cn=ldapuxprofile,ou=ldapuxprofile,${realm}
defaultserverlist: ${oidServerHost}:636
authenticationmethod: tls:simple
serviceauthenticationmethod: pam_ldap:tls:simple
serviceauthenticationmethod: passwd-cmd:tls:simple
cn: ldapuxprofile
defaultsearchbase: ${realm}
credentiallevel: anonymous
servicesearchdescriptor:
 passwd:ou=people,${realm}?one?(|(host=hostC)(host=ALL))
serviceSearchDescriptor: shadow:ou=people,${realm}?sub
servicesearchdescriptor: group:ou=group,${realm}?one
objectclass: top
objectclass: duaconfigprofile
```

6. Re-run `sslConfig_OIDclient.sh` on the client as `root`.

These changes restrict operating system login to those users who either have `host=ALL` or the `host` attribute value matching the particular host name.

A

Troubleshooting

This appendix lists problems you might encounter when configuring or managing Oracle Authentication Services for Operating Systems. It contains the following topics:

Client Configuration Script Errors

This section lists errors you might encounter when executing the client configuration script.

Client Script Failure on AIX 5.3

Before you execute the client script on AIX, you must add at least one user to LDAP. Otherwise, the configuration script might fail with one of these error messages:

```
Cannot find users from all base DN
client setup failed."

Cannot find the group base DN from the LDAP server.
Client setup failed."
```

See "Add At Least One User and One Group to Oracle Internet Directory on AIX" on page 3-7.

SSL Client Script Failure on AIX 6.1

The SSL client configuration script fails on AIX 6.1 due to a problem with the `mksecldap` tool. You can only configure Oracle Authentication Services for Operating Systems in non-SSL mode, using the non-SSL configuration script, on AIX 6.1.

Script Prints Server Hostname with Duplicate Domain

Problem

The server hostname printed by the client script has a duplicate domain name, for example: `myserver.mycompany.com.mycompany.com`.

Solution

When the server script generates the client script, it appends the domain to the server hostname. In most cases, the server hostname is the simple name, so this behavior is correct. If, however, you have set `hostname` to a fully-qualified domain name on your server, the server script generates an incorrect name.

To correct this problem, while executing the client script, type `n` in response to the query:

```
Do you want to configure test-host to authenticate users against the above OID
LDAP server [n]: y
```

which terminates the client script. Then edit the server hostname in the client script and execute the script again. The line to be edited is:

```
oidServerHost="myserver.mycompany.com.mycompany.com"
```

Script Does Not Recognize Non-English Input

Before you run the configuration scripts, you must set your locale by setting the `NLS_LANG` environment variable.

Data Migration Errors

This section lists errors you might encounter when migrating entries to Oracle Authentication Services for Operating Systems.

Sudo Conversion Script Errors

Problem

The `sudo` conversion tool reports parse errors while converting your `/etc/sudoers` file to LDIF format.

Solution

The conversion script in the `sudo` package might not cover all intricacies of your `sudoers` file format. For example, if command aliases are preceded by an exclamation mark (`!`), remove the exclamation mark. Please see the `sudo` package documentation for known limitations.

Tool Problems

This section lists errors you might encounter when using command-line tools with Oracle Authentication Services for Operating Systems.

Error in system-config-users

Problem

You encounter errors when using the `system-config-users` tool.

Solution

Ensure that user entries have all the attributes described in "Migrating from NIS to Oracle Internet Directory" on page 4-2.

Solution

For errors when creating a new group on Red Hat Enterprise Linux, version 4, edit the file `/usr/share/system-config-users/userGroupCheck.py`.

Change:

```
def isGroupnameOk(str, widget):
```

to:

```
def isGroupnameOk(name,widget):
```

The libuser Tools Fail with Python Errors

Problem

You see Python errors when invoking libuser tools such as `system-config-users` and `luseradd`.

Solution

To use `libuser` tools, you must configure your client and server for SSL. See "Switching Between SSL Authentication and Non-SSL Configurations" on page 3-10.

Linux Management Tools Cause Inconsistencies

Problem

Using Linux tools such as `useradd`, `userdel`, `groupadd`, or `groupdel` causes inconsistencies or unexpected behavior.

Solution

These tools are not supported. After you install Oracle Authentication Services for Operating Systems and migrate your data to Oracle Internet Directory, you must use specific tools to manage users, passwords, and other data. Specifically, you must use:

- Oracle Directory Manager
- The LDAP tools and bulk tools in `$ORACLE_HOME/bin`
- The `passwd` command

You can also use the `libuser` tools on Linux distributions that support it, with some limitations. See "Password Policy Not Consistently Enforced" on page A-9.

ldapsearch Error

Problem

When you attempt to perform a search, the server returns this error:

```
Function not implemented. DSA unwilling to perform.
```

Solution

You have attempted to perform a search with a non-indexed attribute specified as a required attribute.

You can search for an attribute in Oracle Internet Directory only if the attribute is indexed. By default, standard attributes of the user and group entries are indexed. If

you use a custom attribute, you can index it by using the `catalog` command. For example:

```
catalog connect="connect_str" add="TRUE" attribute="automountKey"
```

AIX mkuser Command Error

Problem

The AIX `mkuser` command fails with the error:

```
Group "staff" does not exist.
Check "/usr/lib/security/mkuser.default" file.
```

Solution

To resolve this problem, create a group called `staff` in Oracle Internet Directory.

Solution

On AIX 5.3, if the LDAP client and NIS client are configured on the same machine, you cannot create users from the AIX LDAP client. You can rectify this problem by installing APAR IY90556.

> **See Also:** "LDAP configuration management and troubleshooting on AIX" at `http://www.ibm.com/developerworks/`

Solaris id Command Does Not Report Secondary Groups

Problem

Your user account has a primary group and one or more secondary groups. When you type:

```
id -a
```

on a Solaris system after configuring Oracle Authentication Services for Operating Systems, the secondary groups are not displayed.

Solution

Ensure that you are using the `-a` option to the `id` command.

If you are using `id -a` and not seeing secondary groups, you might need to change the LDAP entries for the secondary groups.

First, add the `uid` attribute to objectclass `orclGroup` in your LDAP schema, if it is not there already. See the chapter "Managing Directory Schema" in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for information about adding a new attribute to Oracle Internet Directory.

Then modify the group entries for all secondary groups. Replace `uniquemember: dn` with `memberuid: uid`, where *uid* is an attribute of type `uid` that contains a `uid` value.

Each secondary group entry should resemble this example:

```
dn: cn=dba,cn=groups,dc=us,dc=example,dc=com
memberuid: cms
memberuid: gtest1
```

```
memberuid: oidpam4
memberuid: oidpam5
memberuid: oidpam8
memberuid: orcladmin
objectclass: posixGroup
objectclass: top
objectclass: groupOfUniqueNames
objectclass: orclGroup
cn: cmsdba
displayname: dba
description: DBAgroup.
gidnumber: 7002
```

Testing and Log File Messages

This section describes some testing techniques and explains some messages you might find in log files when running Oracle Authentication Services for Operating Systems.

Enabling Log Messages for All Operations

Problem

Administrators need to monitor Oracle Internet Directory.

Solution

You can enable Oracle Internet Directory debugging, which will cause the Oracle Internet Directory LDAP server to write debug messages to the file *ORACLE_INSTANCE*/diagnostics/logs/OID/*componentName*/oidldapd01s*PID-XXXX*.log where:

- 01 is the instance number, which is 01 by default
- s stands for server
- PID is the server process identifier
- *XXXX* is a number from 0000 to `orclmaxlogfilesconfigured`

To enable debugging, set the debug flag to 1 by using the following command line:

```
ldapmodify -p port -h host -D cn=orcladmin -q -v -f debug.ldif
```

where `debug.ldif` looks like this:

```
dn:cn=componentname,cn=osdldapd,cn=subconfigsubentry
changetype:modify
replace: orcldebugflag
orcldebugflag:1
```

Solution

You can set a debug level that causes Oracle Internet Directory to generate log messages for all operations.

Set the function trace debug level on Oracle Internet Directory by using the following command line:

```
ldapmodify -p port -h host -D cn=orcladmin -q -v -f debug.ldif
```

where `debug.ldif` looks like this:

```
dn:cn=componentname,cn=osdldapd,cn=subconfigsubentry
changetype: modify
replace: orcldebugflag
orcldebugflag: 117440511
-
replace: orcldebugforceflush
orcldebugforceflush: 1
```

Testing StartTLS

Problem

StartTLS, which enables you to negotiate an SSL connection on a previously clear connection, is transparent to the user. Administrators need a way to verify that StartTLS is working.

> **Note:** StartTLS is not used on HP-UX and Solaris Oracle Internet Directory servers. On these platforms, SSL is configured on a different port from non-SSL connections.

Solution

To verify that StartTLS is working, set a debug level that causes Oracle Internet Directory to generate a log message when an SSL negotiation begins. Because the clients are all pointing to the non-SSL port, generation of this message implies that startTLS is working.

Perform the following steps:

1. Set the function trace debug level on Oracle Internet Directory by using the following command line:

   ```
   ldapmodify -p port -h host -D cn=orcladmin -q -f debug.ldif -v
   ```

   where `debug.ldif` looks like this:

   ```
   dn:
   changetype: modify
   replace: orcldebugflag
   orcldebugflag: 25165824
   -
   replace: orcldebugforceflush
   orcldebugforceflush: 1
   ```

2. Perform an authentication operation that invokes the Oracle Internet Directory server. For example, use `ssh` to connect to a client that is configured to authenticate against Oracle Internet Directory.
3. Examine the log files in `$ORACLE_HOME/ldap/log`. Look for messages containing the string `gslsflnNegotiateSSL`.

Password Syntax Errors

Problem

Log files contain messages about password syntax, and Oracle Internet Directory is not being used for password policy enforcement.

Solution

If you are not using Oracle Internet Directory for password policy enforcement, you must disable password policies in Oracle Internet Directory by setting `orclpwdpolicyenable` to 0. To avoid messages about password syntax, you must also disable the password syntax check by setting `pwdCheckSyntax` to 0.

Testing Connection to the Oracle Internet Directory Server on RHEL or OEL

You can test the connection from a Red Hat Enterprise Linux or Oracle Enterprise Linux client to the Oracle Internet Directory server by using the OpenLDAP command `ldapsearch`, as follows:

```
ldapsearch -ZZ -d 1 -x -h your_oid_host -p 389 -b your_realm -D user_dn -W
 -s sub objectclass=*
```

If the invocation succeeds, your connection to the server is working.

Also check the time synchronization between machines by using the `date` or `time` command. The discrepancy should be less than two minutes.

Testing Root CA Certificate on Red Hat Enterprise Linux or Oracle Enterprise Linux

Verify that the root CA certificate under `$ORACLE_INSTANCE/OID/admin/wallet/root/cacert.txt` is the same as the Operating System client certificate `/etc/oracle-certs/oid-test-cert.pem`

Verify that the certificate is valid. Type:

```
openssl x509 -in oid-test-cert.pem -noout -text
```

Look at the `Validity` section of the output. The valid times are specified as `Not Before` and `Not After`, for example:

```
Validity
            Not Before: Mar 25 11:52:37 2010 GMT
            Not After : Mar 24 11:52:37 2011 GMT
```

User Login Errors

This section lists errors users might encounter when attempting to log in when Oracle Authentication Services for Operating Systems is used for authentication.

Users Cannot Log In

Problem

Users cannot log in after you run the client configuration script. Operating system log files might contain an error message similar to this:

```
April 10 14:32:21 myhost sshd: nss_ldap: failed to bind to LDAP server
 ldap://ldaphost: Inappropriate authentication
```

Solution

Users cannot log in unless Oracle Internet Directory allows anonymous binds. If anonymous binds have been disabled, enable them as follows:

Create an LDIF file that looks like this:

```
dn: cn=oid1,cn=osdldapd,cn=subconfigsubentry
changetype: modify
```

```
replace: orclAnonymousBindsFlag
orclAnonymousBindsFlag: 1
```

Execute the command:

```
ldapmodify -D cn=orcladmin -q -p portNum -h hostname -f ldifFileName
```

Problem

Users cannot log in after you run the client configuration script.

Solution

On some operating systems, if `nscd` or `sshd` is running while you execute the `config_OIDclient.sh` or `sslConfig_OIDclient.sh` script, user authentication might not work after the configuration. Restart `sshd` or `nscd` to correct the problem.

Problem

You have configured Active Directory synchronization. After a password change, a user cannot log in using the new password.

Solution

You must change the password in the Active Directory user entry, not the Oracle Internet Directory entry.

Problem

Users cannot log in after you run the SSL version of the configuration script using a custom certificate.

Solution

Examine the subject DN in the server certificate. There should be only one CN and it should contain the hostname of the SSL server. The current implementation of OpenSSL fails to verify the hostname of the server certificate if there are multiple CNs in the subject DN.

User's Home Directory Does Not Exist

Problem

Adding or migrating a user to Oracle Internet Directory does not create that user's home directory.

Solution

On Linux systems, you do not have to create a user's home directory on the client computer when you add that user to Oracle Internet Directory. The client configuration script that you ran on each client computer enabled the creation of each user's home directory on first login. On operating systems other than Linux, however, you must manually create user home directories.

User's Shell Does Not Exist

Problem

When attempting to log in, the user sees a message such as:

```
No shell
```

```
Connection closed by foreign host.
```

Solution

This problem occurs when a user entry in Oracle Internet Directory specifies a shell pathname that does not exist on the computer where the user is logging in. Supported shells and shell pathnames vary from one operating system to another. For example, one operating system might have `sh`, `csh`, `bash`, and `tcsh` under `/bin`, and another might have `sh` and `csh` under `/usr/bin`.

If the user must be able to log in on computers with different shell pathnames, you might have to create a symbolic link to the shell on one of the computers.

Password Policy Not Consistently Enforced

Problem

Oracle Internet Directory fails to enforce password policies, or password policy enforcement is not as expected.

Solution

If you use Oracle Internet Directory to enforce password policies, you cannot use tools in the `libuser` package to add passwords or entries containing passwords. The reason is that the `libuser` tools generate a hashed password before sending it to Oracle Internet Directory, so Oracle Internet Directory cannot determine whether the password meets policy criteria or not. Use the LDAP tools or Oracle Directory Manager instead.

Solution

If you are using Oracle Internet Directory for password policy enforcement, you must set `shadowmax` to `99999` and `shadowexpire` to `-1` to disable password expiration by the operating system.

B

Properties File for LDAP Migration

This is a sample of a properties file, discussed in "Migrating from Another LDAP Directory to Oracle Internet Directory" on page 4-3.

```
######################################################################
## This configuration file provides necessary information for     ##
## performing the bootstrapping of OiD and a Connected directory. ##
######################################################################

# Source Type : Specifies whether, source end of the bootstrapping is
# LDAP or LDIF.
#
#
odip.bootstrap.srctype = LDIF

# Source URL : In case of LDAP source type it specifies the source directory
# location. In case of LDIF it specifies the location of the LDIF file.
#
# NOTE - e.x for LDAP the expeced format is host[:port]
#            for LDIF the expected format is absolute path of the file
#
odip.bootstrap.srcurl = oracle/ldap/odip/scr/IPlanet.ldif

# Source DN : This information supplements the Source URL. In case of LDIF
# binding this parameter is meaningless. However in case of LDAP this parameter
# specifies the Bind DN.
#
#odip.bootstrap.srcdn

# Source Password : Bind password. In case of LDAP binding this is used as
# security credential
#
#odip.bootstrap.srcpasswd

# Destination Type : Specifies whether, destination end of the bootstrapping
# is LDAP or LDIF.
#
# NOTE - In future bootstrapping with a TAGGED and PLSQL based interfaces
# would be supported.
#
odip.bootstrap.desttype = LDIF

# Destination URL : In case of LDAP it specifies the directory location
# In case of LDIF it specifies the location of the LDIF file.
#
# NOTE - e.x for LDAP the expeced format is host[:port]
#            for LDIF the expected format is absolute path of the file
```

```
#
odip.bootstrap.desturl = /oracle/ldap/odip/scr/OiD.ldif

# Destination DN : This information supplements the destination URL.
# In case of LDIF binding this parameter is meaningless. However in case of
# LDAP this parameter specifies the Bind DN.
#
#odip.bootstrap.destdn

# Destination Password : Bind password. In case of LDAP binding this is
# used as security credential
#
# NOTE - It is not recommended to specify the password in this file.
#
#odip.bootstrap.destpasswd

# and domain mappings.
#
odip.bootstrap.mapfile = /oracle/ldap/odip/scr/bootstrap.map

#
# NOTE - If this file already exists then it will be backed up and a new
# version will be created
#
odip.bootstrap.logfile = /oracle/ldap/odip/scr/bootstrap.log

# Log Messages Severity : Specifies the type of the log messages that needs
# to be logged
#                     INFO     ---- 1
#                     WARNING  ---- 2
#                     DEBUG    ---- 4
#                     ERROR    ---- 8
#
# NOTE - A combination of these types could also be given. for ex if you are
interested
# only in WARNING and ERROR message then specify value 8+1 i.e 9 Similarly for all
# types of message use 1 + 2 + 4 + 8 = 15
#
odip.bootstrap.logseverity = /oracle/ldap/odip/scr/bootstrap.log

# Trace file : Specifies the location of the trace file. The default
# trace file will be bootstrap.trc created under
# $ORACLE_HOME/ldap/odi/log directory
#
# NOTE - If this file already exists then it will be backed up and a new
# version will be created
#
odip.bootstrap.trcfile = /oracle/ldap/odip/scr/bootstrap.trc
```

C

Sample Mapfiles

This appendix contains a template mapfile and some sample mapfiles.

This appendix includes the following sections:

Template Mapfile

```
DomainRules
# Specify the list of domain rules
DomainExclusionList
# Specify the list of  domains to be excluded in migration
###
AttributeRules
# List the attributes that are to be migrated
AttributeExclusionList
# Specify the list of attributes that are to be excluded
~
```

Sample Mapfile 1

```
# This file contains the domain rules with the list of containers to be migrated
# and the list of attributes to be migrated.
DomainRules
ou=groups,dc=us,dc=example,dc=com
ou=people,dc=us,dc=example,dc=com
ou=system administrators,dc=us,dc=example,dc=com
###
AttributeRules
Cn
Sn
Givenname
Objectclass
```

Sample Mapfile 2

```
# This file contains the domain rules with the list of containers to be migrated
# and the list of attributes to be filtered
DomainRules
ou=groups,dc=us,dc=example,dc=com
ou=people,dc=us,dc=example,dc=com
ou=system administrators,dc=us,dc=example,dc=com
###
AttributeRules
*:*
AttributeExclusionList
modifytimestamp
createtimestamp
modifiersname
creatorsname
```

Sample Mapfile 3

```
# This file contains domain rules with the list of containers to be excluded and
# the list of attributes to be excluded
DomainRules
*:*
DomainExclusionList
ou=system administrators,dc=us,dc=example,dc=com
###
AttributeRules
*:*
AttributeExclusionList
modifytimestamp
createtimestamp
modifiersname
creatorsname
```

Oracle Directory Server Enterprise Edition Mapfile 1

This is a mapfile for Oracle Directory Server Enterprise Edition, formerly Sun Directory Server Enterprise Edition.

```
# This file contains domain rules with the list of containers to be excluded and
# the list of attributes to be excluded
DomainRules
ou=groups,dc=us,dc=example,dc=com:ou=groups,dc=us,dc=example,dc=com:cn=%,ou=group,
dc=us,dc=example,dc=com
ou=people,dc=us,dc=example,dc=com: ou=people,dc=us,dc=example,dc=com:uid=%
ou=people,dc=us,dc=example,dc=com
ou=system
administrators,dc=us,dc=example,dc=com:ou=people,dc=us,dc=example,dc=com:uid=%,
ou=people,dc=us,dc=example,dc=com
DomainExclusionList
###
AttributeRules
*:*
AttributeExclusionList
modifytimestamp
createtimestamp
modifiersname
creatorsname
```

```
nsuniqueid
aci
```

Oracle Directory Server Enterprise Edition Mapfile 2

This is a mapfile for Oracle Directory Server Enterprise Edition, formerly Sun Directory Server Enterprise Edition.

```
# This file contains domain rules with the list of containers to be excluded and
# the list of attributes to be excluded
DomainRules
*:*
###
AttributeRules
*:*
AttributeExclusionList
modifytimestamp
createtimestamp
modifiersname
creatorsname
nsuniqueid
aci
```

eDirectory Mapfile

```
# This file contains domain rules with the list of containers to be excluded and
# the list of attributes to be excluded
DomainRules
*:*
###
AttributeRules
*:*
AttributeExclusionList
modifytimestamp
createtimestamp
modifiersname
creatorsname
```

D

Synchronization Profile for Active Directory Integration

This properties file was generated by running `expressSyncSetup` and then customizing the file, as described in "Configuring Oracle Directory Integration Platform" on page 5-2. The customizations are shown in **boldface**.

```
# USE THIS MAP FILE, IF DOMAIN IN ACTIVE DIRECTORY IS DIFFERENT FROM DOMAIN IN OID
# FOR ONE-TO-ONE DOMAIN MAPPING USE ACTIVECHG.MAP.MASTER IN ODI/CONF DIRECTORY
DomainRules
CN=USERS,DC=test,DC=com:ou=People,dc=us,dc=example,dc=com:uid=%,ou=People,dc=us,dc
=example,dc=com
###
AttributeRules
# attribute rule common to all objects
objectguid: :binary: :orclobjectguid: : :bin2b64(objectguid)
ObjectSID: :binary: :orclObjectSID: : :bin2b64(ObjectSID)
distinguishedName: : : :orclSourceObjectDN: :orclADObject
# attribute rule for mapping windows organizationalunit
ou: : :organizationalunit:ou: : organizationalunit:
# attribute rule for mapping directory containers
cn: : :container: cn: :orclContainer:
# attribute rule for mapping directordomains
dc: : :domain: dc: :domain:
# USER ENTRY MAPPING RULES
# attribute rule for mapping windows LOGIN id
sAMAccountName,userPrincipalName: : :user:orclSAMAccountName:
:orclADUser:toupper(truncl(userPrincipalName,'@'))+"$"+sAMAccountname
# attribute rule for mapping Active Directory LOGIN id
userPrincipalName: : :user:orclUserPrincipalName: :orclADUser:userPrincipalName
# Map the userprincipalname to the nickname attr by default
#userPrincipalName: : :user:uid: :inetorgperson:userPrincipalName
# Map the SamAccountName to the nickname attr if required
# If this rule is enabled, userprincipalname rule needs to be disabled
sAMAccountName: : :user:uid: :inetorgperson
# Assign the userprincipalname to Kerberaos principalname
userPrincipalName: : :user:krbPrincipalName:
:orcluserv2:trunc(userPrincipalName,'@')+'@'+toupper(truncl(userPrincipalName,'@')
)
# This rule is mapped as SAMAccountName is a mandatory attr on AD
# and sn is mandatory on OID. sn is not mandatory on Active Directory
SAMAccountName: : :user:sn: : person:
# attributes to map to cn - normally this is the given name
cn: : :person:cn: :person:
departmentNumber: : :inetorgperson:departmentnumber: :organizationalperson:
# attribute rule for mapping entry and to create orclUserV2
# There should be a mapping rule with orcluserv2 objectclass
```

```
# without which the PORTAL may not function properly
# The next rule shows any attribute of any objectclass can be mapped
# to different attribute of different objectclass so long as the
# schema and syntax are compatible.
givenName: : :user:displayName: :orclUserV2:
employeeID: : :user:employeeNumber: :inetOrgPerson:
physicalDeliveryOfficeName: : :user:physicalDeliveryOfficeName:
:organizationalPerson:
title: : :user:title: :organizationalPerson:
mobile: : :organizationalperson:mobile: :inetorgperson:
telephonenumber: : :organizationalperson:telephonenumber: :inetorgperson:
facsimileTelephoneNumber: : :organizationalperson:facsimileTelephoneNumber:
:inetorgperson:
l: : :user:l: :person:
# mail needs to be assigned valid value for default settings in DAS
userPrincipalName: : :user:mail: :inetorgperson:
# GROUP ENTRY MAPPING RULES
cn: : :group:cn: :groupofuniquenames:
# displayname needs to be assigned a valid value for default settings on DAS
SAMAccountName: : :group:displayName: :orclgroup:
# Description needs tobe assigned a valid value for default settings on DAS
Description: : :group:Description: :groupOfUniqueNames:
member: : :group:uniquemember: :groupofUniqueNames:dnconvert(member)
managedby: : :group:owner: :orclprivilegegroup:dnconvert(managedby)
sAMAccountName: : :group:orclSAMAccountName: :orclADGroup:
```

E

Sample Script Output

This appendix contains sample script output. It includes the following sections:

- Non-SSL Server Script Run on Oracle Enterprise Linux 4
- SSL Server Script Run on Oracle Enterprise Linux 4
- Non-SSL Client Script Run on Oracle Enterprise Linux 4
- SSL Client Script Run on Oracle Enterprise Linux 4
- Reset Script Run on Oracle Enterprise Linux 4

Non-SSL Server Script Run on Oracle Enterprise Linux 4

```
$ ./config_OIDserver.sh

OAS4OS: Release 11.1.1.3.0 - Production
Copyright (c) 2010 Oracle.  All rights reserved.

Configuring Oracle Authentication Services for Operating Systems on
the Oracle Internet Directory server.
Make sure that your OID server is currently up and running.
Specify the ORACLE_HOME path: /u01/Middleware/Oracle_IDM1
Specify the ORACLE_INSTANCE path: /u01/Middleware/asinst_1
Specify the OID realm: dc=example,dc=com
Specify the OID non-SSL port [3060]: 3060
Specify the OID component name [oid1]: oid1
Enter OID cn=orcladmin password:
The PAM client can be configured to interact with OID anonymously or via
a specific user DN and password.
Do you wish to have the PAM client connect with OID anonymously [y/n]: n
Specify the user DN for connecting to OID: cn=myuser,cn=users,dc=example,dc=com
Enter the user's password:

OAS4OS Client Config Script: /u01/Middleware/asinst_1/OID/oas4os/oid1/scripts_
20100406231223/config_OIDclient.sh

Successfully completed configuration
```

SSL Server Script Run on Oracle Enterprise Linux 4

```
OAS4OS: Release 11.1.1.3.0 - Production
Copyright (c) 2010 Oracle.  All rights reserved.

Configuring Oracle Authentication Services for Operating Systems on
the Oracle Internet Directory server.
```

```
Make sure that your OID server processes were started by using opmnctl or
Oracle Fusion Middleware Control.

Specify the ORACLE_HOME path: /u01/Middleware/Oracle_IDM1
Specify the ORACLE_INSTANCE path: /u01/Middleware/asinst_1
Specify the OID realm: dc=example,dc=com
Specify the OID non-SSL port [3060]: 3060
Specify the OID SSL port [3131]: 3131
Specify the OID component name [oid1]: oid1
Enter OID cn=orcladmin password:
The PAM client can be configured to interact with OID anonymously or via
a specific user DN and password.
Do you wish to have the PAM client connect with OID anonymously [y/n]: n
Specify the user DN for connecting to OID: cn=myuser,cn=users,dc=example,dc=com
Enter the user's password:
You can provide an SSL certificate or use the script to
create and update OID SSL configuration with a test certificate.
Do you have an SSL certificate [y/n]: y
Specify the SSL Certificate file: /home/oracle/pem.cert

OAS4OS Client Config Script: /u01/Middleware/asinst_1/OID/oas4os/oid1/scripts_
20100406231526/sslConfig_OIDclient.sh

Successfully completed configuration
```

Non-SSL Client Script Run on Oracle Enterprise Linux 4

```
$ ./config_OIDclient.sh

OAS4OS: Release 11.1.1.3.0 - Production
Copyright (c) 2010 Oracle.  All rights reserved.

Configuring this client to use LDAP for authentication...
OID server HOST : oid-host.example.com
OID server port : 3060

Do you want to configure client01 to authenticate users against the above OID LDAP
 server [n]: y
User DN for connecting to OID: cn=myuser,cn=users,dc=example,dc=com
Enter the user's password:
Saved original files in /etc/oracle_backup_20100406231757 directory
Executing auth-config ...
Stopping portmap:                                          [  OK  ]
Starting portmap:                                          [  OK  ]
setsebool:  SELinux is disabled.
Shutting down NIS services:                                [  OK  ]
Binding to the NIS domain:                                 [  OK  ]
Listening for an NIS domain server.
Stopping nscd:                                             [  OK  ]
Starting nscd:                                             [  OK  ]
Stopping nscd:                                             [  OK  ]
Starting nscd:                                             [  OK  ]
Configured test-host for LDAP authentication.
```

SSL Client Script Run on Oracle Enterprise Linux 4

```
$ ./sslConfig_OIDclient.sh

OAS4OS: Release 11.1.1.3.0 - Production
Copyright (c) 2010 Oracle.  All rights reserved.

Configuring this client to use LDAP for authentication...
OID server HOST : oid-host.example.com
OID server SSL port : 3131
Do you want to configure client01 to authenticate users against the above OID LDAP
 server [n]: y
User DN for connecting to OID: cn=myuser,cn=users,dc=example,dc=com
Enter the user's password:
Saved original files in /etc/oracle_backup_20100407024405 directory
Executing auth-config ...
Stopping portmap:                                          [  OK  ]
Starting portmap:                                          [  OK  ]
setsebool:  SELinux is disabled.
Shutting down NIS services:                                [  OK  ]
Binding to the NIS domain:                                 [  OK  ]
Listening for an NIS domain server.
Stopping nscd:                                             [  OK  ]
Starting nscd:                                             [  OK  ]
The libuser package can be configured for user management via SSL LDAP.
Do you want to enable this host to manage users on OID  [y]: n
Stopping nscd:                                             [  OK  ]
Starting nscd:                                             [  OK  ]
Configured test-host for LDAP authentication.
```

Reset Script Run on Oracle Enterprise Linux 4

```
$ ./resetClient.sh

OAS4OS: Release 11.1.1.3.0 - Production
Copyright (c) 2009 Oracle.  All rights reserved.

Resetting OAS4OS client ...
Executing this script will reset this OAS4OS client machine to backed up state.
Do you want to reset test-host and remove OAS4OS configuration [n]: y
Specify the OAS4OS backup folder path: /etc/oracle_backup_20100406231757
Executing auth-config ...
Stopping portmap:                                          [  OK  ]
Starting portmap:                                          [  OK  ]
setsebool:  SELinux is disabled.
Shutting down NIS services:                                [  OK  ]
Binding to the NIS domain:                                 [  OK  ]
Listening for an NIS domain server.
Stopping nscd:                                             [  OK  ]
Starting nscd:                                             [  OK  ]
Stopping nscd:                                             [  OK  ]
Starting nscd:                                             [  OK  ]
Client reset completed successfully.
```

F

LDAP Containers Added by Configuration Script

The Oracle Authentication Services for Operating Systems server configuration script adds the following empty containers under the realm DN. These containers are the default locations for data when you migrate from local files, such as `/etc/passwd` and `/etc/group`, or from a NIS database.

Table F–1 LDAP Containers Added by Server Configuration Script

Container	Use
`ou=people`	Stores login and password information similar to `/etc/password` and `/etc/shadow`. The objects stored here are `posixAccount` and `shadowAccount`.
`ou=group`	Stores group information, similar to `/etc/group`. Objects of the type `posixGroup` are stored here.
`ou=services`	Stores information about available services, similar to `/etc/services`. Objects of the type `ipService` are stored here.
`ou=protocols`	Stores information about protocols, similar to `/etc/protocols`. Objects of the type `ipProtocols` are stored here.
`ou=rpc`	Stores information related to remote procedure calls (RPCs) similar to `/etc/rpc`. Objects of the type `oncRPC` are stored here.
`ou=hosts`	Stores the host table, similar to `/etc/hosts`. Objects of the type `ipHost` are stored here.
`ou=networks`	Stores names of networks, similar to `/etc/networks`. Objects of the type `ipNetwork` are stored here.
`ou=netgroup`	Stores netgroup information in the object type `nisNetwork`, `nisNetgroup`.
`ou=aliases`	Stores mailgroup information in the object type `mailGroup`.
`ou=mounts`	Stores automount information.
`nismapname=netgroup.byuser`	Sets an NIS map containing group name, user name and host name. The username is the key in the map.

G

Working Configuration Files

If your configuration files are corrupted at some point, you can return them to the correct state by editing them. This appendix contains some examples.

Red Hat Enterprise Linux and Oracle Enterprise Linux Configuration Files

This section contains example files for Red Hat Enterprise Linux and Oracle Enterprise Linux

/etc/pam.d/system-auth

```
# %PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      /lib/security/$ISA/pam_env.so
auth        sufficient    /lib/security/$ISA/pam_unix.so likeauth nullok
auth        required      /lib/security/$ISA/pam_deny.so

account     required      /lib/security/$ISA/pam_unix.so
account     sufficient    /lib/security/$ISA/pam_succeed_if.so uid < 100 quiet
account     required      /lib/security/$ISA/pam_permit.so

password    requisite     /lib/security/$ISA/pam_cracklib.so retry=3
password    sufficient    /lib/security/$ISA/pam_unix.so nullok use_authtok md5
shadow nis
password    required      /lib/security/$ISA/pam_deny.so
session     required      /lib/security/$ISA/pam_limits.so
session     required      /lib/security/$ISA/pam_unix.so
```

/etc/pam.d/sshd

```
#%PAM-1.0
auth       required     pam_stack.so service=system-auth
auth       required     pam_nologin.so
account    required     pam_stack.so service=system-auth
password   required     pam_stack.so service=system-auth
session    required     pam_stack.so service=system-auth
session    required     pam_loginuid.so
```

/etc/sysconfig/authconfig

```
USEWINBINDAUTH=no
USEKERBEROS=no
USESYSNETAUTH=no
USEPAMACCESS=no
```

```
USEMKHOMEDIR=no
FORCESMARTCARD=no
USESMBAUTH=no
USESMARTCARD=no
USELDAPAUTH=yes
USEPASSWDQC=no
PASSWDALGORITHM=md5
USEWINBIND=no
USESHADOW=no
USEDB=no
USEHESIOD=no
USELDAP=no
SELOCAUTHORIZE=yes
USECRACKLIB=yes
USENIS=yes
```

H

Prerequisite Packages

This appendix lists packages that are required as prerequisites for configuring Oracle Authentication Services for Operating Systems.

For information about platforms not covered in this chapter, please see Note 1064891.1: Oracle Authentication Services for Operating Systems Documentation Addendum (11.1.1.3). This document is available on My Oracle Support at https://support.oracle.com.

Red Hat Enterprise Linux and Oracle Enterprise Linux

The following packages are required on Red Hat Enterprise Linux and Oracle Enterprise Linux.

Cyrus-sasl

```
cyrus-sasl-gssapi-2.1.22-4
cyrus-sasl-devel-2.1.22-4
cyrus-sasl-devel-2.1.22-4
cyrus-sasl-2.1.22-4
cyrus-sasl-sql-2.1.22-4
cyrus-sasl-ntlm-2.1.22-4
cyrus-sasl-ntlm-2.1.22-4
cyrus-sasl-gssapi-2.1.22-4
cyrus-sasl-sql-2.1.22-4
cyrus-sasl-ldap-2.1.22-4
cyrus-sasl-plain-2.1.22-4
cyrus-sasl-lib-2.1.22-4
cyrus-sasl-lib-2.1.22-4
cyrus-sasl-plain-2.1.22-4
cyrus-sasl-ldap-2.1.22-4
cyrus-sasl-2.1.22-4
gnu-crypto-sasl-jdk1.4-2.1.0-2jpp.1
cyrus-sasl-md5-2.1.22-4
cyrus-sasl-md5-2.1.22-4
```

Open SSL

```
openssl-0.9.8e-7.el5
openssl097a-0.9.7a-9.el5_2.1
xmlsec1-openssl-1.2.9-8.1
xmlsec1-openssl-devel-1.2.9-8.1
xmlsec1-openssl-devel-1.2.9-8.1
openssl-devel-0.9.8e-7.el5
openssl-perl-0.9.8e-7.el5
```

```
openssl-devel-0.9.8e-7.el5
xmlsec1-openssl-1.2.9-8.1
openssl-0.9.8e-7.el5
```

Open LDAP

```
openldap-2.3.43-3.el5
openldap-devel-2.3.43-3.el5
compat-openldap-2.3.43_2.2.29-3.el5
openldap-servers-overlays-2.3.43-3.el5
openldap-clients-2.3.43-3.el5
compat-openldap-2.3.43_2.2.29-3.el5
openldap-servers-sql-2.3.43-3.el5
openldap-2.3.43-3.el5
openldap-devel-2.3.43-3.el5
openldap-servers-2.3.43-3.el5
```

Index

Numerics

A

C

D

E

H

I

L

M

N

O

P

R

S

T

U