Working With Oracle® Solaris 11.4 Directory and Naming Services: LDAP

Part No: E61012
November 2020

ORACLE

Working With Oracle Solaris 11.4 Directory and Naming Services: LDAP

Part No: E61012

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle recognizes the influence of ethnic and cultural values and is working to remove language from our products and documentation that might be considered insensitive. While doing so, we are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is an ongoing, long-term process.

Référence: E61012

Accessibilité de la documentation

Pour plus d'informations sur l'engagement d'Oracle pour l'accessibilité de la documentation, visitez le site Web Oracle Accessibility Program, à l'adresse : http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Accès aux services de support Oracle

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info ou le site http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs si vous êtes malentendant.

Contents

Using This Documentation

- **Overview** – Describes the LDAP naming service, methods for planning its use, and steps to implement LDAP.
- **Audience** – Technicians, system administrators, and authorized service providers.
- **Required knowledge** – Familiarity with concepts and terminologies related to LDAP.

Product Documentation Library

Documentation and resources for this product and related products are available at http://www.oracle.com/pls/topic/lookup?ctx=E37838-01.

Feedback

Provide feedback about this documentation at http://www.oracle.com/goto/docfeedback.

♦ ♦ ♦ CHAPTER 1

Introduction to the LDAP Naming Service

The Lightweight Directory Access Protocol (LDAP) is the secure network protocol used to access directory servers for distributed naming and other directory services. This standards-based protocol supports a hierarchical database structure. You can use this protocol to provide naming services in both UNIX and multiplatform environments. This chapter covers the following topics:

For a description of the example IP addresses used in this guide, see the "IP address" entry in "LDAP Glossary" on page 139.

Overview of the LDAP Naming Service

Oracle Solaris supports LDAP on Oracle Unified Directory (OUD) and OpenLDAP directory servers. However, any generic directory server can function as an LDAP server. In this guide, the terms *directory server* and *LDAP server* are synonymous and used interchangeably.

For more information about OUD, see *Oracle® Fusion Middleware Administering Oracle Unified Directory*. For more information about OpenLDAP, see *OpenLDAP Software 2.4 Administrator's Guide*.

LDAP has become a term that refers more to the naming service than to the protocol. Throughout this guide, the term LDAP is used to refer to the service rather than to the protocol.

The LDAP naming service is one naming service that is supported in Oracle Solaris. For information about other naming services, see *Working With Oracle Solaris 11.4 Directory and Naming Services: DNS and NIS*. For a comparison of the different naming services in Oracle Solaris, see "Comparing the Naming Services" in *Working With Oracle Solaris 11.4 Directory and Naming Services: DNS and NIS*.

LDAP performs the following services:

- Naming service – LDAP provides naming data in accordance with a client request. For example, when resolving host names, LDAP functions like DNS by providing the fully qualified domain names. Suppose that the name of a domain is `west.example.net`. If an application requests the host name by using `gethostbyname()` or `getnameinfo()`, LDAP returns the value *server*`.west.example.net`. While LDAP naming service can be used to look up host names, Oracle recommends using DNS to look up host names.
- Authentication service – LDAP manages and provides information that relates to client identity, authentication, and accounts. Therefore, LDAP implements security measures to provide information only to authorized requesters.

The LDAP naming service provides the following advantages:

- With the replacement of application-specific databases, information is consolidated and the number of distinct databases to manage is reduced.
- Different naming services can share data.
- Uses a central repository for data.
- Performs frequent data synchronization between masters and replicas.
- Multiplatform and multi-vendor compatible.

The following restrictions apply to the LDAP naming service:

- An LDAP server cannot be its own client.
- A client cannot be a client of NIS and LDAP at the same time.

Setting up and managing an LDAP naming service is complex and requires careful planning. For information about planning for LDAP services, see Chapter 3, “Planning Requirements for LDAP Naming Services”.

How LDAP Stores Information

The LDAP naming service stores information in a directory information tree (DIT). The DIT consists of hierarchically structured containers of information that follow a defined LDAP schema.

The default schema that is followed by most DITs suffices for most networks that use LDAP. However, the DIT is flexible. You can specify search descriptors in the client profile to override the default structure of a DIT. For more information about search descriptors, see “Service Search Descriptors and Schema Mapping” on page 37.

The following table shows the containers of a DIT and the type of information each container stores. For more information, see “Directory Information Tree” on page 34.

TABLE 1 Types of Information in Default DIT Containers

Default Container	Information Type
ou=Ethers	bootparams, ethers
ou=Group	group
ou=Hosts	hosts, ipnodes, publickey for hosts
ou=Aliases	aliases
ou=Netgroup	netgroup
ou=Networks	networks, netmasks
ou=People	passwd, shadow, user_attr, audit_user, publickey for users
ou=Protocols	protocols
ou=Rpc	rpc
ou=Services	services
ou=SolarisAuthAttr	auth_attr
ou=SolarisProfAttr	prof_attr, exec_attr
ou=projects	project
automountMap=auto_*	auto_* (automount maps)

LDAP Commands

Oracle Solaris provides general LDAP commands and LDAP configuration commands. The general LDAP commands do not require the system to be configured with the LDAP naming service. LDAP configuration commands can be run on clients that are configured with the LDAP naming service.

General LDAP Commands

General LDAP commands can be run on any system and do not require the system to be configured with the LDAP naming service. LDAP commands support a common set of options, including authentication and bind parameters. These commands support a common text-based format called LDAP Data Interchange Format (LDIF) for representing directory information. The following commands manipulate directory entries:

- `ldapsearch` – Searches the LDAP schema for specified entries. See the `ldapsearch(1oldap)` man page.
- `ldapmodify` – Modifies LDAP entries in the schema. See the `ldapmodify(1oldap)` man page.

- `ldapadd` – Adds LDAP entries in the schema. See the `ldapmodify(1oldap)` man page.
- `ldapdelete` – Removes LDAP entries from the schema. See the `ldapdelete(1oldap)` man page.

LDAP Configuration Commands

The following commands configure the LDAP client system or modify the client configuration:

- `ldaplist` – Displays information retrieved from the LDAP server. See the `ldaplist(1)` man page.
- `ldapaddent` – Creates LDAP entries in the schema from corresponding `/etc` files. See the `ldapaddent(8)` man page.
- `ldapservercfg` – Prepares and populates the DIT with data to serve clients. See the `ldapservercfg(8)` man page.
- `ldapclient` – Initializes an LDAP client machine. See the `ldapclient(8)` man page.

♦ ♦ ♦ CHAPTER 2

LDAP and Authentication Service

The LDAP naming service can use the LDAP repository to provide authentication service. This chapter discusses LDAP's authentication services and covers the following topics:

LDAP Naming Service Security Model

LDAP supports security features such as authentication and controlled access to ensure integrity and privacy of the information that LDAP clients obtain. This section describes how an LDAP client authenticates to the LDAP server and how a user authenticates to a client.

To access the information in the LDAP repository, an LDAP client establishes its identity with the directory server. The identity can be either anonymous or as a host or user that is recognized by the LDAP server. LDAP supports the proxy authentication and the per-user authentication of identities.

The pluggable authentication module (PAM) service determines whether a user login is successful. Based on the client's identity and the server's access control information, the LDAP server enables the LDAP client to read directory information. For more information about access control, refer to the documentation for the directory server that you are using.

The types of LDAP Authentication are as follows:

- Proxy authentication – The identity is based on the system where the request originates. After the system is authenticated, all users on that system can access the directory server.
- Per-user authentication – The identity is based on each user. Every user must be authenticated to access the directory server and issue various LDAP requests.

The basis for user authentication differs depending on the PAM module. See “Pluggable Authentication Methods” on page 22. LDAP can use the following PAM modules:

- `pam_krb5` module – Uses the Kerberos server for authentication. For more information, see the `pam_krb5`(7) man page. For a more extensive description about Kerberos, see *Managing Kerberos in Oracle Solaris 11.4*.
- `pam_ldap` module – Uses the LDAP server and local host server for authentication. For more information, see the `pam_ldap`(7) man page. For information about using the `pam_ldap` module, see “LDAP Account Management” on page 28.
- Equivalent `pam_unix_*` modules – Information is provided by the system and the authentication is determined locally.

Note - The `pam_unix` module is no longer supported in Oracle Solaris. This module has been replaced by a different set of service modules that provides equivalent or greater functionality. In this book, `pam_unix` refers to the modules that provide equivalent functionality, not to the `pam_unix` module.

If the `pam_ldap` module is used, the naming service and the authentication service access the directory in the following ways:

- The naming service reads various entries and their attributes from the directory based on predefined identity.
- The authentication service authenticates a user’s name and password with the LDAP server to determine whether the correct password has been specified.

You can use Kerberos and LDAP at the same time to provide both authentication and naming services to the network. With Kerberos, you can support a single sign-on (SSO) environment in the enterprise. You can use the Kerberos identity system for querying LDAP naming data on a per-user or per-host basis.

If you use Kerberos to perform authentication, enable LDAP naming services as a requirement of the per-user mode. Kerberos can provide dual functions: It authenticates to the LDAP server, and the Kerberos identity for the user or host is used to authenticate to the directory. In this way, the same user identity that is used to authenticate to the system is also used to authenticate to the directory for lookups and updates. If required, you can use access control in the directory to limit the results out of the naming service.

Transport Layer Security

You can use Transport Layer Security (TLS) to secure communication between an LDAP client and the directory server and hence ensure both privacy and data integrity. The TLS protocol is a

superset of the Secure Sockets Layer (SSL) protocol. The LDAP naming service supports TLS security using either the `STARTTLS` operation on an opened LDAP connection or by opening a raw SSL (`LDAPS`) connection.

The requirements to use TLS are as follows:

- Configure the directory server and LDAP clients for TLS using `STARTTLS` and/or raw SSL.

  See Chapter 4, “Setting Up an Oracle Unified Directory Server or OpenLDAP Server” and Chapter 5, “Setting Up LDAP Clients”.
- Install the mandatory certificate PEM files and link databases as described in “How to Set Up TLS Security” on page 78.
- If necessary, update `/etc/openldap/ldap.conf` to include the location of the certificates used by LDAP with the `TLS_CACERTDIR` and `TLS_CACERT` options. See the `ldap.conf`(5oldap) man page for more details.

For information about setting up TLS security, see “Setting Up TLS Security” on page 77.

Client Credential Levels

The LDAP server authenticates LDAP clients according to the client credential level. You can assign any one of the following credential levels for LDAP clients:

- `anonymous` – With an `anonymous` credential level, you can access only the data that is available to everyone. No LDAP BIND operation occurs. An anonymous credential level is a high security risk. Any client can change information in the DIT to which the client has write access, including another user’s password or their own identity. Further, the `anonymous` level enables all clients to have read access to all LDAP naming entries and attributes.

  Note - Both OUD and OpenLDAP server enable you to implement security measures by restricting access based on, for example, IP addresses, DNS name, and authentication method. See “Understanding Access Control Model in Oracle Unified Directory” in *Oracle® Fusion Middleware Administering Oracle Unified Directory* and see “Security Considerations” in the *OpenLDAP Software 2.4 Administrator’s Guide*.

- `proxy` – With a `proxy` credential level, the client binds to a single shared set of LDAP bind credentials. The shared set is also called a *proxy account*. The proxy account can be any entry that is allowed to bind to the directory. The account requires sufficient access to perform the naming service functions on the LDAP server.

The proxy account is a shared-per-system resource, which means that users, including the root user, who are logged into a system using proxy access see the same information. You must configure the `proxyDN` and `proxyPassword` attributes on every client system that uses the `proxy` credential level. Further, the `proxyDN` must have the same `proxyPassword` on all of the LDAP servers.

The encrypted `proxyPassword` is stored locally on the client. If the password changes for a proxy user, you must update the password on every client system that uses that proxy user. Also, if you use password aging on LDAP accounts, make sure to exempt proxy users.

You can set up different proxies for different groups of clients. For example, you can configure a proxy that limits all the sales clients to access only the company-wide accessible directories and sales directories. Access to Human Resource directories with payroll information are forbidden. Or, in the most extreme cases, you can either assign different proxies to each client or assign just one proxy to all clients.

If you plan to set up multiple proxies for different clients, consider the choices carefully. Too few proxy agents can limit your ability to control user access to resources. However, too many proxies complicate the setup and maintenance of the system. You need to grant the appropriate rights to the proxy user depending on your environment. For more information about how to determine which authentication method to use, see “Storing Credential for LDAP Clients” on page 19.

The proxy credential level applies to all users and processes on any specific system. Users that need to use different naming policies must log in to different systems, or use the per-user authentication model.

- `proxy anonymous` – The `proxy anonymous` credential level is a multi-valued entry where more than one credential level is defined. With this level, a client first attempts to be authenticated by using its proxy identity. If the authentication fails because of user lockout or expired password, then the client uses anonymous access. Depending on how the directory is configured, different credential levels might be associated with different levels of service.
- `self` – The `self` credential level is also known as the per-user mode. This mode uses the Kerberos identity, called the principal, to perform a lookup for each system or user for authentication. With per-user authentication, the system administrator can use access control instructions (ACIs), access control lists (ACLs), roles, groups or other directory access control mechanisms to grant or deny access to specific naming-service data for specific users or systems.

  To use the per-user authentication model, the following configurations are required:

  - Deployment of the Kerberos single sign-on service
  - Support for the SASL and the SASL/GSSAPI authentication mechanism in one or more directory servers
  - Configuration of DNS, which Kerberos uses together with files to perform host name lookups

- Enabling of the `nscd` daemon

Enabling Shadow Data Updates

If the `enableShadowUpdate` switch is set to `true` on the client, administrator credentials are used to update the shadow data. Shadow data is stored in the `shadowAccount` object class on the directory server. Administrator credentials are defined by the values of the `adminDN` and `adminPassword` attributes, as described in "Defining LDAP Local Client Attributes" on page 73.

Administrator credentials have properties similar to `proxy` credentials. However, for administrator credentials, the user must have all privileges for the zone or have an effective UID of `root` to read or update the shadow data.

You can assign administrator credentials to any entry that is allowed to bind to the directory. However, do not use the same directory manager identity (`cn=Directory Manager`) of the LDAP server.

An entry with administrator credentials must have sufficient access to read and write the shadow data to the directory. The entry is a shared-per-system resource. Therefore, you must configure the `adminDN` and `adminPassword` attributes on every client.

The encrypted `adminPassword` is stored locally on the client. The admin password uses the same authentication methods that are configured for the client. All users and processes on a specific system uses the administrator credentials to read and update the shadow data.

Storing Credential for LDAP Clients

In the LDAP implementation, proxy credentials that are set during initialization are stored in the SMF repository instead of a client's profile. This implementation improves security surrounding a proxy's distinguished name (DN) and password information.

The SMF repository is `svc:/network/ldap/client`. It stores proxy information of clients that use a proxy identity. Likewise, shadow data updates of clients whose credential level is not `self` are also saved to this repository.

For clients that use per-user authentication, the Kerberos identity and Kerberos ticket information for each principal is used during authentication. The directory server maps the Kerberos principal to a DN and the Kerberos credentials are used to authenticate to that DN.

The directory server can use its access control mechanisms to allow or deny access to naming service data as necessary.

In this environment, Kerberos ticket information is used to authenticate to the directory server. The system does not store authentication DNs or passwords. Therefore, setting the `adminDN` and `adminPassword` attributes is unnecessary when you initialize the client with the `ldapclient` command.

Authentication Methods for the LDAP Naming Service

When you assign the `proxy` or `proxy-anonymous` credential level to a client, you must also select a method by which the proxy is authenticated. By default, the authentication method is `none`, which implies anonymous access. The authentication method might also have an associated transport security option.

The authentication method, like the credential level, can be multi-valued. For example, in the client profile, you can specify that the client tries to bind by using the `simple` method that is secured by TLS. If unsuccessful, the client would try to bind with the `sasl/digest-MD5` method. In this case, you would configure the `authenticationMethod` attribute as `tls:simple; sasl/digest-MD5`.

LDAP naming service supports some Simple Authentication and Security Layer (SASL) mechanisms. These mechanisms enable a secure password exchange without requiring TLS. However, these mechanisms do not provide data integrity or privacy. For information about SASL, see RFC 4422.

Note - Do not use the `CRAM-MD5` and `DIGEST-MD5` mechanisms without an encrypted TLS connection.

LDAP supports the following authentication mechanisms:

- `none` – The client does not authenticate to the directory. This method is equivalent to the `anonymous` credential level.
- `simple` – The client system sends the user's password in the clear to bind to the LDAP server. The password is subject to snooping unless the session is protected by IPsec. This method is easy to set up and all directory servers support it.

  Note - Oracle does not recommend using the `simple` authentication method in combination with the `none` credential level.

- `sasl/cram-MD5` – The LDAP session is not encrypted but the client's password is protected during authentication. Do not use this obsolete authentication method.
- `sasl/digest-MD5` – The client's password is protected during authentication but the session is not encrypted. The primary advantage of `digest-MD5` is that the password is not sent in clear text during authentication and is more secure than the `simple` authentication method. Refer to RFC 2831 for information on `digest-MD5`. `digest-MD5` is an improvement over `cram-MD5`.

  With `sasl/digest-MD5`, the authentication is secure but the session is not protected.
- `sasl/GSSAPI` – This authentication method is used in conjunction with the per-user mode to enable per-user lookups. A per-user `nscd` session with the client's credentials binds to the directory server by using the `sasl/GSSAPI` method and the client's Kerberos credentials. Access can be controlled in the directory server on a per-user basis.
- `tls:simple` – The client binds using the `simple` method and the session is encrypted. The password is protected.
- `tls:sasl/cram-MD5` – The LDAP session is encrypted and the client authenticates to the directory server using `sasl/cram-MD5`.
- `tls:sasl/digest-MD5` – The LDAP session is encrypted and the client authenticates to the directory server using `sasl/digest-MD5`.

The following table summarizes the various authentication methods and their characteristics.

TABLE 2 Authentication Methods

Method	Bind	Password over the wire	Password on OUD	Session
`none`	No	N/A	N/A	No encryption
`simple`	Yes	Clear	Any	No encryption
`sasl/digest-MD5`	Yes	Encryption	Clear	No encryption
`sasl/cram-MD5`	Yes	Encryption	N/A	No encryption
`sasl/GSSAPI`	Yes	Kerberos	Kerberos	Encryption
`tls:simple`	Yes	Encryption	Any	Encryption
`tls:sasl/cram-MD5`	Yes	Encryption	N/A	Encryption
`tls:sasl/digest-MD5`	Yes	Encryption	Clear	Encryption

For more information about the authentication methods that are supported for LDAP naming service, see the `ldapclient(8)` man page.

Specifying Authentication Methods for Specific Services in LDAP

The `serviceAuthenticationMethod` attribute determines the authentication method for a specific service. If this attribute is not set for the service, then the value of the `authenticationMethod` attribute is used.

Similarly, when the `enableShadowUpdate` switch is set to true, the `ldap_cachemgr` daemon uses the value for the `authenticationMethod` attribute if the `serviceAuthenticationMethod` attribute is not configured. The daemon does not use the `none` authentication method.

You can select authentication methods for the following services:

- `passwd-cmd` – Enables the `passwd` command to change the login password and password attributes. For more information, see the `passwd(1)` man page.
- `keyserv` – Enables the `chkey` and `newkey` utilities to create and change a user's Diffie-Hellman key pair. For more information, see the `chkey(1)` and `newkey(8)` man pages.
- `pam_ldap` – Enables authentication of users that use the `pam_ldap` service. The `pam_ldap` service supports account management.

Note - In per-user mode, the Kerberos service module is used as the authentication service and `ServiceAuthenticationMethod` is not needed.

The following example shows a section of a client profile in which the users use `sasl/digest-MD5` to authenticate to the directory server but use an SSL session to change the password.

```
serviceAuthenticationMethod=pam_ldap:sasl/digest-MD5
serviceAuthenticationMethod=passwd-cmd:tls:simple
```

Pluggable Authentication Methods

With the Pluggable Authentication Method (PAM) framework, you can choose among several authentication services, including the `pam_unix_*`, `pam_krb5`, and `pam_ldap_*` modules.

To use per-user authentication, you must enable `pam_krb5`. You can also use `pam_krb5` authentication if you do not assign the per-user credential level. If `proxy` or `anonymous`

credential levels are used to access directory server data, then you cannot restrict access to directory data on a per-user basis.

If you choose `anonymous` or `proxy` authentication, use the `pam_ldap` module instead of the equivalent `pam_unix_*` modules. The `pam_ldap` module is more flexible, supports stronger authentication methods, and can perform account management.

The following table summarizes the differences between authentication mechanisms.

TABLE 3 Authentication Behavior of PAM Modules

Event	`pam_unix_*`	`pam_ldap`	`pam_krb5`
Password Sent	Uses `passwd` service authentication method	Uses `passwd` service authentication method	Uses Kerberos single sign-on technology.
New Password Sent	Encrypted	No encryption (unless TLS is used)	Uses Kerberos. Passwords are not sent over the wire.
New Password Stored	`crypt` format	Password storage scheme defined on OUD	Uses Kerberos to manage passwords.
Requires password read?	Yes	No	No
`sasl/digest-MD5` compatibility after changing password	No. Password is not stored unencrypted. User cannot authenticate.	Yes. User can authenticate if the default storage scheme is set to `clear`.	No. Uses `sasl/GSSAPI`. There are no passwords over the wire and there are no passwords to be stored in the directory server except when using a Kerberos kdc that manages its password database in the LDAP directory server.
Password policy supported?	Yes. `enableShadowUpdate` must be set to `true`.	Yes, if configured.	See the `pam_krb5(7)` man page and Kerberos V5 Account Management Module.

LDAP Service Module

This section describes how to implement account management for clients that use `pam_ldap` module, and how to use the `pam_ldap` module to enable passwordless authentication. With passwordless authentication, users can log in with commands such as `ssh` and `sftp` without giving a password.

Enabling Account Management for Clients That Use the `pam_ldap` Module

In order for `pam_ldap` to work properly, you must properly configure the password and account lockout policy on the server. Use the `ldapmodify` command to configure the account management policy for the LDAP directory.

Ensure that the passwords for `proxy` users do not expire. If proxy passwords expire, clients using the `proxy` credential level cannot retrieve naming service information from the server. To ensure that `proxy` users have passwords that do not expire, modify the `proxy` accounts with the following script:

```
# ldapmodify -H ldapuri -D administrator-DN \
-w administrator-password <<EOF
dn: proxy-user-DN
DNchangetype: modify
replace: passwordexpirationtime
passwordexpirationtime: 20380119031407Z
EOF
```

The `pam_ldap` account management relies on the directory server to maintain and provide password aging and account expiration information for users. The directory server does not interpret the corresponding data from shadow entries to validate user accounts. Because the shadow data is not kept up to date by the LDAP naming service or the directory server, the modules should not grant access based on the shadow data. The shadow data is retrieved using the `proxy` identity. Therefore, do not allow `proxy` users to have read access to the `userPassword` attribute. Denying `proxy` users read access to `userPassword` prevents the PAM service from making an invalid account validation.

Configuring Oracle Unified Directory for Passwordless Public Key Authentication

The `1.3.6.1.4.1.42.2.27.9.5.8` control on the directory server is enabled by default. This control only applies to OUD. To modify the default control configuration, add ACIs on the directory server as shown in the following example:

```
dn: oid=1.3.6.1.4.1.42.2.27.9.5.8,cn=features,cn=config
objectClass: top
objectClass: directoryServerFeature
oid:1.3.6.1.4.1.42.2.27.9.5.8
cn:Password Policy Account Usable Request Control
aci: (targetattr != "aci")(version 3.0; acl "Account Usable";
```

```
allow (read, search, compare, proxy)
(groupdn = "ldap:///cn=Administrators,cn=config");)
creatorsName: cn=server,cn=plugins,cn=config
modifiersName: cn=server,cn=plugins,cn=config
```

The pam_ldap module does not read the userPassword attribute. If no client uses UNIX authentication, granting read access to the userPassword attribute is unnecessary. Similarly, the pam_ldap module does not support none as an authentication method.

Note - If the simple authentication method is used, the userPassword attribute can be read unencrypted by third parties.

The serviceAuthenticationMethod attribute, if defined, determines the manner in which the user binds to the LDAP server. Otherwise, the authenticationMethod attribute is used. After the pam_ldap module successfully binds to the server with the user's identity and password, the module authenticates the user. You can perform account management and retrieve the account status of users while the user is logging in without authenticating to the directory server.

Configuring OpenLDAP Server for Passwordless Public Key Authentication

The pam_ldap module can retrieve and use the account status of users from a properly configured OpenLDAP server in a limited way: The pam_ldap module will determine only whether a user account has been permanently locked or might have been locked due to repeated bind failures.

The following are required configuration for the OpenLDAP server:

- Use the Password Policy overlay with the pwdLockout option set to TRUE.
- Enable the LDAP client's proxy user to read the ppolicy operational attributes, specifically pwdAccountLockedTime, that are stored in the user's entry.

See the slapo-ppolicy(5oldap) man page for descriptions of the operational attributes.

Configuring Microsoft Active Directory Server for Passwordless Public Key Authentication

The pam_ldap module can also retrieve the account status of users from an AD server to allow passwordless public key authentication for commands such as ssh and sftp.

AD must delegate `ReaduserAccountControl` permission to the security group to which the LDAP client's `proxy` user belongs. The `pam_ldap` module uses the `proxy` user to retrieve account status information. For each user, the `user` account control attributes to be read are: `userAccountControl`, `msDS-User-Account-Control-Computed`, `msDS-UserPasswordExpiryTimeComputed`, `accountExpires`, and `pwdLastSet`. Consult the Microsoft Active Directory Server documentation for how to delegate the `ReaduserAccountControl` permission.

`pam_unix_*` Service Modules

If the `/etc/pam.conf` file is unconfigured, UNIX authentication is enabled by default.

Note - The `pam_unix` module has been removed and is no longer supported in Oracle Solaris. The module has been replaced by a different set of service modules that provides equivalent or greater functionality. In this guide, `pam_unix` refers to the modules that provide equivalent functionality, not to the `pam_unix` module itself.

The following modules provide the equivalent functionality as the original `pam_unix` module. The modules are listed by using their corresponding man pages.

- `pam_authtok_check`(7)
- `pam_authtok_get`(7)
- `pam_authtok_store`(7)
- `pam_dhkeys`(7)
- `pam_passwd_auth`(7)
- `pam_unix_account`(7)
- `pam_unix_auth`(7)
- `pam_unix_cred`(7)
- `pam_unix_session`(7)

The `pam_unix_*` modules use the following UNIX authentication model:

1. The client retrieves the user's encrypted password from the name service.
2. The user is prompted for the password.
3. The user's password is encrypted.
4. The client compares the two encrypted passwords to determine whether the user should be authenticated.

The `pam_unix_*` modules have the following restrictions:

- The password must be stored in UNIX `crypt` format.
- The `userPassword` attribute must be readable by the name service.

  For example, if you set the credential level to `anonymous`, then anyone must be able to read the `userPassword` attribute. Similarly, if you set the credential level to `proxy`, then the proxy user must be able to read the `userPassword` attribute.

Note - UNIX authentication is incompatible with the `sasl/digest-MD5` authentication method. In OUD, passwords must be stored unencrypted to use `digest-MD5`.

The `pam_unix_account` module supports account management when the `enableShadowUpdate` switch is set to `true`. The controls for a remote LDAP user account are applied in the same manner that controls are applied to a local user account that is defined in the `passwd` and `shadow` files. For the LDAP account in `enableShadowUpdate` mode, the system updates and uses the shadow data on the LDAP server for password aging and account locking. The shadow data of the local account only applies to the local client system, while the shadow data of an LDAP user account applies to the user on all client systems.

You can check the password history only for the local client and not for an LDAP user account.

Kerberos Service Module

For information about Kerberos, see *Managing Kerberos in Oracle Solaris 11.4* and the `pam_krb5`(7) man page.

Changing Passwords That Use PAM

Use the `passwd` command to change a password. If the `enableShadowUpdate` switch is not enabled, the `userPassword` attribute must be writable by the user as well as by the administrator credentials. The `serviceAuthenticationMethod` for `passwd-cmd` overrides the `authenticationMethod` for this operation. Depending on the authentication method, the current password might be unencrypted.

In UNIX authentication, the new `userPassword` attribute is encrypted with the UNIX `crypt` format. The attribute is tagged before being written to LDAP. Thus, the new password is encrypted regardless of the authentication method used to bind to the server. For more information, see the `pam_authtok_store`(7) man page.

If the `enableShadowUpdate` switch is enabled, the `pam_unix_*` modules update the related shadow information when the user password is changed. Similarly, the `pam_unix_*` modules

update the `shadow` fields in the local `shadow` files that the modules update when the local user password is changed.

To support password update, the `pam_ldap` module can use the `pam_authtok_store` module with the `server_policy` option. When you use `pam_authtok_store`, the new password is sent to the LDAP server unencrypted. Use TLS to ensure privacy. Otherwise, the new `userPassword` becomes subject to snooping.

If you set an untagged password with OUD, the software uses the `passwordStorageScheme` attribute to encrypt the password. For more information about the `passwordStorageScheme` attribute, see Security, Access Control, and Password Policies in *Oracle® Fusion Middleware Administering Oracle Unified Directory*.

If NIS or any other client that uses UNIX authentication uses LDAP as a repository, then you must configure the `passwordStorageScheme` attribute with `crypt`. Also, if you use `sasl/digest-MD5` LDAP authentication with the OUD, you must configure the `passwordStorageScheme` attribute to clear text.

LDAP Account Management

With `pam_krb5` performing account and password management, the Kerberos environment manages all of the account, password, account lockout, and other account management details.

If you do not use `pam_krb5`, then configure the LDAP naming service to take advantage of the password and account lockout policy support in OUD. You can configure `pam_ldap` to support user account management. With the proper PAM configuration, the `passwd` command enforces password syntax rules set by the OUD password policy. However, do not enable account management for `proxy` accounts.

The following account management features are supported by `pam_ldap`. These features depend on the OUD password and account lockout policy configuration. You can enable the following account management features:

- Password aging and expiration notification – Users must change their passwords according to a schedule. Otherwise, the password expires and user authentication fails.

  Users are warned whenever they log in within the expiration warning period. The warning includes the remaining time before password expiration.
- Password syntax checking – New passwords must meet the minimum password length requirements. A password must not match the value of the `uid`, `cn`, `sn`, or `mail` attributes in the user's directory entry.

- Password history checking – Users cannot reuse passwords. LDAP administrators can configure the number of passwords kept in the server's history list.
- User account lockout - A user account can be locked out after a specified number of repeated authentication failures. Users can also be locked out if their accounts are inactivated by an administrator. Authentication failure continues until the account lockout time is passed or the administrator reactivates the account.

These account management features work only with the OUD. For information about configuring the password and account lockout policy on the LDAP server, see Directory Server Password Policy in *Oracle® Fusion Middleware Administering Oracle Unified Directory*.

Before configuring the password and account lockout policy on the OUD, make sure all hosts use the most recent version of the LDAP client with `pam_ldap` account management. Additionally, make sure the clients have a properly configured `pam.conf` file. Otherwise, the LDAP naming service fails when `proxy` or user passwords expire.

LDAP Account Management With the `pam_unix_*` Modules

The LDAP naming service supports the full functionality of the `passwd` command and the `pam_unix_*` modules in the files naming service. If the `enableShadowUpdate` switch is enabled, account management functionality becomes available to both local accounts and LDAP accounts. The functionality includes password aging, account expiry and notification, and failed login account locking. Also, LDAP supports the `-dluNfnwx` options of the `passwd` command. The `enableShadowUpdate` switch enables the implementation of consistent account management for users who are defined in both the files and the LDAP scope.

The `pam_ldap` and the `pam_unix_*` modules are incompatible. The `pam_ldap` module requires that passwords be modifiable by users, but the `pam_unix_*` modules do not allow the users to modify passwords. Therefore, you cannot use the two modules together in the same LDAP naming domain. Either all clients use the `pam_ldap` module or all clients use the `pam_unix_*` modules. As a consequence of this limitation, you might need to use a dedicated LDAP server in cases where a web or email application, for example, might require users to change their own passwords on the LDAP server.

Implementing `enableShadowUpdate` also requires that the administrator credential (`adminDN` and `adminPassword`) is stored locally on every client in the `svc:/network/ldap/client` service.

Do not change the `/etc/pam.conf` file to use the `pam_unix_*` modules for account management. The default `/etc/pam.conf` file is sufficient.

♦ ♦ ♦ CHAPTER 3

Planning Requirements for LDAP Naming Services

This chapter discusses the high-level planning that you must do before beginning the server and client setup and installation processes.

This chapter covers the following topics:

LDAP Planning Overview

An LDAP client uses the collection of configuration information in the LDAP client profile to access naming service information from the LDAP server. You must specify the configuration information when you build the profile on the LDAP server. During the server setup, you are prompted for the configuration information. Some of the information that is prompted is required, while other information is optional. In most cases, you accept the default values that are already provided. The individual types of information that are prompted for the profile are called client attributes.

As you gather the configuration information for the profile, you can refer to the template checklists used for configuring LDAP in "Checklists for Configuring LDAP" on page 43.

The LDAP client profile attributes are as follows:

- `profileName` – Specifies the profile name that is used by clients to select the profile. The default value is `default`. See the `ldapclient(8)` man page for a description.

- `preferredServerList` – Specifies the host addresses of the preferred servers as a space-separated list of server addresses. Do not use host names of the servers in this list. The servers in this list are tried in order *before* those in `defaultServerList` until a successful connection is made. This attribute has no default value. You must specify at least one server in either `preferredServerList` or `defaultServerList`.

Note - If you are using host names to define both `defaultServerList` and `preferredServerList`, then you must not use LDAP for host server lookup searches. Do not configure the `config/host` property of the `svc:/network/name-service/switch` service with the value `ldap`. For more information about LDAP and service management facility (SMF), see "LDAP and the Service Management Facility" on page 72.

- `defaultServerList` – Specifies the host addresses of the default servers as a space-separated list of server addresses. Do not use host names of the servers in this list. After the servers in `preferredServerList` are tried, the default servers on the client's subnet are tried, followed by the remaining default servers, until a connection is made. You must specify at least one server in either `preferredServerList` or `defaultServerList`. The servers in this list are tried only after the servers in the preferred server list. This attribute has no default value.
- `defaultSearchBase` – Specifies the DN relative to which to locate the well-known containers. This attribute has no default value. However, this value can be overridden for a given service by the `serviceSearchDescriptor` attribute.
- `defaultSearchScope` – Defines the scope of a database search by an LDAP client. It can be overridden by the `serviceSearchDescriptor` attribute. The possible values are `one` or `sub`. The default value is a single-level search.
- `authenticationMethod` – Identifies the method of authentication used by the LDAP client. The default value is `none`. For more information, see "Authentication Methods for the LDAP Naming Service" on page 20.
- `credentialLevel` – Identifies the type of credentials an LDAP client must use to authenticate. The possible values are `anonymous`, `proxy`, or `self`. `self` is also known as "per-user". The default value is `anonymous`
- `serviceSearchDescriptor` – Defines how and where an LDAP client should search for a naming database, for example, whether the LDAP client should look in one or more points in the DIT. By default, no SSDs are defined.
- `serviceAuthenticationMethod` – Defines the authentication method used by an LDAP client for the specified service. By default, no service authentication methods are defined. If a service does not have `serviceAuthenticationMethod` defined, it defaults to the value of `authenticationMethod`.
- `attributeMap` – Defines the attribute mappings that the LDAP client uses. By default, no `attributeMap` is defined.

- `objectclassMap` – Defines object class mappings that the LDAP client uses. By default, no `objectclassMap` is defined.
- `searchTimeLimit` – Specifies the maximum time, in seconds, that an LDAP client must allow for a search to complete before timing out. This value does not affect the time the LDAP server will allow for a search to complete. The default value is `30` seconds.
- `bindTimeLimit` – Specifies maximum time in seconds an LDAP client must allow to bind with a server before timing out. The default value is `30` seconds.
- `followReferrals` – Specifies whether an LDAP client should follow an LDAP referral. Possible values are `TRUE` or `FALSE`. The default value is `TRUE`.
- `profileTTL` – Specifies time between refreshes of the LDAP client profile from the LDAP server by the `ldap_cachemgr` daemon. The default value is `43200` seconds or 12 hours. If given a value of 0, the profile will never be refreshed. For more information, see the `ldap_cachemgr`(8) man page.

The LDAP client profile attributes are automatically set up when you run the `ldapservercfg` command on the server. Additional profiles can be generated by using `ldapclient genprofile`, as described in the `ldapclient`(8) man page.

You can use the `ldapclient` command to set up local client attributes. For more information, see “Defining LDAP Local Client Attributes” on page 73.

Planning the Configuration of the LDAP Client Profile

To set up the LDAP naming service, you must first plan the configuration of the LDAP client profile. The default values of the profile attributes suffice for most networks. However, based on the network topology, you might specify non-default values for some profile attributes. This section describes the different attributes that you might want to configure.

LDAP Network Model

When planning the LDAP network model, you must determine the physical servers to be deployed for the LDAP naming service. To ensure availability and performance, each subnet of the network must have one LDAP server to service the LDAP clients in that subnet. When planning for this model, you should consider the following factors:

- Number of systems to be deployed as LDAP servers

  Which servers are designated master servers, and which servers are replicas that serve as backups?

- The manner of access to the servers

  Should all the LDAP servers have equal priority for access by LDAP client requests? Or, should the servers have different priorities and those with higher priorities be accessed first? If access to the servers is not equal, list the order in which these servers are accessed.

  The information that you specify is managed by the `defaultServerList` and `preferredServerList` attributes. Note the following guidelines for the server list:

  - Use LDAP servers, not a concentrator, balancer, or pool.
  - Use multiple LDAP servers.
  - If you are using SSL/TLS, the host names must match the certificate names.
  - Host names must resolve to IP addresses.
- Timeout factors

  Determine the timeout values as follows:

  - `bindTimeLimit` attribute determines how long a TCP connect request continues before the request is dropped.
  - `searchTimeLimit` attribute determines how long an LDAP search operation continues before the search is cancelled.
  - `profileTTL` attribute determines how often an LDAP client downloads profiles from the servers.

  For example, in a slow network, you might increase the length of time for searching and for allowing TCP connect requests. In a development environment, you might limit the frequency of downloading a profile by an LDAP client.

Directory Information Tree

The LDAP naming service uses a default Directory Information Tree (DIT) to store information. The DIT is based on an LDAP schema.

The DIT consists of containers of information that are hierarchically structured. The structure follows the standard LDAP schema described in RFC 2307 and RFC 4876.

The default structure of the DIT suffices for most network setups to implement LDAP. With the default structure, you only need to determine the following:

- The base node distinguished name (DN) of the tree that naming service will search for information about a specific domain. The `defaultSearchBase` attribute manages the base node information.

- The scope of search that a naming service lookup functionality should perform. The scope can cover either only one level below the DN, or the entire subtree below the DN. This information is managed by the attribute `defaultSearchScope`.

A DIT can also have a more complicated structure for storing data. For example, you can store the data about user accounts in different parts of the DIT. You should determine how to customize the behavior of the search operation such as the base DN, the scope, and the filters to use that overrides the default search sequence. The customized search sequence information is managed by the attributes `serviceSearchDescriptor`, `attributeMap`, and `objectclassMap`. For a detailed explanation about customizing the search sequence operation, see "Service Search Descriptors and Schema Mapping" on page 37.

Multiple servers can serve a single DIT. In this setup, the subtrees of a DIT might be distributed across multiple servers. Therefore, you must further configure LDAP servers to redirect LDAP client requests to the appropriate LDAP servers which can provide the requested information. The `followReferrals` attribute manages the information about how to redirect LDAP client requests to the correct server.

Having a single LDAP server providing all the naming data for a specific domain is the typical and recommended setup. Even in this scenario, however, you can still configure the `followReferrals` attribute to direct LDAP clients to read-only replica servers for most of the information requests. Access to a master server to perform read and write operations is not typically provided. With a referral configuration, you prevent the master server from overload.

Security Considerations

For the security of LDAP operations that process requests for directory information, consider the following:

- The manner by which LDAP clients identify themselves to access information, which is determined by the credential level that you specify for the clients. The credential level is managed by the `credentialLevel` attribute, to which you can assign one of the following values:
  - `anonymous`
  - `proxy`
  - `proxy anonymous`
  - `self`

  For detailed descriptions of each of these values, see "Client Credential Levels" on page 17.
- The method of authenticating the LDAP client, which is managed by the `authenticationMethod` attribute. You can specify the authentication method by assigning one of the following options:

- `none`
- `simple`
- `sasl/digest-MD5`
- `sasl/cram-MD5`
- `sasl/GSSAPI`
- `tls:simple`
- `tls:sasl/cram-MD5`
- `tls:sasl/digest-MD5`

  For detailed descriptions of each of these values, see "Authentication Methods for the LDAP Naming Service" on page 20.

In addition to the credential level to assign to LDAP clients as well as the authentication method to use, you should also consider the following:

- Whether to use Kerberos and per-user authentication
- Value to specify for the servers' `passwordStorageScheme` attribute
- Setup of access control information

  For more information about ACIs, consult the administration guide for the version of OUD that you are using.
- Whether to use the `pam_unix_*` or `pam_ldap` module to perform LDAP account management

  This consideration is related to whether the LDAP naming service is compatible with NIS.

Planning the Deployment of LDAP Master and Replica Servers

Oracle recommend the use of replication to provide high availability.

You can configure multiple master servers to store read-write copies of the same directories. For large-scale enterprise deployments, you must use multi-master replication.

You must establish a conflict resolution policy because updating the same directories in different master servers can cause conflicts.

For information about how to set up replica servers, see "Understanding the Oracle Unified Directory Replication Model" in *Oracle® Fusion Middleware Administering Oracle Unified Directory* and see "Replication" in the *OpenLDAP Software 2.4 Administrator's Guide*.

Planning the LDAP Data Population

After the LDAP server has been configured with the proper DIT and schema, you need to populate the DIT with data. The source of the data are the `/etc` files in multiple systems. Consider the following methods for populating the DIT:

- Merge the `/etc` files of a specific data type into a single file for that data type. For example, merge all `/etc/passwd` files from different systems into a single `/etc/passwd` file. You can populate the server from the single host that stores all the merged `/etc` files.
- Populate the server by using the appropriate command from each LDAP client system that accesses the directory server.

Service Search Descriptors and Schema Mapping

The LDAP naming service can use the DIT only if it is structured in a certain way. If required you can use SSDs to enable the LDAP naming service to search in locations other than the default location. Additionally, you can define attributes and object classes in place of the ones specified by the default schema. Use the `ldaplist -v` command to list the default filters.

Note - The default filters are listed in "Default Filters Used by the LDAP Naming Service" on page 39.

If you use schema mapping, you must make sure that the syntax of the mapped attribute is consistent with the attribute it is mapped to. For example, the single-valued attributes must map to single-valued attributes and the attributes must have the same syntax. Also, ensure that the mapped object classes have the correct mandatory attributes.

About Service Search Descriptors

The `serviceSearchDescriptor` attribute defines how and where an LDAP naming service client should search for information for a particular service. The `serviceSearchDescriptor` contains a service name followed by one or more semicolon-separated base-scope-filter triples. These base-scope-filter triples are used to define searches only for the specific service and are searched in order. If multiple base-scope-filters are specified for a given service, when that service looks for a particular entry, it will search in each base with the specified scope and filter.

Note - The default location is not searched for a service (database) with an SSD unless it is included in the SSD. Unpredictable behavior will result if multiple SSDs are specified for a service.

In the following example, the LDAP naming service client performs a single-level search in `ou=west,dc=example,dc=com` followed by a single-level search in `ou=east,dc=example,dc=com` for the `passwd` service. To look up the `passwd` data for a user's `username`, the default LDAP filter `(&(objectClass=posixAccount)(uid=username))` is used for each `BaseDN`.

```
serviceSearchDescriptor: passwd:ou=west,dc=example,dc=com;ou=east,
dc=example,dc=com
```

In the following example, the LDAP naming service client would perform a subtree search in `ou=west,dc=example,dc=com` for the `passwd` service. To look up the `passwd` data for user `username`, the subtree `ou=west,dc=example,dc=com` would be searched with the LDAP filter `(&(fulltimeEmployee=TRUE)(uid=username))`.

```
serviceSearchDescriptor: passwd:ou=west,dc=example,
dc=com?sub?fulltimeEmployee=TRUE
```

You can also associate multiple containers with a particular service type. In the following example, the service search descriptor specifies searching for the password entries in three containers.

```
ou=myuser,dc=example,dc=com
ou=newuser,dc=example,dc=com
ou=extuser,dc=example,dc=com
```

Note that a trailing ',' in the example implies that the `defaultSearchBase` is appended to the relative base in the SSD.

```
defaultSearchBase: dc=example,dc=com
serviceSearchDescriptor: \
passwd:ou=myuser,;ou=newuser,;ou=extuser,dc=example,dc=com
```

`attributeMap` Attributes

The LDAP naming service enables one or more attribute names to be remapped for any of its services. If you map an attribute, you must be sure that the attribute has the same meaning and syntax as the original attribute. Note that mapping the `userPassword` attribute might cause problems.

Consider using schema mappings in situations where you want to map attributes in an existing directory server. If you have user names that differ only in case, you must map the `uid` attribute, which ignores case, to an attribute that does not ignore case.

The format for this attribute is `service:attribute-name=mapped-attribute-name`.

If you want to map more than one attribute for a given service, you can define multiple `attributeMap` attributes.

In the following example, the `employeeName` and `home` attributes would be used whenever the `uid` and `homeDirectory` attributes would be used for the `passwd` service.

```
attributeMap: passwd:uid=employeeName
attributeMap: passwd:homeDirectory=home
```

You can map the `passwd` service's `gecos` attribute to several attributes, as shown in the following example.

```
attributeMap: gecos=cn sn title
```

This example maps the `gecos` values to a space separated list of the `cn`, `sn`, and `title` attribute values.

`objectclassMap` Attribute

The LDAP naming service enables object classes to be remapped for any of its services. If you want to map more than one object class for a given service, you can define multiple `objectclassMap` attributes. In the following example, the `myUnixAccount` object class is used whenever the `posixAccount` object class is used.

```
objectclassMap: passwd:posixAccount=myUnixAccount
```

Default Filters Used by the LDAP Naming Service

If you do not specify a parameter for a given service using an SSD, the default filter is used. To list the default filters for a given service, use the `ldaplist` command with the `-v` option.

In the following example, `filter=(&(objectclass=iphost)(cn=abcde)`defines the default filters.

```
database=hosts
filter=(&(objectclass=iphost)(cn=abcde)
user data=(&(%s) (cn=abcde))
```

The `ldaplist` command generates the following list of default filters, where `%s` signifies a string and `%d`, a number.

```
hosts
(&(objectclass=iphost)(cn=%s))
--------------
passwd
(&(objectclass=posixaccount)(uid=%s))
--------------
```

```
services
(&(objectclass=ipservice)(cn=%s))
--------------
group
(&(objectclass=posixgroup)(cn=%s))
--------------
netgroup
(&(objectclass=nisnetgroup)(cn=%s))
--------------
networks
(&(objectclass=ipnetwork)(ipnetworknumber=%s))
--------------
netmasks
(&(objectclass=ipnetwork)(ipnetworknumber=%s))
--------------
rpc
(&(objectclass=oncrpc)(cn=%s))
--------------
protocols
(&(objectclass=ipprotocol)(cn=%s))
--------------
bootparams
(&(objectclass=bootableDevice)(cn=%s))
--------------
ethers
(&(objectclass=ieee802Device)(cn=%s))
--------------
publickey
(&(objectclass=niskeyobject)(cn=%s))
or
(&(objectclass=niskeyobject)(uidnumber=%d))
--------------
aliases
(&(objectclass=mailGroup)(cn=%s))
--------------
```

The following table lists the LDAP filters used in the `getXbyY` call.

TABLE 4 LDAP Filters Used in `getXbyY` Calls

Filter	Definition
`bootparamByName`	`(&(objectClass=bootableDevice)(cn=%s))`
`etherByHost`	`(&(objectClass=ieee802Device)(cn=%s))`
`etherByEther`	`(&(objectClass=ieee802Device)(macAddress=%s))`
`groupByName`	`(&(objectClass=posixGroup)(cn=%s))`
`groupByGID`	`(&(objectClass=posixGroup)(gidNumber=%ld))`

Filter	Definition
groupByMember	(&(objectClass=posixGroup)(memberUid=%s))
hostsByName	(&(objectClass=ipHost)(cn=%s))
hostsByAddr	(&(objectClass=ipHost)(ipHostNumber=%s))
keyByUID	(&(objectClass=nisKeyObject)(uidNumber=%s))
keyByHost	(&(objectClass=nisKeyObject)(cn=%s))
netByName	(&(objectClass=ipNetwork)(cn=%s))
netByAddr	(&(objectClass=ipNetwork)(ipNetworkNumber=%s))
nisgroupMember	(membernisnetgroup=%s)
maskByNet	(&(objectClass=ipNetwork)(ipNetworkNumber=%s))
printerByName	(&(objectClass=sunPrinter)(|(printer-name=%s) (printer-aliases=%s)))
projectByName	(&(objectClass=SolarisProject)(SolarisProjectName=%s))
projectByID	(&(objectClass=SolarisProject)(SolarisProjectID=%ld))
protoByName	(&(objectClass=ipProtocol)(cn=%s))
protoByNumber	(&(objectClass=ipProtocol)(ipProtocolNumber=%d))
passwordByName	(&(objectClass=posixAccount)(uid=%s))
passwordByNumber	(&(objectClass=posixAccount)(uidNumber=%ld))
rpcByName	(&(objectClass=oncRpc)(cn=%s))
rpcByNumber	(&(objectClass=oncRpc)(oncRpcNumber=%d))
serverByName	(&(objectClass=ipService)(cn=%s))
serverByPort	(&(objectClass=ipService)(ipServicePort=%ld))
serverByNameAndProto	(&(objectClass=ipService)(cn=%s)(ipServiceProtocol=%s))
specialByNameserver	(ipServiceProtocol=%s))
ByPortAndProto	(&(objectClass=shadowAccount)(uid=%s))
netgroupByTriple	(&(objectClass=nisNetGroup)(cn=%s))
netgroupByMember	(&(objectClass=nisNetGroup)(cn=%s))
authName	(&(objectClass=SolarisAuthAttr)(cn=%s))
auditUserByName	(&(objectClass=SolarisAuditUser)(uid=%s))
execByName	(&(objectClass=SolarisExecAttr)(cn=%s) (SolarisKernelSecurityPolicy=%s) (SolarisProfileType=%s))
execByPolicy	(&(objectClass=SolarisExecAttr)(SolarisProfileId=%s) (SolarisKernelSecurityPolicy=%s)(SolarisProfileType=%s))
profileByName	(&(objectClass=SolarisProfAttr)(cn=%s))
userByName	(&(objectClass=SolarisUserAttr)(uid=%s))

The following table lists the `getent` attribute filters.

TABLE 5 `getent` Attribute Filters

Filter	Definition
aliases	(objectClass=rfc822MailGroup)
auth_attr	(objectClass=SolarisAuthAttr)
audit_user	(objectClass=SolarisAuditUser)
exec_attr	(objectClass=SolarisExecAttr)
group	(objectClass=posixGroup)
hosts	(objectClass=ipHost)
networks	(objectClass=ipNetwork)
prof_attr	(objectClass=SolarisProfAttr)
protocols	(objectClass=ipProtocol)
passwd	(objectClass=posixAccount)
printers	(objectClass=sunPrinter)
rpc	(objectClass=oncRpc)
services	(objectClass=ipService)
shadow	(objectclass=shadowAccount)
project	(objectClass=SolarisProject)
usr_attr	(objectClass=SolarisUserAttr)

Default Client Profile Attributes for LDAP Implementation

There are several significant attributes that you might configure to implement the LDAP naming service. Note that not all of these attributes require configuration. Of the following attributes, only `defaultServerList` and `defaultSearchBase` require you to provide values. For other attributes, you can accept the default values or leave the other attributes without any configuration.

- `profileName`
- `defaultServerList`
- `preferredServerList`
- `bindTimeLimit`
- `searchTimeLimit`
- `profileTTL`
- `defaultSearchBase`
- `defaultSearchScope`

- `serviceSearchDescriptor`
- `attributeMap`
- `objectclassMap`
- `followReferrals`
- `credentialLevel`
- `authenticationMethod`
- `serviceCredentialLevel`
- `serviceAuthenticationMethod`

Checklists for Configuring LDAP

TABLE 6 Checklist for Server Variable Definitions

Variable	Definition for _____ Network
Port number at which the directory server instance is installed (389)	
Name of the LDAP server	
Replica servers (*IP number*:*port number*)	
Directory manager [`dn: cn=directory manager`]	
Domain name to be served	
Maximum time (in seconds) to process client requests before timing out	
Maximum number of entries returned for each search request	

TABLE 7 Checklist for Client Profile Variable Definitions

Variable	Definition for _____ Network
Profile name	
Server list (defaults to the local subnet)	
Preferred server list (listed in order of which server to try first, second, and so on)	
Search scope (number of levels down through the directory tree). Possible values are '`One`' or '`Sub`'.	
Credential used to gain access to server. The default is `anonymous`.	
Follow Referrals? (Referrals are a pointer to another server if the main server is unavailable.) The default is `no`.	

Variable	Definition for _____ Network
Search time limit (in seconds) for waiting for the server to return information. The default is `30` seconds.	
Bind time limit (in seconds) for contacting the server. The default is `30` seconds.	
Authentication method. Default is `none`.	

♦ ♦ ♦ CHAPTER 4

Setting Up an Oracle Unified Directory Server or OpenLDAP Server

This chapter describes how to configure Oracle Unified Directory (OUD) and OpenLDAP servers to support LDAP clients for Oracle Solaris.

The LDAP Server Configuration Utility

The `ldapservercfg` utility configures and prepares a compatible directory server, installed on the system where `ldapservercfg` is run, to serve LDAP clients. OUD and OpenLDAP are compatible directory servers.

The directory server is configured to support the following. See also `ldap(7)`.

- Oracle Solaris naming services as defined in the `/usr/share/lib/ldif/nameservice.ldif` file.
- Kerberos services as defined in the `kerberos.ldif` file.
- RFC2037bis-02 Directory Information Tree (DIT) structure.
- A default LDAP client configuration profile.

Note - The `ldapservercfg` utility cannot configure a remote server.

The type of server to configure is appended as a command line option to `ldapservercfg`, as described in the following sections:

- "Setting Up the Oracle Unified Directory Server" on page 46
- "Setting Up the OpenLDAP Server" on page 52

Setting Up the Oracle Unified Directory Server

To set up an OUD server, specify `oud` as the `ldapservercfg` *server-type* operand. Ensure that the OUD server has been installed and enabled according to the procedures documented in Setting Up Oracle Unified Directory as a Directory Server in *Oracle® Fusion Middleware Installing Oracle Unified Directory*. Make sure a security feature such as SSL/TLS is enabled in OUD if you want to access the OUD server by using the security mechanism. See Security, Access Control, and Password Policies in *Oracle® Fusion Middleware Administering Oracle Unified Directory*.

Note - The OUD server must already be installed and configured on the server where you are running the `ldapservercfg` utility before you can perform this procedure.

▼ How to Configure the Oracle Unified Directory Server

1. **Become an administrator.**
   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

2. **Create the `admin` user to install the OUD server.**

   ```
   # useradd -u 11424 -m -d /export/home/admin -s /usr/bin/bash admin
   ```

3. **As the `admin` user, create the OUD server instance.**
   Provide information as prompted. See "Creating the OUD Server Instance" on page 46.

4. **Run the `ldapservercfg` utility to configure the OUD server.**
   Provide information as prompted. See "Configuring the OUD Server" on page 49.

Creating the OUD Server Instance

For information about OUD administration, see *Oracle® Fusion Middleware Installing Oracle Unified Directory*. For more information about OUD setup, see Setting Up the Directory Server.

```
$ su - admin
$ /export/home/admin/Oracle/Middleware/Oracle_OUD1/oud-setup -i
OUD Instance location successfully created - /export/home/admin/Oracle/Middleware/
Oracle_OUD1/../asinst_1"
```

```
Oracle Unified Directory 11.1.2.3.0
Please wait while the setup program initializes...

What would you like to use as the initial root user DN for the Directory
Server? [cn=Directory Manager]:
Please provide the password to use for the initial root user:
Please re-enter the password for confirmation:

On which port would you like the Directory Server to accept connections from
LDAP clients? [1389]:

On which port would you like the Administration Connector to accept
connections? [4444]:

Do you want to create base DNs in the server? (yes / no) [yes]:

Provide the base DN for the directory data: [dc=example,dc=com]:
Options for populating the database:

    1)  Only create the base entry
    2)  Leave the database empty
    3)  Import data from an LDIF file
    4)  Load automatically-generated sample data

Enter choice [1]:

Do you want to enable SSL? (yes / no) [no]: yes
On which port would you like the Directory Server to accept connections from
LDAPS clients? [1636]:

Do you want to enable Start TLS? (yes / no) [no]: yes
Certificate server options:
    1)  Generate self-signed certificate (recommended for testing purposes
        only)
    2)  Use an existing certificate located on a Java Key Store (JKS)
    3)  Use an existing certificate located on a JCEKS key store
    4)  Use an existing certificate located on a PKCS#12 key store
    5)  Use an existing certificate on a PKCS#11 token

Enter choice [1]:
Provide the fully-qualified host name or IP address that will be used to
generate the self-signed certificate
[abc.example.com]:

Specify the Oracle components with which the server integrates.  It is
recommended to choose the option covering only your requirements.
```

```
    1)  No Integration
    2)  DIP (Directory Integration Platform)
    3)  Generic: Database Net Services, EBS and DIP
    4)  EUS (Enterprise User Security), Database Net Services, EBS and DIP

    c)  cancel

Enter choice [1]:

How do you want the OUD server to be tuned?

    1)  Use specific Java Virtual Machine arguments
    2)  Use the default Java Virtual Machine settings
    3)  Provide the Java heap size to be used by the server
    4)  Provide the percentage of system memory to be used by the server
    5)  Provide the size of system memory to be used by the server

Enter choice [1]:

How do you want the off-line tools (import-ldif, export-ldif, verify-index and
rebuild-index) to be tuned?

    1)  Use specific Java Virtual Machine arguments
    2)  Use the default Java Virtual Machine settings
    3)  Automatic Tuning
    4)  Provide the Java heap size to be used by the off-line tools

Enter choice [1]: 1

Do you want to start the server when the configuration is completed? (yes /
no) [yes]:


Setup Summary
=============
LDAP Listener Port:                 1389
Administration Connector Port:      4444
LDAP Secure Access:                 Enable StartTLS
                                    Enable SSL on LDAP Port 1636
                                    Create a new Self-Signed Certificate
Root User DN:                       cn=Directory Manager
Directory Data:                     Create New Base DN dc=example,dc=com
                                    Base DN Data: Only Create Base Entry
                                    (dc=example,dc=com)
Integration with Oracle components: No Integration
Server Runtime Settings:            Use the default Java Virtual Machine
                                    settings
Off-line Tools Runtime Settings:    Use the default Java Virtual Machine
```

```
                                    settings
Start Server when the configuration is completed
What would you like to do?
    1)  Set up the server with the parameters above
    2)  Provide the setup parameters again
    3)  Print equivalent non-interactive command-line
    4)  Cancel and exit
Enter choice [1]:
See /export/home/admin/Oracle/Middleware/asinst_1/OUD/logs/oud-setup for a
detailed log of this operation.
Configuring Directory Server ..... Done.
Configuring Certificates ..... Done.
Creating Base Entry dc=example,dc=com ..... Done.
Starting Directory Server ............. Done.
To see basic server configuration status and configuration you can launch
/export/home/admin/Oracle/Middleware/asinst_1/OUD/bin/status
~$
```

Configuring the OUD Server

Ensure that SSL/TLS is enabled on the LDAP server if you want to access the server by using the corresponding security mechanism.

```
~$ /usr/sbin/ldapservercfg oud
Enter the administration port number for DS (h=help):  [4444]
Enter the port number for DS (h=help):  [389] 1389
Enter the directory manager CN: [Directory Manager]
Enter password for Directory Manager:
The following are existing base DNs

    [1] dc=example,dc=com

Please select LDAP base DN: (1-1) [1]
  Validating LDAP Base DN and Suffix ...
  Found valid LDAP entry: dc=example,dc=com
  Found an existing backend "userRoot"

  sasl/GSSAPI is not supported by this LDAP server.
  If you want to enable sasl/GSSAPI authentication, please refer to server
  manual guide to setup sasl/GSSAPI and Kerberos first.

Enter the profile name (h=help): [default]
Default server list (h=help): [abc.example.com:1389]
Choose desired search scope (one, sub, h=help):  [one]

The following are the supported credential levels:
```

```
  1  anonymous
  2  proxy

Choose Credential level [h=help]: [2]
Enter CN for proxy agent:  [proxyagent]
Enter password for proxy agent:
Re-enter password:

The following are the supported Authentication Methods:
  1  simple
  2  tls:simple

Choose Authentication Method: [2]
Do you want the clients to follow referrals? (yes/[no])  [yes]
Do you want to store passwords in "crypt" format? (yes/[no])  [yes]
Do you want to enable shadow update? (yes/[no])  [yes]
Enter CN for the administrator: [admin]
Enter password for the administrator:
Re-enter password:
Do you wish to setup Service Search Descriptors? (yes/[no])  [no]
No replicated server found for base dn "dc=example,dc=com".

               Summary of Configuration

  1  Profile name to create        : default
  2  Base DN to setup              : dc=example,dc=com
  3  Default Search Scope          : one
  4  Default Server List           : abc.example.com:1389
  5  Credential Level              : proxy
  6  Authentication Method         : tls:simple
  7  Enable crypt password storage : True
  8  Enable shadow update          : True
  9  Service Search Descriptors Menu

Enter config value to change: (1-9 0=commit changes) [0]
WARNING: About to start committing changes. (yes/[no]) yes

  == Begin Directory Server Configuration ==

  1. Doing compatible configuration...
     Configuring server "abc.example.com" ...
  2. Schema have been updated.
  3. Adding suffix...
     Suffix dc=example,dc=com already existed.
  4. NisDomainObject added to "dc=example,dc=com".
  5. ACI "Anonymous access" was added for suffix "dc=example,dc=com".
  6. ACI "Allow self entry modification except for some attributes" was added for suffix
 "dc=example,dc=com".
```

```
  7. ACI "Configuration Administrator" was added for suffix "dc=example,dc=com".
  8. ACI "Configuration Administrators Group" was added for suffix "dc=example,dc=com".
     Entry "people" was added into the directory.
     Entry "group" was added into the directory.
     Entry "rpc" was added into the directory.
     Entry "protocols" was added into the directory.
     Entry "networks" was added into the directory.
     Entry "aliases" was added into the directory.
     Entry "hosts" was added into the directory.
     Entry "services" was added into the directory.
     Entry "ethers" was added into the directory.
     Entry "profile" was added into the directory.
     Entry "printers" was added into the directory.
     Entry "netgroup" was added into the directory.
     Entry "projects" was added into the directory.
     Entry "SolarisAuthAttr" was added into the directory.
     Entry "SolarisProfAttr" was added into the directory.
     Entry "Timezone" was added into the directory.
     Entry "ipTnet" was added into the directory.
  9. Top level "ou" containers complete.
     Entry "auto_home" was added into the directory.
     Entry "auto_direct" was added into the directory.
     Entry "auto_master" was added into the directory.
     Entry "auto_shared" was added into the directory.
  10. automount maps: ['auto_home', 'auto_direct', 'auto_master', 'auto_shared']
 processed.
  11. ACI for dc=example,dc=com modified to disable self modification.
  12. Proxy Agent cn=proxyagent,ou=profile,dc=example,dc=com added.
  13. Administrator identity cn=admin,ou=profile,dc=example,dc=com added.
  14. Add password-reset privilege to cn=admin,ou=profile,dc=example,dc=com.
  Proxy ACI LDAP_Naming_Services_proxy_password_read does not exist for dc=example,
dc=com.
  15. Give cn=admin,ou=profile,dc=example,dc=com read/write access to shadow data.
  16. Non-Admin access to shadow data denied.
  17. Generated client profile and loaded on server.
  18. Setup indexes ...
     Checking indexes for server "abc.example.com":
     Will create index uidNumber (eq, pres)
     Will create index ipNetworkNumber (eq, pres)
     Will create index gidnumber (eq, pres)
     Will create index oncrpcnumber (eq, pres)
     Will create index automountKey (eq, pres)
     Will create index ipHostNumber (eq, pres, sub)
     Will create index membernisnetgroup (eq, pres, sub)
     Will create index nisnetgrouptriple (eq, pres, sub)

     Adding Access Control Information for VLV Index...
```

```
   Will create vlv_index example.com.getgrent
   Will create vlv_index example.com.gethostent
   Will create vlv_index example.com.getnetent
   Will create vlv_index example.com.getpwent
   Will create vlv_index example.com.getrpcent
   Will create vlv_index example.com.getspent
   Will create vlv_index example.com.getauhoent
   Will create vlv_index example.com.getsoluent
   Will create vlv_index example.com.getsolquent
   Will create vlv_index example.com.getauthent
   Will create vlv_index example.com.getexecent
   Will create vlv_index example.com.getprofent
   Will create vlv_index example.com.getmailent
   Will create vlv_index example.com.getbootent
   Will create vlv_index example.com.getethent
   Will create vlv_index example.com.getngrpent
   Will create vlv_index example.com.getipnent
   Will create vlv_index example.com.getmaskent
   Will create vlv_index example.com.getprent
   Will create vlv_index example.com.getip4ent
   Will create vlv_index example.com.getip6ent

19. Creating indexes...
   Configuring server "abc.example.com" ...
20. Rebuilding indexes...
21. Verifying indexes...

== End Directory Server Configuration ==

Setup LDAP server is complete.
```

Setting Up the OpenLDAP Server

To set up an OpenLDAP server, specify `openldap` as the `ldapservercfg` *server-type* operand. Use the OpenLDAP rights profile in order to have the authorizations and privileges to configure and enable the `slapd` Standalone LDAP daemon.

The `ldapservercfg` utility reads initial parameter values from the `svc:/network/ldap/server:openldap` service configuration and deploys OpenLDAP using an Online Configuration (OLC), also known as `cn=config` or `slapd-config`. See the description of the configuration repository in the `slapd-config`(5oldap) man page, and see the `-F` option in the `slapd`(8) man page.

The OpenLDAP server is configured to accept unencrypted connections on port 389, encrypted connections with STARTTLS on port 389, and encrypted connections using raw TLS on port 636. When the server configuration is successful, the configuration properties in `svc:/network/ldap/server:openldap` are updated.

Note - OpenLDAP must already be installed on the server where you are running the `ldapservercfg` utility before you can configure the OpenLDAP server to work with Oracle Solaris LDAP clients.

▼ How to Pre-Configure a Newly Installed System to be an OpenLDAP Server

Perform this procedure if you are running the `ldapservercfg` utility on a newly installed Oracle Solaris system.

1. **Make sure the `service/network/ldap/openldap` package is installed.**

Note - The OpenLDAP server package, `service/network/ldap/openldap`, must be installed on the same Oracle Solaris server where you will execute the `ldapservercfg` utility.

```
$ pkg list service/network/ldap/openldap
```

If the `service/network/ldap/openldap` package is not installed, use the following command:

```
# pkg install service/network/ldap/openldap
```

2. **Check whether the Domain Name System (DNS) service is working correctly.**

Use the following command to verify that the server's Fully Qualified Domain Name (FQDN) is available:

```
$ host hostname
```

Verify that the `hostname` resolves to its FQDN name in DNS:

```
# uname -n
server.example.com
# host server
server.example.com has address 192.0.2.0
```

For more information about the DNS service, see Chapter 3, "Managing DNS Server and Client Services" in *Working With Oracle Solaris 11.4 Directory and Naming Services: DNS and NIS*.

How to Migrate Existing OpenLDAP Server Configuration

To transition to a new version of OpenLDAP, databases must be exported to LDAP Data Interchange Format (LDIF) and imported following the system upgrade.

Oracle Solaris packaging of OpenLDAP no longer provides support for the Berkeley DB (BDB) format static backends. Import data using the default Lightning Memory-Mapped Database (LMDB) format.

Note - The first steps of this procedure must be completed before upgrading.

1. **Make sure that the `openldap` service is in the disabled state.**

   Use the `svcs` command to check the state of the `openldap` service. If the state is any state other than `disabled`, disable the `openldap` service:

   ```
   # svcadm disable ldap/server:openldap
   ```

2. **Dump the OpenLDAP database to LDIF.**

   ```
   # mkdir -p /var/share/openldap
   # slapcat -l /var/share/openldap/data.ldif
   ```

   See the `slapcat`(8oldap) man page for more information.

3. **Perform the system upgrade and boot into the upgraded boot environment.**

   If the `openldap` service is in the `maintenance` state (if it was enabled at the time of system upgrade), disable the `openldap` service.

4. **Update the `slapd.conf` configuration file.**

   This step is not necessary if the system is using OLC, which stores data in the `/etc/openldap/slapd.d` directory.

   Edit the `/etc/openldap/slapd.conf` file to update the following configuration. You might want to back up your existing `slapd.conf` file first.

   - `TLSProtocolMin`: Minimum protocol version. Make sure `TLSProtocolMin` is set to `3.2`.

     ```
     TLSProtocolMin        3.2
     ```

   - `database`: OpenLDAP database type. Make sure `database` is set to `mdb`.

     ```
     database      mdb
     ```

5. **Remove the old database in `/var/openldap/openldap-data`.**
   You might want to back up your existing database files first.
   Do not remove the directory itself, which is owned by user and group `openldap`.

   ```
   # rm -rf /var/openldap/openldap-data/*
   ```

6. **Import LDIF data.**
   As user `openldap`, use the `slapadd` command to import the LDIF data.
   Execute the import from a directory that is accessible to the `openldap` user to prevent `getcwd` errors from `slapadd`.

   ```
   # cd /tmp
   $ su openldap -c "slapadd -l /var/share/openldap/data.ldif"
   ```

   For more information, see the `slapadd`(8oldap) man page.

7. **Enable the `openldap` service.**

   ```
   # svcadm enable ldap/server:openldap
   ```

8. **Check the service status.**

   ```
   # svcs ldap/server:openldap
   ```

   If the service status is not `online`, check the service log file to troubleshoot.

   ```
   # svcs -Lv ldap/server:openldap
   ```

Configuring the OpenLDAP Server for LDAP Clients

The `ldapservercfg` utility can configure an OpenLDAP server instance interactively or with default settings read from an SMF service instance.

Ensure that the following requirements are met:

- The `/usr/sbin/ldapservercfg` utility is installed.
- The `python-ldap-27` package is installed.
- The `dnspython` package is installed.

By default, the Oracle Solaris OpenLDAP server instance uses Online Configuration (OLC) instead of the legacy configuration file `slapd.conf`.

- When OLC exists, the `ldap/server:openldap` service starts the `slapd` daemon using the OLC configuration. After running `ldapservercfg openldap`, LDAP configuration is contained in `/etc/openldap/slapd.d`.
- When OLC is not available, the `ldap/server:openldap` service uses a plain configuration file, `/etc/openldap/slapd.conf`. This `slapd.conf` file is useful for manually configuring OpenLDAP or for migrating from another server.

▼ How to Configure an OpenLDAP Server With Settings from SMF

The `-a` option of `ldapservercfg` can be used to configure OpenLDAP with no human interaction; `ldapservercfg` reads required configuration values from property values of the `ldap/server:openldap` SMF service.

If the `ldap/server:openldap` service is online and no OpenLDAP configuration exists in `/etc/openldap`, then the service executes the following command:

```
# ldapservercfg -a openldap
```

The server is configured using property values of the service. If the values of the service properties have not been changed from their default values, the directory is configured to serve the distinguished name `dc=example,dc=com`. See the default configuration shown below in Step 2.

1. Check that the service is disabled and no OpenLDAP configuration exists.

```
$ svcs ldap/server:openldap
STATE          STIME    FMRI
disabled       Jun_17   svc:/network/ldap/server:openldap
```

The following files and directories must not exist or must be empty:

- The `/etc/openldap/slapd.conf` legacy configuration file does not exist.
- The `/etc/openldap/slapd.d` directory does not exist.
- The `/etc/openldap/certs` directory does not exist.
- The `/var/openldap/openldap-data` directory is empty.

2. Check the default SMF properties for the `openldap` service.

a. Check credential properties.

```
$ svcprop -p cred ldap/server:openldap
cred/admin_cn astring admin
```

```
cred/admin_passwd astring ""
cred/backend_cn astring Manager
cred/backend_passwd astring ""
cred/proxy_cn astring proxyagent
cred/proxy_passwd astring ""
cred/read_authorization astring solaris.smf.read.name-service.ldap.server
cred/stability astring Evolving
cred/value_authorization astring solaris.smf.value.name-service.ldap.server
```

If `backend_passwd` is not specified, `root` password is used. If `admin_passwd` is not specified, the `admin` account is not created. If `proxy_passwd` is not specified, the `proxyagent` account is not created.

b. Check the LDAP Name Service Profile data.

These values are used to configure the DIT and default profiles used by `ldapclient`.

```
$ svcprop -p profile/default ldap/server:openldap
profile/default/authentication_method astring tls:simple
profile/default/credential_level astring proxy
profile/default/search_base astring dc=example,dc=com
profile/default/search_scope astring one
profile/default/server_list astring ""
profile/default/service_search_descriptor astring ""
profile/default/value_authorization astring solaris.smf.value.name-service.ldap.
server
```

For more information about these values, see the `ldapclient`(8) and `ldapservercfg`(8) man pages.

3. (Optional) Configure the `openldap` service as required.

This step is necessary unless you want a server for `dc=example,dc=com`. See “Configuring `openldap` Service Properties” on page 62.

4. Use the `ldapservercfg` utility to configure the OpenLDAP server.

The following example of configuration using `openldap` service property values shows performing this configuration as the `openldap` user. You can perform this OpenLDAP server configuration as any user that is assigned the `OpenLDAP Server Administration` rights profile.

```
$ su - openldap
$ /usr/sbin/ldapservercfg -a openldap
  TLS CA certificate directory: /etc/openldap/certs
  TLS CA certificate file: /etc/certs/ca-certificates.crt
  TLS public certificate file: /etc/openldap/certs/certdb.pem
  TLS private key file: /etc/openldap/certs/server.key  Starting server...Succeeded.
```

```
  The server is set as a master server.

WARNING: The client profile credential_level is proxy, but there is no
         'cred/proxy_passwd' provided, can't create proxy account, will
         use anonymous credential level instead.


               Summary of Configuration

  1  Profile name to create        : default
  2  Base DN to setup              : dc=example,dc=com
  3  Default Search Scope          : one
  4  Default Server List           : abc.example.com
  5  Credential Level              : anonymous
  6  Authentication Method         : tls:none
  7  Enable crypt password storage : True
  8  Enable shadow update          : False
  9  Service Search Descriptors Menu


  == Begin Directory Server Configuration ==

  1. Schema "{4}solaris" has been created.
  2. Schema "{5}kerberos" has been created.
  3. Adding suffix...
  4. Suffix dc=example,dc=com successfully created.
  5. ACIs was added for suffix "dc=example,dc=com".
     Entry "people" was added into the directory.
     Entry "group" was added into the directory.
     Entry "rpc" was added into the directory.
     Entry "protocols" was added into the directory.
     Entry "networks" was added into the directory.
     Entry "aliases" was added into the directory.
     Entry "hosts" was added into the directory.
     Entry "services" was added into the directory.
     Entry "ethers" was added into the directory.
     Entry "profile" was added into the directory.
     Entry "printers" was added into the directory.
     Entry "netgroup" was added into the directory.
     Entry "projects" was added into the directory.
     Entry "SolarisAuthAttr" was added into the directory.
     Entry "SolarisProfAttr" was added into the directory.
     Entry "Timezone" was added into the directory.
     Entry "ipTnet" was added into the directory.
  6. Top level "ou" containers complete.
     Entry "auto_home" was added into the directory.
     Entry "auto_direct" was added into the directory.
     Entry "auto_master" was added into the directory.
```

```
    Entry "auto_shared" was added into the directory.
  7. automount maps: ['auto_home', 'auto_direct', 'auto_master', 'auto_shared']
processed.
  8. Generated client profile and loaded on server.
  9. Overlay ppolicy has been already activated.
    ppolicy overlay added successfully.
    Default password policy was added into the directory.
  10. Setup indexes ...
    Checking indexes for server "abc.example.com":
  11. Index uidNumber successfully created.
  12. Index ipNetworkNumber successfully created.
  13. Index gidnumber successfully created.
  14. Index oncrpcnumber successfully created.
  15. Index automountKey successfully created.
  16. Index uid successfully created.
  17. Index krbPrincipalName successfully created.
  18. Index membernisnetgroup successfully created.

  == End Directory Server Configuration ==


  Setup LDAP server is complete.
```

How to Configure OpenLDAP Server Interactively

The `ldapservercfg openldap` command (no `-a` option) prompts you for settings. Default values are taken from `openldap` service property values as discussed in "Configuring `openldap` Service Properties" on page 62.

1. **Switch to the `openldap` user.**

   The following example of interactive configuration using `openldap` service property values shows performing this configuration as the `openldap` user. You can perform this OpenLDAP server configuration as any user that is assigned the `OpenLDAP Server Administration` rights profile.

   ```
   $ su - openldap
   ```

2. **Use the `ldapservercfg` utility to configure the OpenLDAP server.**

   Provide information as prompted.

   ```
   $ /usr/sbin/ldapservercfg openldap

   Do you want to configure this server as a master server? (yes/[no]) [yes]
   Do you want to start server with TLS support? (yes/[no]) [no] yes
     TLS CA certificate directory: /etc/openldap/certs
   ```

```
  TLS CA certificate file: /etc/certs/ca-certificates.crt
  TLS public certificate file: /etc/openldap/certs/certdb.pem
  TLS private key file: /etc/openldap/certs/server.key
Starting server...Succeeded.
Enter LDAP Search Base: [dc=example,dc=com] dc=scdev,dc=sfbay,dc=sun,dc=com
Enter the directory manager CN: [Manager]
Enter password for cn=Manager,dc=scdev,dc=sfbay,dc=sun,dc=com:
Re-enter password:

  The server is set as a master server.

The following are the supported credential levels:
  1 anonymous
  2 proxy


Choose Credential level [h=help]: [1] 2
Enter CN for proxy agent: [proxyagent]
Enter password for proxy agent:
Re-enter password:

The following are the supported Authentication Methods:
  1 simple
  2 tls:simple

Choose Authentication Method: [1] 2
Do you want to enable shadow update? (yes/[no]) [yes]
Enter CN for the administrator: [admin]
Enter password for the administrator:
Re-enter password:

Do you wish to setup Service Search Descriptors? (yes/[no])  [no]

Summary of Configuration

  1  Profile name to create        : default
  2  Base DN to setup              : dc=scdev,dc=sfbay,dc=sun,dc=com
  3  Default Search Scope          : sub
  4  Default Server List           : abc.example.com
  5  Credential Level              : proxy
  6  Authentication Method         : tls:simple
  7  Enable crypt password storage : True
  8  Enable shadow update          : True
  9  Service Search Descriptors Menu

Enter config value to change: (1-9 0=commit changes) [0]
WARNING: About to start committing changes. (yes/[no]) yes
```

```
== Begin Directory Server Configuration ==

1. Schema "{4} solaris" has been created.
2. Schema "{5} kerberos" has been created.
3. Adding suffix...
4. Suffix dc=scdev,dc=sfbay,dc=sun,dc=com successfully created.
5. ACIs was added for suffix "dc=scdev,dc=sfbay,dc=sun,dc=com".
   Entry "people" was added into the directory.
   Entry "group" was added into the directory.
   Entry "rpc" was added into the directory.
   Entry "protocols" was added into the directory.
   Entry "networks" was added into the directory.
   Entry "aliases" was added into the directory.
   Entry "hosts" was added into the directory.
   Entry "services" was added into the directory.
   Entry "ethers" was added into the directory.
   Entry "profile" was added into the directory.
   Entry "printers" was added into the directory.
   Entry "netgroup" was added into the directory.
   Entry "projects" was added into the directory.
   Entry "SolarisAuthAttr" was added into the directory.
   Entry "SolarisProfAttr" was added into the directory.
   Entry "Timezone" was added into the directory.
   Entry "ipTnet" was added into the directory.
6. Top level "ou" containers complete.
   Entry "auto_home" was added into the directory.
   Entry "auto_direct" was added into the directory.
   Entry "auto_master" was added into the directory.
   Entry "auto_shared" was added into the directory.
7. automount maps: ['auto_home', 'auto_direct', 'auto_master', 'auto_shared'] processed.
8. Proxy Agent cn=proxyagent,ou=profile,dc=scdev,dc=sfbay,dc=sun,dc=com added.
9. Administrator identity cn=admin,ou=profile,dc=scdev,dc=sfbay,dc=sun,dc=com added.
10. Give "cn=admin,ou=profile,dc=scdev,dc=sfbay,dc=sun,dc=com" read/write access to
 shadow data.
11. Generated client profile and loaded on server.
12. Overlay ppolicy has been already activated.
   ppolicy overlay added successfully.
   Default password policy was added into the directory.
13. Setup indexes ...
   Checking indexes for server "abc.example.com":
14. Index uidNumber successfully created.
15. Index ipNetworkNumber successfully created.
16. Index gidnumber successfully created.
17. Index oncrpcnumber successfully created.
18. Index automountKey successfully created.
19. Index uid successfully created.
20. Index krbPrincipalName successfully created.
21. Index membernisnetgroup successfully created.
```

```
== End Directory Server Configuration ==

Setup LDAP server is complete.
```

Configuring `openldap` Service Properties

The `ldapservercfg` utility uses the values of `openldap` service properties to configure the server, both when used interactively and when used non-interactively.

▼ How to Specify Credentials

Use this procedure to change the credentials names and passwords before initial configuration.

The password (`passwd`) properties are not used when running `ldapservercfg` interactively.

1. Create password hashes.

Use `slappasswd` to create password hashes. See the `slappasswd`(8oldap) man page.

```
# slappasswd -h "{SSHA}"
New password: yoursecret
Reenter new password: yoursecret
 {SSHA}password-hash
```

2. Store the hashes in relevant SMF properties.

Use the `editprop` tool as shown in "Using `editprop` to Modify `openldap` Service Properties" on page 63, or use the `svccfg setprop` command as shown in the following example.

```
# svccfg s ldap/server:openldap
svc:/network/ldap/server:openldap> setprop cred/backend_passwd = astring:
 "{SSHA}password-hash"
svc:/network/ldap/server:openldap> setprop cred/proxy_passwd = astring: "{SSHA}password-
hash"
svc:/network/ldap/server:openldap> setprop cred/admin_passwd = astring: "{SSHA}password-
hash"
svc:/network/ldap/server:openldap> refresh
svc:/network/ldap/server:openldap> quit
```

Using `editprop` to Modify `openldap` Service Properties

Using the `svccfg editprop` command to modify service property values presents all properties that you can edit and their current values in your editor (`$EDITOR`). For more information about `editprop`, see "Invoking a Property Editor" in *Managing System Services in Oracle Solaris 11.4*.

The following command opens an editor on the properties of the `openldap` service:

```
# svccfg -s ldap/server:openldap editprop
```

When you issue the preceding command, your editor opens with content very similar to the following content:

```
##
## Change property values by removing the leading '#' from the
## appropriate lines and editing the values. svccfg subcommands
## such as delprop can also be added to the script.
##

## Property group "config"
## The following properties are defined in the selected instance
## (svc:/network/ldap/server:openldap)

##
## Hostname and Port
##
# setprop config/urls = astring: ("ldap:///" "ldaps:///" "ldapi:///")
# setprop config/value_authorization = astring: solaris.smf.value.name-service.ldap.
server

## Property group "cred"
## The following properties are defined in the selected instance
## (svc:/network/ldap/server:openldap)

##
## Admin Common Name
##
# setprop cred/admin_cn = astring: admin

##
## Admin Password
##
# setprop cred/admin_passwd =

##
## Backend Common Name
##
```

```
# setprop cred/backend_cn = astring: Manager

##
## Backend Password
##
# setprop cred/backend_passwd =

##
## Proxy Common Name
##
# setprop cred/proxy_cn = astring: proxyagent

##
## Proxy Password
##
# setprop cred/proxy_passwd =
# setprop cred/read_authorization = astring: solaris.smf.read.name-service.ldap.server
# setprop cred/stability = astring: Evolving
# setprop cred/value_authorization = astring: solaris.smf.value.name-service.ldap.server

## Property group "profile"

## Property group "profile/default"
## The following properties are defined in the selected instance
## (svc:/network/ldap/server:openldap)

##
## Authentication Method(s)
##
# setprop profile/default/authentication_method = astring: tls:simple

##
## Credential Level(s)
##
# setprop profile/default/credential_level = astring: proxy

##
## Search Base
##
# setprop profile/default/search_base = astring: "dc=example,dc=com"

##
## Search Scope
##
# setprop profile/default/search_scope = astring: one

##
## Server List
```

```
##
# setprop profile/default/server_list =

##
## Service Search Descriptor(s)
##
# setprop profile/default/service_search_descriptor =
# setprop profile/default/value_authorization = astring: solaris.smf.value.name-service.
ldap.server

## Uncomment to apply these changes to this instance.
# refresh
```

The following partial file shows how to change the passwords and the search base:

```
##
## Admin Password
##
setprop cred/admin_passwd = astring: {SSHA}j9X9QwojnqelDhdl2qR6+OeWkkNVCRSD

##
## Backend Password
##
setprop cred/backend_passwd = astring: {SSHA}j9X9QwojnqelDhdl2qR6+OeWkkNVCRSD

##
## Proxy Password
##
setprop cred/proxy_passwd = astring: {SSHA}j9X9QwojnqelDhdl2qR6+OeWkkNVCRSD

##
## Search Base
##
setprop profile/default/search_base = astring: "dc=sample,dc=example,dc=com"

## Uncomment to apply these changes to this instance.
refresh
```

After you exit your editor, use the following command to verify the changes you made:

```
# svcprop -p cred -p profile ldap/server:openldap
cred/admin_cn astring admin
cred/admin_passwd astring {SSHA}j9X9QwojnqelDhdl2qR6+OeWkkNVCRSD
cred/backend_cn astring Manager
cred/backend_passwd astring {SSHA}j9X9QwojnqelDhdl2qR6+OeWkkNVCRSD
cred/proxy_cn astring proxyagent
cred/proxy_passwd astring {SSHA}j9X9QwojnqelDhdl2qR6+OeWkkNVCRSD
cred/read_authorization astring solaris.smf.read.name-service.ldap.server
cred/stability astring Evolving
```

```
cred/value_authorization astring solaris.smf.value.name-service.ldap.server
profile/default/authentication_method astring tls:simple
profile/default/credential_level astring proxy
profile/default/search_base astring dc=sample,dc=example,dc=com
profile/default/search_scope astring one
profile/default/server_list astring
profile/default/service_search_descriptor astring
profile/default/value_authorization astring solaris.smf.value.name-service.ldap.server
```

The output from the following `ldapservercfg` command shows that the changes to the credential and profile properties have been applied:

```
$ /usr/sbin/ldapservercfg -a openldap
TLS CA certificate directory: /etc/openldap/certs
TLS CA certificate file: /etc/certs/ca-certificates.crt
TLS public certificate file: /etc/openldap/certs/certdb.pem
TLS private key file: /etc/openldap/certs/server.key
Starting server…Succeeded.

The server is set as a master server.

             Summary of Configuration

1 Profile name to create : default
2 Base DN to setup : dc=sample,dc=example,dc=com
3 Default Search Scope : one
4 Default Server List : abc.example.com
5 Credential Level : proxy
6 Authentication Method : tls:simple
7 Enable crypt password storage : True
8 Enable shadow update : True
9 Service Search Descriptors Menu

= Begin Directory Server Configuration =

1. Schema "{4}solaris" has been created.
2. Schema "{5}kerberos" has been created.
3. Adding suffix…
4. Suffix dc=sample,dc=example,dc=com successfully created.
5. ACIs was added for suffix "dc=sample,dc=example,dc=com".
   Entry "people" was added into the directory.
   Entry "group" was added into the directory.
   Entry "rpc" was added into the directory.
   Entry "protocols" was added into the directory.
   Entry "networks" was added into the directory.
   Entry "aliases" was added into the directory.
   Entry "hosts" was added into the directory.
   Entry "services" was added into the directory.
```

```
   Entry "ethers" was added into the directory.
   Entry "profile" was added into the directory.
   Entry "printers" was added into the directory.
   Entry "netgroup" was added into the directory.
   Entry "projects" was added into the directory.
   Entry "SolarisAuthAttr" was added into the directory.
   Entry "SolarisProfAttr" was added into the directory.
   Entry "Timezone" was added into the directory.
   Entry "ipTnet" was added into the directory.
6. Top level "ou" containers complete.
   Entry "auto_home" was added into the directory.
   Entry "auto_direct" was added into the directory.
   Entry "auto_master" was added into the directory.
   Entry "auto_shared" was added into the directory.
7. automount maps: ['auto_home', 'auto_direct', 'auto_master', 'auto_shared'] processed.
8. Proxy Agent cn=proxyagent,ou=profile,dc=sample,dc=example,dc=com added.
9. Administrator identity cn=admin,ou=profile,dc=sample,dc=example,dc=com added.
10. Give "cn=admin,ou=profile,dc=sample,dc=example,dc=com" read/write access to shadow
 data.
11. Generated client profile and loaded on server.
12. Overlay ppolicy has been already activated.
   ppolicy overlay added successfully.
   Default password policy was added into the directory.
13. Setup indexes …
   Checking indexes for server "abc.example.com":
14. Index uidNumber successfully created.
15. Index ipNetworkNumber successfully created.
16. Index gidnumber successfully created.
17. Index oncrpcnumber successfully created.
18. Index automountKey successfully created.
19. Index uid successfully created.
20. Index krbPrincipalName successfully created.
21. Index membernisnetgroup successfully created.

= End Directory Server Configuration =

Setup LDAP server is complete.
```

Troubleshooting OpenLDAP Server Configuration

This section shows:

- How to use the `ldapservercfg` command to determine whether a system is or has been previously configured as an OpenLDAP server.
- How to remove OpenLDAP server configuration.

`ldapservercfg` Warns that the System is Already Configured

The following are true for a newly installed system or a system that has not been configured as an OpenLDAP server:

- The `ldap/server:openldap` service instance is in the `disabled` state.

  ```
  $ svcs ldap/server:openldap
  STATE          STIME    FMRI
  disabled       Jun_17   svc:/network/ldap/server:openldap
  ```

  If the `ldap/server:openldap` service instance was previously enabled and online, then the service created the default configuration as specified in its SMF properties.
- The following files and directories do not exist or are empty:
  - The `/etc/openldap/slapd.conf` legacy configuration file does not exist.
  - The `/etc/openldap/slapd.d` directory does not exist.
  - The `/etc/openldap/certs` directory does not exist.
  - The `/var/openldap/openldap-data` directory is empty.

  If any of these files or directories exist, then the system might have been used as an OpenLDAP server previously.

When you run the `ldapservercfg` command on a system that has previously been configured as an OpenLDAP server, you are warned and prompted to confirm whether you want to continue with this reconfiguration:

```
*******************************************************************************
WARNING: The OpenLDAP server has already been configured.
         If you want to rerun the initial configuration, be sure to do
         necessary backups using slapcat, i.e.

         /usr/sbin/slapcat -F /etc/openldap/slapd.d -b \
             "dc=example,dc=com" > example.com.ldif

         The tool will do the following actions:
         /usr/sbin/svcadm disable -s ldap/server:openldap
         rm -rf /etc/openldap/slapd.d
         rm -rf /etc/openldap/certs
         rm -rf /var/openldap/openldap-data/*.mdb
*******************************************************************************

WARNING: About to reset the server configuration. (yes/[no])
```

`ldapservercfg` Shows Existing Base DNs

The following occurs when the server has already been configured interactively:

```
# su openldap -c '/usr/sbin/ldapservercfg  openldap'
Do you want to configure this server as a master server? (yes/[no])  [yes]
The following are existing base DNs

    [1] dc=example,dc=com

Please select LDAP base DN: (1-1) [1]
```

▼ How to Remove OpenLDAP Configuration

1. **Disable the `openldap` service instance.**

   ```
   # svcadm disable ldap/server:openldap
   ```

2. **Back up data and configuration.**

   Back up to an LDIF file in a safe place, as shown in the following example:

   ```
   # slapcat -n 1 -l /var/tmp/dit.ldif
   # slapcat -n 0 -l /var/tmp/config.ldif
   ```

3. **Remove configuration and data files.**

   ```
   # rm -rf /etc/openldap/slapd.d /var/openldap/openldap-data/*
   ```

4. **(Optional) Remove TLS certificates.**

   If you intend to reconfigure the server, and you already have clients that are using these certificates, you might choose to keep these certificates.

   ```
   # rm -rf /etc/openldap/certs
   ```

♦ ♦ ♦ CHAPTER 5

Setting Up LDAP Clients

This chapter describes how to set up an LDAP naming service client. It covers the following topics:

Requirements for LDAP Client Setup

Oracle Solaris clients using LDAP as a naming service must meet the following requirements:

- The client's domain name must be provided by the LDAP server.
- The name service switch must point to LDAP for the required services.
- The client must be configured with all the parameters that define its behavior.
- `ldap_cachemgr` must be running on the client.
- At least one server for which a client is configured must be running.
- For the `ldapclient` services to function, ensure that the `svc:/system/name-service/cache` service is enabled and that the `nscd` daemon is running.

The `ldapclient` utility performs all of the listed configuration steps except for starting the server. This chapter provides examples of how to use the `ldapclient` utility to set up an LDAP client and how to use the various other LDAP utilities to get information about an LDAP client.

Note - Because LDAP and NIS use the same domain name component that is defined in the `network/nis/domain` service, Oracle Solaris does not support a configuration in which an NIS client and a native LDAP client coexist on the same client system.

LDAP and the Service Management Facility

The Oracle Solaris SMF manages the LDAP client service. For more information about SMF, refer to *Managing System Services in Oracle Solaris 11.4*. For more information about the commands used to modify the SMF service, see the svcadm(8) and svcs(1) man pages.

The features of SMF that relate to administering the LDAP client service are as follows:

- The `svcadm` command is used to enable, disable, or restart the LDAP client service.

  Tip - You can use the `-t` option to temporarily disable a service to provide protection for the service configuration. If the service is disabled with the `-t` option, the original settings are restored for the service after a reboot. If the service is disabled without `-t`, the service remains disabled after reboot.

- The Fault Management Resource Identifier (FMRI) for the LDAP client service is `svc:/network/ldap/client`.
- The LDAP client configuration process enables the `network/nis/domain` service to supply the domain name to be used by the `network/ldap/client` service.
- Use the `svcs` command to query the status of the LDAP client and the `ldap_cachemgr` daemon.
  - The following example shows the `svcs` command and its output.

    ```
    # svcs \*ldap\*
    STATE          STIME     FMRI
    online         15:43:46  svc:/network/ldap/client:default
    ```

  - Use the `-l` option if you want to provide the instance name in the FMRI.

    ```
    # svcs -l network/ldap/client:default
    fmri         svc:/network/ldap/client:default
    name         LDAP Name Service Client
    enabled      true
    state        online
    next_state   none
    restarter    svc:/system/svc/restarter:default
    manifest     /lib/svc/manifest/network/ldap/client.xml
    manifest     /lib/svc/manifest/network/network-location.xml
    manifest     /lib/svc/manifest/system/name-service/upgrade.xml
    manifest     /lib/svc/manifest/milestone/config.xml
    dependency   require_all/none svc:/system/filesystem/minimal (online)
    dependency   require_all/none svc:/network/initial (online)
    ```

```
dependency    optional_all/none svc:/network/location:default (online)
dependency    require_all/restart svc:/network/nis/domain (online)
dependency    optional_all/none svc:/system/name-service/upgrade (online)
dependency    optional_all/none svc:/milestone/config (online)
dependency    optional_all/none svc:/system/manifest-import (online)
dependency    require_all/none svc:/milestone/unconfig (online)
```

Configuration information specified in the LDAP client profiles is automatically imported into the SMF repository when the `svc:/network/ldap/client` service is started.

Defining LDAP Local Client Attributes

You can define the attributes of the LDAP client profile to configure the LDAP server. For more information about the LDAP client profile attributes, see Chapter 3, “Planning Requirements for LDAP Naming Services”. You use the `ldapservercfg` command to set up the client profile attributes on the server.

Use the `ldapclient` command to set up the following local client attributes:

- `adminDN` – Specifies the administrator entry's distinguished name for the admin credential. If the value of the `enableShadowUpdate` switch is `true` on the client system and `credentialLevel` has a value other than `self`, then you must specify the `adminDN` attribute.
- `adminPassword` – Specifies the administrator entry's password for the admin credential. If the value of the `enableShadowUpdate` switch is `true` on the client system and `credentialLevel` has a value other than `self`, then you must define the `adminPassword` attribute.
- `domainName` – Specifies the client's domain name, which becomes the default domain for the client system. You must specify the value of the attribute as it has no default value.
- `proxyDN` – Specifies the proxy's distinguished name. If the client system is configured with `credentialLevel` set to `proxy`, you must specify the `proxyDN`.
- `proxyPassword` – Specifies the proxy's password. If the client system is configured with `credentialLevel` set to `proxy`, you must define the `proxyPassword`.
- `certificatePath` – Specifies the directory on the local file system containing the certificate databases. You must use this attribute if a client system is configured with `authenticationMethod` or `serviceAuthenticationMethod` using TLS. The default value is `/var/ldap`.

Note - If the `BaseDN` in an SSD contains a trailing comma, it is used as a relative value of the `defaultSearchBase`. The values of the `defaultSearchBase` are appended to the `BaseDN` before a search is performed.

Initializing an LDAP Client

You can initialize the LDAP client with the `ldapclient` in one of two ways:

- Using a profile – When you use the `ldapclient` command, you must specify the server address of the profile and the domain. If you do not specify a profile, the default profile is assumed. The server provides the rest of the required information from the profile except the proxy and certificate database information.

  If a client's credential level is `proxy` or `proxy anonymous`, you must supply the proxy bind DN and password. For more information, see "Client Credential Levels" on page 17.

  To enable shadow data update, you must provide the administrator's credentials (`adminDN` and the `adminPassword`).

  Using a profile reduces the complexity of LDAP configuration, particularly in enterprise environments.
- Defining all the parameters in a single command line – If profile does not exist, you can create the profile on the client itself. With this method, the profile information is stored in cache files and is never refreshed by the server.

You can use various options with the `ldapclient` command to initialize the client depending on the type of client and the client profile:

- Initializing a client by using a profile that is configured with default values. For example:

```
# ldapclient init -a profilename=new -a domainname=west.example.com 192.0.2.1
System successfully configured
```

- Initialize a client whose profile is configured with per-user credentials and uses the `sasl/GSSAPI` authentication method.

Note - Several requirements must be fulfilled when you initialize a client that is configured with per-user credentials, such as Kerberos configuration and DNS server configuration to work with LDAP. For information about Kerberos, see *Managing Kerberos in Oracle Solaris 11.4*. For information about DNS configuration, see Chapter 3, "Managing DNS Server and Client Services" in *Working With Oracle Solaris 11.4 Directory and Naming Services: DNS and NIS*.

```
# ldapclient init -a profilename=gssapi_EXAMPLE.COM \
-a domainname=example.com 192.0.2.1
```

- Initializing a client that uses proxy credentials. For example:

```
# ldapclient init \
```

```
-a proxyDN=cn=proxyagent,ou=profile,dc=west,dc=example,dc=com \
-a domainname=west.example.com \
-a profilename=pit1 \
-a proxypassword=test1234 192.0.2.1
```

The `-a proxyDN` and `-a proxyPassword` options are required if the profile to be used is set up for `proxy`. Because the credentials are not stored in the profile saved on the server, you must supply the information when you initialize the client. This method is more secure than the older method of storing the proxy credentials on the server.

The proxy information is stored in the `svc:/network/ldap/client` service in the `config` and `cred` property groups.

- Initializing a client to enable the shadow data to be updated. For example:

```
# ldapclient init \
-a adminDN=cn=admin,ou=profile,dc=west,dc=example,dc=com \
-a adminPassword=admin-password \
-a domainName=west.example.com \
-a profileName=WestUserProfile \
-a proxyDN=cn=proxyagent,ou=profile,dc=west,dc=example,dc=com \
-a proxyPassword=proxy-password \
-a enableShadowUpdate=TRUE \
192.0.2.1
System successfully configured
```

Modifying an LDAP Client Configuration

You can use the `ldapclient` command without a profile to modify a client configuration. Because the modification affects only a limited number of client attributes, you can use the following commands to modify all the selected attributes.

- Modify an LDAP client to use simple authentication method. For example:

```
# ldapclient mod -a authenticationMethod=simple
```

- Modify a configured LDAP client to enable updating of shadow data. For example:

```
# ldapclient mod -a enableShadowUpdate=TRUE \
-a adminDN=cn=admin,ou=profile,dc=west,dc=example,dc=com \
-a adminPassword=admin-password
System successfully configured
```

Uninitializing an LDAP Client

Uninitializing an LDAP client means restoring the client name service to its status prior to the last time the `ldapclient` command was issued with the `init`, `modify`, or `manual` options. In other words, the `-uninit` option of the command cancels the last changes caused by the other options of the `ldapclient` command. For example, if the client was configured to use `profile1` and was then changed to use `profile2`, using `ldapclient uninit` would cause the client to revert to using `profile1`.

You use the `ldapclient` command to uninitialize an LDAP client.

```
# ldapclient uninit
System successfully recovered.
```

Using LDAP for Client Authentication

This section describes various configuration tasks that use LDAP authentication services.

Configuring PAM for LDAP

The `pam_ldap` module is a PAM module option for LDAP to authenticate clients and to perform account management. If you configured the client profile's authentication mode as `simple` and the credential level as `self`, you must also enable the `pam_krb` module.

For more information, see:

- `pam_ldap`(7) man page
- `pam_krb5`(7) man page
- *Managing Kerberos in Oracle Solaris 11.4*

If PAM policy is not explicitly specified in `/etc/pam.conf` or `/etc/security/policy.conf`, UNIX authentication is enabled by default. See the `policy.conf`(5) man page for information about the preferred configuration mechanisms and lookup order for PAM.

The preferred way to configure PAM to use LDAP policy is to update the `PAM_POLICY` entry in `/etc/security/policy.conf` to be the following:

```
PAM_POLICY=ldap
```

If you need to configure PAM to use UNIX authentication (the default), update the `PAM_POLICY` entry in `/etc/security/policy.conf` to be the following:

```
PAM_POLICY=unix
```

Setting Up TLS Security

If you are using Transport Layer Security (TLS), you must install the mandatory PEM certificate files before using the `ldapclient` command. These PEM certificate files are the self-signed server certificate and CA certificate files that must first be installed to validate the LDAP server, and possibly to validate client access to the server. For example, if you have the PEM CA certificate `certdb.pem`, you must ensure that `certdb.pem` is added to the certificate path and readable from the certificate path.

Starting with Oracle Solaris 11.4, the OpenLDAP LDAP library uses OpenSSL for security services. OpenSSL offers more robust certificate management than the certificate management that was used in prior Oracle Solaris releases.

You must install the necessary CA or self-signed certificate into the certificate directory prior to configuring the LDAP client. By default, the certificate directory location is `/var/ldap`. To change the location, use the `ldapclient` command to set the `certificatePath` attribute or change the location in the LDAP profile on the server. See the `ldapclient`(8) and `ldapaddent`(8) man pages for details. The `certificatePath` attribute is discussed in more detail in the `ldapclient`(8) man page.

When you upgrade a system from Oracle Solaris 11.3 or earlier to Oracle Solaris 11.4, the Mozilla certificate databases, if they exist, are automatically converted to the newer OpenSSL PEM format. The `svc:/system/name-service/upgrade:default` SMF service converts the Mozilla certificate databases to the OpenSSL PEM format and writes them to files within the certificate directory. OpenSSL hash links to those PEM files are also created. After the certificate databases are converted, they are renamed and can be deleted. If any unconverted Mozilla certificate databases remain in the certificate directory, they can be converted to PEM files by manually restarting the `name-service/upgrade:default` service. For information about restarting a service, see "Restarting a Service" in *Managing System Services in Oracle Solaris 11.4* and the `svcadm`(8) man page.

The OpenSSL library also supports the option of storing all the mandatory CA or self-signed certificates within a single PEM file, thus negating the need for the PEM hashing scheme. If you use this option, then LDAP naming services look for a `certdb.pem` file in the certificate directory by default instead of hashes. If the value of `certificatePath` points to a directory, then the LDAP client looks for PEM file hashes, and then for a `certdb.pem` file, and uses the certificate format that it discovers.

In order to allow OpenLDAP commands such as `ldapsearch`, `ldapadd`, and `ldapmodify` to work with the TLS configuration, the location of the PEM certificate files must be specified

in `/etc/openldap/ldap.conf`. See the `TLS_CACERTDIR` option or, if using `certdb.pem`, the `TLS_CACERT` option in the `ldap.conf`(5oldap) man page.

Note - The PEM certificate files must be readable by everyone. Do not encrypt or remove read permissions on these files. Otherwise, commands such as `ldaplist` will fail.

For information about how to create and manage PEM formatted certificates, see Directory Server Security.

How to Set Up TLS Security

1. **Create the necessary PEM certificate file. For example, `certdb.pem`.**
2. **Copy that file to the default location.**

   For example:

   ```
   # cp certdb.pem /var/ldap
   ```

3. **Ensure that everyone can read the PEM certificate file.**

   ```
   # chmod 444 /var/ldap/certdb.pem
   ```

Note - More than one certificate file might reside in the certificate path. Additionally, any given PEM certificate file might contain multiple PEM format certificates that are concatenated together. Refer to your server documentation for further details. The certificate files must be stored on a local file system if you are using them for an LDAP naming service client.

♦ ♦ ♦ CHAPTER 6

Troubleshooting LDAP Configurations

This chapter describes common LDAP configuration problems and suggests solutions for resolving them. It covers the following topics:

Displaying the LDAP Naming Service Information

You can use the `ldaplist` utility to display information about LDAP naming service. This LDAP utility lists the naming information from the LDAP servers in LDIF format, which can be useful for troubleshooting. For more information, see the `ldaplist(1)` man page.

Displaying All LDAP Containers

The `ldaplist` command displays its output with a blank line separating records, which is helpful for big multiline records.

The output of `ldaplist` depends upon the client configuration. For example, if the value of `ns_ldap_search` is `sub` rather than `one`, `ldaplist` lists all the entries under the current search `baseDN`.

The following example shows sample `ldaplist` output.

```
# ldaplist
dn: ou=people,dc=west,dc=example,dc=com

dn: ou=group,dc=west,dc=example,dc=com
```

```
dn: ou=rpc,dc=west,dc=example,dc=com

dn: ou=protocols,dc=west,dc=example,dc=com

dn: ou=networks,dc=west,dc=example,dc=com

dn: ou=netgroup,dc=west,dc=example,dc=com

dn: ou=aliases,dc=west,dc=example,dc=com

dn: ou=hosts,dc=west,dc=example,dc=com

dn: ou=services,dc=west,dc=example,dc=com

dn: ou=ethers,dc=west,dc=example,dc=com

dn: ou=profile,dc=west,dc=example,dc=com

dn: automountmap=auto_home,dc=west,dc=example,dc=com

dn: automountmap=auto_direct,dc=west,dc=example,dc=com

dn: automountmap=auto_master,dc=west,dc=example,dc=com

dn: automountmap=auto_shared,dc=west,dc=example,dc=com
```

Displaying All User Entry Attributes

To list specific information such as a user's `passwd` entry, use the `getent` command. For example:

```
# getent passwd user1
user1::30641:10:Joe Q. User:/home/user1:/bin/csh
```

You also use the `getent` command to perform lookups on databases that are listed in the automount table, for example, `getent automount/`*map* `[`*key*`]`. In the following example, `auto_home` is the name of the automount map and `user1` is the search key. If you do not specify any search key, then the entire content of the specified automount map is listed.

```
# getent automount/auto_home user1
user1 server-name:/home/user1
```

To list all attributes, use `ldaplist` with the `-l` option.

```
# ldaplist -l passwd user1
```

```
dn: uid=user1,ou=People,dc=west,dc=example,dc=com
uid: user1
cn: user1
uidNumber: 30641
gidNumber: 10
gecos: Joe Q. User
homeDirectory: /home/user1
loginShell: /bin/csh
objectClass: top
objectClass: shadowAccount
objectClass: account
objectClass: posixAccount
shadowLastChange: 6445
```

Monitoring LDAP Client Status

This section describes commands that are used to determine the state of the LDAP client environment. For additional information about the command options, see the related man pages.

For information about Service Management Facility (SMF), refer to *Managing System Services in Oracle Solaris 11.4*. Also refer to the `svcadm`(8) and `svcs`(1) man pages for more details.

Verifying the `ldap_cachemgr` Daemon Status

The `ldap_cachemgr` daemon must be online and functioning correctly at all times for the system to work. When you set up and start the LDAP client service, `svc:/network/ldap/client`, the client SMF method automatically starts the `ldap_cachemgr` daemon.

Viewing the State of the Service

To view the state of the service, use the `svcs` command.

```
# svcs \*ldap\*
STATE          STIME    FMRI
disabled       Aug_24   svc:/network/ldap/client:default
```

Viewing the Information About the Service

To view all information about the service, use the `-l` option.

```
# svcs -l network/ldap/client:default
fmri svc:/network/ldap/client:default
name LDAP Name Service Client
enabled false
state disabled
next_state none
state_time Thu Oct 20 23:04:11 2011
logfile /var/svc/log/network-ldap-client:default.log
restarter svc:/system/svc/restarter:default
contract_id
manifest /lib/svc/manifest/network/ldap/client.xml
manifest /lib/svc/manifest/milestone/config.xml
manifest /lib/svc/manifest/network/network-location.xml
manifest /lib/svc/manifest/system/name-service/upgrade.xml
dependency optional_all/none svc:/milestone/config (online)
dependency optional_all/none svc:/network/location:default (online)
dependency require_all/none svc:/system/filesystem/minimal (online)
dependency require_all/none svc:/network/initial (online)
dependency require_all/restart svc:/network/nis/domain (online)
dependency optional_all/none svc:/system/manifest-import (online)
dependency require_all/none svc:/milestone/unconfig (online)
dependency optional_all/none svc:/system/name-service/upgrade (online)
```

Viewing Detailed Information About the State of the Service

To view more extensive status information, which is useful for diagnosing a problem, pass the `-g` option to `ldap_cachemgr`.

```
# /usr/lib/ldap/ldap_cachemgr -g

cachemgr configuration:
server debug level          0
server log file "/var/ldap/cachemgr.log"
number of calls to ldapcachemgr       2144
SMF service state: online

cachemgr cache data statistics:
Configuration refresh information:
  Previous refresh time: 2021/09/16 11:32:05
  Next refresh time:     2021/09/28 10:17:43
Server information:
  Previous refresh time: 2021/09/28 09:47:47
  Next refresh time:     2021/09/28 10:02:47
  server: aldap.example.com, ODSEE/RAWSSL, status: UP
    vendor: Oracle Corporation, version: Sun-Directory-Server/11.1.1.7.181016
    last seen: 2021/09/28 09:47:47, round trip: 604.094 ms
  server: bldap.example.com, ODSEE/RAWSSL, status: UP
```

```
    vendor: Oracle Corporation, version: Sun-Directory-Server/11.1.1.7.181016
    last seen: 2021/09/28 09:47:47, round trip: 630.718 ms
  server: cldap.example.com, ODPS/RAWSSL, status: ERROR
    error message: Can't connect to the LDAP server
Cache data information:
  Maximum cache entries:          256
  Number of cache entries:          0
```

If the `ldap_cachemgr` daemon is disabled, use the `svcadm enable network/ldap/client` command to enable the daemon.

For more information about the `ldap_cachemgr` daemon, see the `ldap_cachemgr(8)` man page.

Checking the Client Profile Information

Become a superuser or assume an equivalent role, and use the `ldapclient` command with the `list` option to view the current profile information. In addition to the `ldapclient list` command, you can also use the `svccfg` or `svcprop` commands to obtain current profile information.

```
# ldapclient list
NS_LDAP_FILE_VERSION= 2.0
NS_LDAP_BINDDN= cn=proxyagent,ou=profile,dc=west,dc=example,dc=com
NS_LDAP_BINDPASSWD= {NS1}4a3788e8c053424f
NS_LDAP_SERVERS= 192.0.2.1, 192.0.2.10
NS_LDAP_SEARCH_BASEDN= dc=west,dc=example,dc=com
NS_LDAP_AUTH= simple
NS_LDAP_SEARCH_REF= TRUE
NS_LDAP_SEARCH_SCOPE= one
NS_LDAP_SEARCH_TIME= 30
NS_LDAP_SERVER_PREF= 192.0.2.1
NS_LDAP_PROFILE= pit1
NS_LDAP_CREDENTIAL_LEVEL= proxy
NS_LDAP_SERVICE_SEARCH_DESC= passwd:ou=people,?sub
NS_LDAP_SERVICE_SEARCH_DESC= group:ou=group,dc=west,dc=example,dc=com?one
NS_LDAP_BIND_TIME= 5
```

Verifying Basic Client-Server Communication

Use the `ldaplist` command to verify whether communication exists between the LDAP client and the LDAP server.

- To display all the containers of the DIT on the server, use the `ldaplist` command without options.

- To display the contents of the specific database, use the `ldaplist` *database* command, for example, `ldaplist passwd` *username* or `ldaplist host` *hostname*.

Checking LDAP Server Data From a Non-Client Machine

To check for information on a system that has no existing LDAP client, use the `ldapsearch` command. The information that is displayed depends on the filter you use for searching. The following example lists all of the containers in the DIT:

```
# ldapsearch -H ldapuri -b "dc=west,dc=example,dc=com" -s one "objectclass=*"
```

For a list of options and filters that you can use with the `ldapsearch` command, see the `ldapsearch`(1oldap) man page.

`name-service/cache` Must be Enabled for Oracle Solaris 11.4

To ensure that the `ldapclient` services function correctly, ensure that you enable the `svc:/system/name-service/cache` service, that the service is online, and that the `nscd` daemon is running.

```
# svcs name-service/cache
STATE          STIME           FMRI
online         10:58:21        svc:/system/name-service/cache:default
```

`ldaplist` Fails and is Restricted to Privileged Users

```
$ ldaplist 2>&1 | fmt -s
ldaplist: libsldap.so.1 internal error Partial results due to timeout:
LDAP ERROR (50): Error occurred while receiving results. Insufficient
access. and libldap returned: (50) VLV Control using
ldaps://ldaps.example.com:636
```

The access to LDAP credentials is restricted to users and programs that have the `solaris.smf.value.name-service.ldap.client` authorization. Without this authorization, a user can only see LDAP server lookups that do not require credentials. This situation also affects any tool that uses the LDAP naming service configuration, such as `ldaplist`.

Note that any user can use the `getent` command to retrieve LDAP information. Such lookups occur inside `nscd`, which has access to the credentials.

LDAP Configuration Problems and Solutions

This section describes possible LDAP configuration problems and solutions.

Unresolved Host Name

The LDAP client software returns fully qualified host names for host lookups, such as host names returned by `gethostbyname()` and `getaddrinfo()`.

- If the name stored is qualified, that is, if it contains at least one dot, the client returns the name as is. For example, if the name stored is `hostB.eng`, the returned name is `hostB.eng`.
- If the name stored in the LDAP directory is not qualified, that is, it does not contain a dot, the client appends the domain part to the host name as set in the `nisDomain` attribute set at the root DN in the object class of `nisDomainObject`. For example, if the name stored is `hostA`, the returned name is `hostA.`*domain-name*.

Unable to Reach Systems in the LDAP Domain Remotely

If the DNS domain name is different from the LDAP domain name, then the LDAP naming service cannot be used to serve host names unless the host names are stored as fully qualified names.

Login Does Not Work

LDAP clients use the PAM modules for user authentication during login. When using the standard UNIX PAM module, the password is read from the server and checked on the client side. This process can fail for any of the following reasons:

- `ldap` is not associated with the `passwd` database in the name service switch.
- The proxy agent cannot read the user's `userPassword` attribute on the server list. You must enable at least the proxy agent to read the password because the proxy agent returns the password to the client for comparison. `pam_ldap` does not require read access to the password.
- The proxy agent does not have the correct password.
- The entry does not have the `shadowAccount` object class.
- No password is defined for the user.

  Make sure the user's `userPassword` attribute exists.
- LDAP Server TLS Connection issues.

  Ensure that either a local `/etc/hosts` or DNS entry (if `nsswitch.conf` is configured for DNS) exists for the LDAP server and that the X.509 Certificate CN attribute of the Subject DN or `subjectAltName` extension in the X.509 certificate matches that `/etc/hosts` or DNS entry for the configured LDAP server.

  To determine what certificate the server has configured, you can attempt connection by using the `openssl` command:

  ```
  $ openssl s_client -verify 2 -verify_hostname ldapservername \
  -verify 1 -connect ldapservername:636 </dev/null
  ```
- No LDAP servers are reachable.

  Check the status of the servers.

  ```
  # /usr/lib/ldap/ldap_cachemgr -g
  ```
- `pam.conf` is configured incorrectly.
- The user is not defined in the LDAP namespace.
- `NS_LDAP_CREDENTIAL_LEVEL` is set to `anonymous` for the `pam_unix_*` modules, and `userPassword` is not available to anonymous users.
- The password is not stored in `crypt` format.
- If `pam_ldap` is configured to support account management, a login failure could be the result of one of the following causes:
  - The user's password has expired.
  - The user's account is locked out due to too many failed login attempts.
  - The user's account has been deactivated by the administrator.
  - The user tried to log in using a non-password based program, such as `ssh` or `sftp`.
- If you are using per-user authentication and `sasl/GSSAPI`, then some component of Kerberos or the `pam_krb5` configuration might be set up incorrectly. For more information about resolving Kerberos issues, see the *Managing Kerberos in Oracle Solaris 11.4*.

Lookup Too Slow

The LDAP database relies on indexes to improve search performance. You must index a common set of attributes that is included in the LDAP documentation provided by Oracle and other vendors. You can also add your own indexes to improve performance at your site.

`ldapclient` Command Cannot Bind to a Server

The possible reasons for failure of the `ldapclient` command to initialize the client when using the `init` option with the `profileName` attribute are as follows:

- The incorrect domain name was specified on the command line.
- The `nisDomain` attribute is not set in the DIT to represent the entry point for the specified client domain.
- Access control information is not set up properly on the server, thus disabling anonymous search in the LDAP database.
- An incorrect server address was passed to the `ldapclient` command. Use the `ldapsearch` command to verify the server address.
- An incorrect profile name was passed to the `ldapclient` command. Use the `ldapsearch` command to verify the profile name in the DIT.

As a troubleshooting aid, use `snoop` on the client's network interface to see what sort of traffic is going out, and determine the server to which it is talking.

Using the `ldap_cachemgr` Daemon for Debugging

Running the `ldap_cachemgr` daemon with the `-g` option to view the current client configuration and statistics can be useful for debugging. This command displays current configuration and statistics to standard output, including the status of all LDAP servers. Note that you do *not* need to become superuser to execute this command. See "Viewing Detailed Information About the State of the Service" on page 82.

As the `root` user, you can use the `-l` *log-file* option to specify an alternate log file for `ldap_cachemgr` instead of the default `/var/ldap/cachemgr.log` log file. In addition, you can use the `-d` *value* option to enable or disable debugging in the running `ldap_cachemgr` process. A value of `0` disables debugging and the values `1` through `6` show increasingly detailed information.

`ldapclient` Command Hangs During Setup

If the `ldapclient` command hangs, press Ctrl-C to exit after restoring the previous environment. In such an event, check with the server administrator to ensure that the server is running.

Also check the server list attributes either in the profile or from the command line and make sure that the server information is correct.

Resolving Per-User Credentials Issues

Using per-user credentials requires configuration such as a Kerberos setup. Refer to the following issues when configuring per-user profiles.

`syslog` File Indicates `82 Local Error`

The `syslog` file might contain the following error message:

```
libsldap: Status: 7 Mesg: openConnection: GSSAPI bind failed -82 Local error
```

Kerberos might not be initialized or its ticket is expired. Use the `klist` command to browse. Use either the `kinit -p` command or `kinit -R` command to reinitialize Kerberos.

Kerberos Not Initializing Automatically

To enable the `kinit` command to run automatically whenever you log in, add `pam_krb5.so.1` to the `/etc/pam.conf` file. For example:

```
login      auth optional pam_krb5.so.1
rlogin     auth optional pam_krb5.so.1
other      auth optional pam_krb5.so.1
```

`syslog` File Indicates Invalid Credentials

The `syslog` file might contain `Invalid credential` after you use the `kinit` command. This problem might occur due to one of the following reasons:

- The `root` host entry or the user entry is not in the LDAP directory.
- Mapping rules are incorrect.

The `ldapclient init` Command Fails in the Switch Check

You can use the `ldapclient init` command to check the LDAP profile for the presence of the `self/sasl/GSSAPI` configuration. If the switch check fails, the error lies in DNS not being used as the search criteria for the host database. You can resolve this issue as follows:

- Use the following commands to check the status of the DNS service and to enable it.

```
# svcs -l dns/client
# svcadm enable dns/client
```

- If the failure is in the bind operation of `sasl/GSSAPI`, check the `syslog` file to determine the problem.

♦ ♦ ♦ CHAPTER 7

LDAP Schemas

This chapter describes LDAP schemas and the different types of schemas supported by Oracle Solaris. It covers the following topics:

IETF Schemas for LDAP

Schemas are definitions that describe what types of information can be stored as entries in a server's directory.

For a directory server to support LDAP naming clients, the schemas defined in this section must be configured in the server unless schema is mapped using the schema mapping feature of the clients.

IETF defines several LDAP schemas: the RFC 2307 Network Information Service (NIS) schema and RFC 2307bis, and a Configuration Profile Schema for LDAP-Based Agents (RFC 4876), and the LDAP Schema for Printer Services. To support NIS services, you must add the definition of these schemas to the directory server. You can access the RFCs on the IETF web site at `https://www.ietf.org`.

Note - Internet drafts, such as RFC 2307bis, are draft documents valid for a maximum of six months and might be updated, or rendered obsolete, by other documents at any time.

RFC 2307bis Network Information Service Schema

You must configure the LDAP servers to support the revised RFC 2307bis schema.

The `nisSchema` OID is `1.3.6.1.1`. The RFC 2307bis attributes are as follows:

```
( nisSchema.1.0 NAME 'uidNumber'
DESC 'An integer uniquely identifying a user in an
  administrative domain'
EQUALITY integerMatch SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.1 NAME 'gidNumber'
DESC 'An integer uniquely identifying a group in an
  administrative domain'
EQUALITY integerMatch SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.2 NAME 'gecos'
DESC 'The GECOS field; the common name'
EQUALITY caseIgnoreIA5Match
SUBSTRINGS caseIgnoreIA5SubstringsMatch
SYNTAX 'IA5String' SINGLE-VALUE )

( nisSchema.1.3 NAME 'homeDirectory'
DESC 'The absolute path to the home directory'
EQUALITY caseExactIA5Match
SYNTAX 'IA5String' SINGLE-VALUE )

( nisSchema.1.4 NAME 'loginShell'
DESC 'The path to the login shell'
EQUALITY caseExactIA5Match
SYNTAX 'IA5String' SINGLE-VALUE )

( nisSchema.1.5 NAME 'shadowLastChange'
EQUALITY integerMatch
SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.6 NAME 'shadowMin'
EQUALITY integerMatch
SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.7 NAME 'shadowMax'
EQUALITY integerMatch
SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.8 NAME 'shadowWarning'
EQUALITY integerMatch
SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.9 NAME 'shadowInactive'
EQUALITY integerMatch
SYNTAX 'INTEGER' SINGLE-VALUE )
```

```
( nisSchema.1.10 NAME 'shadowExpire'
EQUALITY integerMatch
SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.11 NAME 'shadowFlag'
EQUALITY integerMatch
SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.12 NAME 'memberUid'
EQUALITY caseExactIA5Match
SUBSTRINGS caseExactIA5SubstringsMatch
SYNTAX 'IA5String' )

( nisSchema.1.13 NAME 'memberNisNetgroup'
EQUALITY caseExactIA5Match
SUBSTRINGS caseExactIA5SubstringsMatch
SYNTAX 'IA5String' )

( nisSchema.1.14 NAME 'nisNetgroupTriple'
DESC 'Netgroup triple'
SYNTAX 'nisNetgroupTripleSyntax' )

( nisSchema.1.15 NAME 'ipServicePort'
EQUALITY integerMatch
SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.16 NAME 'ipServiceProtocol'
SUP name )

( nisSchema.1.17 NAME 'ipProtocolNumber'
EQUALITY integerMatch
SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.18 NAME 'oncRpcNumber'
EQUALITY integerMatch
SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.19 NAME 'ipHostNumber'
DESC 'IP address as a dotted decimal, eg. 192.0.2.1
      omitting leading zeros'
SUP name )

( nisSchema.1.20 NAME 'ipNetworkNumber'
DESC 'IP network as a dotted decimal, eg. 192.0.2.1,
      omitting leading zeros'
SUP name SINGLE-VALUE )

( nisSchema.1.21 NAME 'ipNetmaskNumber'
```

```
DESC 'IP netmask as a dotted decimal, eg. 255.255.255.0,
       omitting leading zeros'
EQUALITY caseIgnoreIA5Match
SYNTAX 'IA5String{128}' SINGLE-VALUE )

( nisSchema.1.22 NAME 'macAddress'
DESC 'MAC address in maximal, colon separated hex
      notation, eg. 00:00:5E:00:53:00'
EQUALITY caseIgnoreIA5Match
SYNTAX 'IA5String{128}' )

( nisSchema.1.23 NAME 'bootParameter'
DESC 'rpc.bootparamd parameter'
SYNTAX 'bootParameterSyntax' )

( nisSchema.1.24 NAME 'bootFile'
DESC 'Boot image name'
EQUALITY caseExactIA5Match
SYNTAX 'IA5String' )

( nisSchema.1.26 NAME 'nisMapName'
SUP name )

( nisSchema.1.27 NAME 'nisMapEntry'
EQUALITY caseExactIA5Match
SUBSTRINGS caseExactIA5SubstringsMatch
SYNTAX 'IA5String{1024}' SINGLE-VALUE )

( nisSchema.1.28 NAME 'nisPublicKey'
DESC 'NIS public key'
SYNTAX 'nisPublicKeySyntax' )

( nisSchema.1.29 NAME 'nisSecretKey'
DESC 'NIS secret key'
SYNTAX 'nisSecretKeySyntax' )

( nisSchema.1.30 NAME 'nisDomain'
DESC 'NIS domain'
SYNTAX 'IA5String' )

( nisSchema.1.31 NAME 'automountMapName'
DESC 'automount Map Name'
EQUALITY caseExactIA5Match
SUBSTR caseExactIA5SubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )

( nisSchema.1.32 NAME 'automountKey'
DESC 'Automount Key value'
```

```
EQUALITY caseExactIA5Match
SUBSTR caseExactIA5SubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )

( nisSchema.1.33 NAME 'automountInformation'
DESC 'Automount information'
EQUALITY caseExactIA5Match
SUBSTR caseExactIA5SubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

The RFC 2307 `objectClasses` are as follows:

```
( nisSchema.2.0 NAME 'posixAccount' SUP top AUXILIARY
  DESC 'Abstraction of an account with POSIX attributes'
  MUST ( cn $ uid $ uidNumber $ gidNumber $ homeDirectory )
  MAY ( userPassword $ loginShell $ gecos $ description ) )

( nisSchema.2.1 NAME 'shadowAccount' SUP top AUXILIARY
  DESC 'Additional attributes for shadow passwords'
  MUST uid
  MAY ( userPassword $ shadowLastChange $ shadowMin
        shadowMax $ shadowWarning $ shadowInactive $
        shadowExpire $ shadowFlag $ description ) )

( nisSchema.2.2 NAME 'posixGroup' SUP top STRUCTURAL
  DESC 'Abstraction of a group of accounts'
  MUST ( cn $ gidNumber )
  MAY ( userPassword $ memberUid $ description ) )

( nisSchema.2.3 NAME 'ipService' SUP top STRUCTURAL
  DESC 'Abstraction an Internet Protocol service.
        Maps an IP port and protocol (such as tcp or udp)
        to one or more names; the distinguished value of
        the cn attribute denotes the service's canonical
        name'
  MUST ( cn $ ipServicePort $ ipServiceProtocol )
  MAY ( description ) )

( nisSchema.2.4 NAME 'ipProtocol' SUP top STRUCTURAL
  DESC 'Abstraction of an IP protocol. Maps a protocol number
        to one or more names. The distinguished value of the cn
        attribute denotes the protocol's canonical name'
  MUST ( cn $ ipProtocolNumber )
  MAY  description )

( nisSchema.2.5 NAME 'oncRpc' SUP top STRUCTURAL
  DESC 'Abstraction of an Open Network Computing (ONC)
        [RFC1057] Remote Procedure Call (RPC) binding.
        This class maps an ONC RPC number to a name.
```

```
        The distinguished value of the cn attribute denotes
        the RPC service's canonical name'
  MUST ( cn $ oncRpcNumber $ description )
  MAY  description )

( nisSchema.2.6 NAME 'ipHost' SUP top AUXILIARY
  DESC 'Abstraction of a host, an IP device. The distinguished
        value of the cn attribute denotes the host's canonical
        name. Device SHOULD be used as a structural class'
  MUST ( cn $ ipHostNumber )
  MAY ( l $ description $ manager $ userPassword ) )

( nisSchema.2.7 NAME 'ipNetwork' SUP top STRUCTURAL
  DESC 'Abstraction of a network. The distinguished value of
        the cn attribute denotes the network's canonical name'
  MUST ipNetworkNumber
  MAY ( cn $ ipNetmaskNumber $ l $ description $ manager ) )

( nisSchema.2.8 NAME 'nisNetgroup' SUP top STRUCTURAL
  DESC 'Abstraction of a netgroup. May refer to other netgroups'
  MUST cn
  MAY ( nisNetgroupTriple $ memberNisNetgroup $ description ) )

( nisSchema.2.9 NAME 'nisMap' SUP top STRUCTURAL
  DESC 'A generic abstraction of a NIS map'
  MUST nisMapName
  MAY description )

( nisSchema.2.10 NAME 'nisObject' SUP top STRUCTURAL
  DESC 'An entry in a NIS map'
  MUST ( cn $ nisMapEntry $ nisMapName )
  MAY description )

( nisSchema.2.11 NAME 'ieee802Device' SUP top AUXILIARY
  DESC 'A device with a MAC address; device SHOULD be
        used as a structural class'
  MAY macAddress )

( nisSchema.2.12 NAME 'bootableDevice' SUP top AUXILIARY
  DESC 'A device with boot parameters; device SHOULD be
  used as a structural class'
  MAY ( bootFile $ bootParameter ) )

( nisSchema.2.14 NAME 'nisKeyObject' SUP top AUXILIARY
  DESC 'An object with a public and secret key'
  MUST ( cn $ nisPublicKey $ nisSecretKey )
  MAY ( uidNumber $ description ) )
```

```
( nisSchema.2.15 NAME 'nisDomainObject' SUP top AUXILIARY
  DESC 'Associates a NIS domain with a naming context'
  MUST nisDomain )

( nisSchema.2.16 NAME 'automountMap' SUP top STRUCTURAL
  MUST ( automountMapName )
  MAY description )

( nisSchema.2.17 NAME 'automount' SUP top STRUCTURAL
  DESC 'Automount information'
  MUST ( automountKey $ automountInformation )
  MAY description )

( nisSchema.2.18 NAME 'groupOfMembers' SUP top STRUCTURAL
  DESC 'A group with members (DNs)'
  MUST cn
  MAY ( businessCategory $ seeAlso $ owner $ ou $ o $
        description $ member ) )
```

Mail Alias Schema

Mail alias information uses the schema defined by the Internet draft.

The original LDAP mail groups schema contains a large number of attributes and object classes. LDAP clients use only two attributes and a single object class. The mail alias attributes are as follows:

```
( 0.9.2342.19200300.100.1.3
  NAME 'mail'
  DESC 'RFC822 email address for this person'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 'IA5String(256)'
  SINGLE-VALUE )

( 2.16.840.1.113730.3.1.30
  NAME 'mgrpRFC822MailMember'
  DESC 'RFC822 mail address of email only member of group'
  EQUALITY CaseIgnoreIA5Match
  SYNTAX 'IA5String(256)' )
```

The schema for the `mailGroup` object class is as follows:

```
( 2.16.840.1.113730.3.2.4
  NAME 'mailGroup'
  SUP top
  STRUCTURAL
  MUST mail
```

```
MAY ( cn $ mailAlternateAddress $ mailHost $ mailRequireAuth $
 mgrpAddHeader $ mgrpAllowedBroadcaster $ mgrpAllowedDomain $
 mgrpApprovePassword $ mgrpBroadcasterModeration $ mgrpDeliverTo $
 mgrpErrorsTo $ mgrpModerator $ mgrpMsgMaxSize $
 mgrpMsgRejectAction $ mgrpMsgRejectText $ mgrpNoMatchAddrs $
 mgrpRemoveHeader $ mgrpRFC822MailMember ))
```

Directory User Agent Profile (DUAProfile) Schema

The DUAConfSchemaOID is 1.3.6.1.4.1.11.1.3.1.

```
    ( DESC 'Default LDAP server host address used by a DUA'
            EQUALITY caseIgnoreMatch
            SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
            SINGLE-VALUE )

          ( DUAConfSchemaOID.1.0 NAME 'defaultServerList'
            DESC 'Default LDAP server host address used by a DUAList'
            SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
            SINGLE-VALUE )

          ( DUAConfSchemaOID.1.1 NAME 'defaultSearchBase'
            DESC 'Default LDAP base DN used by a DUA'
            EQUALITY distinguishedNameMatch
            SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
            SINGLE-VALUE )

          ( DUAConfSchemaOID.1.2 NAME 'preferredServerList'
            DESC 'Preferred LDAP server host addresses to be used by a
            DUA'
            EQUALITY caseIgnoreMatch
            SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
            SINGLE-VALUE )

          ( DUAConfSchemaOID.1.3 NAME 'searchTimeLimit'
            DESC 'Maximum time in seconds a DUA should allow for a
            search to complete'
            EQUALITY integerMatch
            SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
            SINGLE-VALUE )

          ( DUAConfSchemaOID.1.4 NAME 'bindTimeLimit'
            DESC 'Maximum time in seconds a DUA should allow for the
            bind operation to complete'
            EQUALITY integerMatch
            SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
```

```
  SINGLE-VALUE )

( DUAConfSchemaOID.1.5 NAME 'followReferrals'
  DESC 'Tells DUA if it should follow referrals
  returned by a DSA search result'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
  SINGLE-VALUE )

( DUAConfSchemaOID.1.6 NAME 'authenticationMethod'
  DESC 'A keystring which identifies the type of
  authentication method used to contact the DSA'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE )

( DUAConfSchemaOID.1.7 NAME 'profileTTL'
  DESC 'Time to live before a client DUA
  should re-read this configuration profile'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
  SINGLE-VALUE )

( DUAConfSchemaOID.1.9 NAME 'attributeMap'
  DESC 'Attribute mappings used by a DUA'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

( DUAConfSchemaOID.1.10 NAME 'credentialLevel'
  DESC 'Identifies type of credentials a DUA should
  use when binding to the LDAP server'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE )

( DUAConfSchemaOID.1.11 NAME 'objectclassMap'
  DESC 'Objectclass mappings used by a DUA'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

( DUAConfSchemaOID.1.12 NAME 'defaultSearchScope'
  DESC 'Default search scope used by a DUA'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE )

( DUAConfSchemaOID.1.13 NAME 'serviceCredentialLevel'
  DESC 'Identifies type of credentials a DUA
  should use when binding to the LDAP server for a
  specific service'
```

```
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

      ( DUAConfSchemaOID.1.14 NAME 'serviceSearchDescriptor'
        DESC 'LDAP search descriptor list used by Naming-DUA'
        EQUALITY caseIgnoreMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

      ( DUAConfSchemaOID.1.15 NAME 'serviceAuthenticationMethod'
        DESC 'Authentication Method used by a service of the DUA'
        EQUALITY caseIgnoreMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

( DUAConfSchemaOID.2.4 NAME 'DUAConfigProfile'
SUP top STRUCTURAL
DESC 'Abstraction of a base configuration for a DUA'
MUST ( cn )
MAY ( defaultServerList $ preferredServerList $
            defaultSearchBase $ defaultSearchScope $
            searchTimeLimit $ bindTimeLimit $
            credentialLevel $ authenticationMethod $
            followReferrals $ serviceSearchDescriptor $
            serviceCredentialLevel $ serviceAuthenticationMethod $
            objectclassMap $ attributeMap $
            profileTTL ) )
```

Oracle Solaris Schemas

Oracle Solaris requires the following schemas:

- Projects schema
- Role-based access control and execution profile schemas
- Printer schemas

Projects Schema

The `/etc/project` file is a local source of attributes associated with projects. For more information, see the `user_attr(5)` man page.

The project attributes are as follows:

```
( 1.3.6.1.4.1.42.2.27.5.1.1 NAME 'SolarisProjectID'
  DESC 'Unique ID for a Solaris Project entry'
  EQUALITY integerMatch
```

```
  SYNTAX INTEGER SINGLE )

( 1.3.6.1.4.1.42.2.27.5.1.2 NAME 'SolarisProjectName'
  DESC 'Name of a Solaris Project entry'
  EQUALITY caseExactIA5Match
  SYNTAX IA5String SINGLE )

( 1.3.6.1.4.1.42.2.27.5.1.3 NAME 'SolarisProjectAttr'
  DESC 'Attributes of a Solaris Project entry'
  EQUALITY caseExactIA5Match
  SYNTAX IA5String )

( 1.3.6.1.4.1.42.2.27.5.1.30 NAME 'memberGid'
  DESC 'Posix Group Name'
  EQUALITY caseExactIA5Match
  SYNTAX 'IA5String' )
```

The Project `objectClass` is as follows:

```
( 1.3.6.1.4.1.42.2.27.5.2.1 NAME 'SolarisProject'
  SUP top STRUCTURAL
  MUST ( SolarisProjectID $ SolarisProjectName )
  MAY ( memberUid $ memberGid $ description $ SolarisProjectAttr ) )
```

Role-Based Access Control and Execution Profile Schema

The `/etc/user_attr` file is the local source of extended attributes associated with users and roles. For more information, see the `user_attr`(5) man page.

You can add the `SolarisQualifiedUserAttr` object class to the existing Oracle Solaris RBAC schema. You can specify multiple values to the attributes of this class and thus enhance the current `SolarisUserQualifier` class. If you already have an existing LDAP configuration prior to the availability of the `SolarisQualifiedUserAttr` class, you can use the `ldapadd` command to add the class to the configuration.

The role-based access control attributes are as follows:

```
( 1.3.6.1.4.1.42.2.27.5.1.4 NAME 'SolarisAttrKeyValue'
  DESC 'Semi-colon separated key=value pairs of attributes'
  EQUALITY caseIgnoreIA5Match
  SUBSTRINGS caseIgnoreIA5Match
  SYNTAX 'IA5String' SINGLE-VALUE )

( 1.3.6.1.4.1.42.2.27.5.1.7 NAME 'SolarisAttrShortDesc'
  DESC 'Short description about an entry, used by GUIs'
```

```
  EQUALITY caseIgnoreIA5Match
  SYNTAX 'IA5String' SINGLE-VALUE )

( 1.3.6.1.4.1.42.2.27.5.1.8 NAME 'SolarisAttrLongDesc'
  DESC 'Detail description about an entry'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 'IA5String' SINGLE-VALUE )

( 1.3.6.1.4.1.42.2.27.5.1.9 NAME 'SolarisKernelSecurityPolicy'
  DESC 'Solaris  kernel security policy'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 'IA5String' SINGLE-VALUE )

( 1.3.6.1.4.1.42.2.27.5.1.10 NAME 'SolarisProfileType'
  DESC 'Type of object defined in profile'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 'IA5String' SINGLE-VALUE )

( 1.3.6.1.4.1.42.2.27.5.1.11 NAME 'SolarisProfileId'
  DESC 'Identifier of object defined in profile'
  EQUALITY caseExactIA5Match
  SYNTAX 'IA5String' SINGLE-VALUE )

( 1.3.6.1.4.1.42.2.27.5.1.12 NAME 'SolarisUserQualifier'
  DESC 'Per-user login attributes'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 'IA5String' SINGLE-VALUE )

( 1.3.6.1.4.1.42.2.27.5.1.13 NAME 'SolarisReserved1'
  DESC 'Reserved for future use'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 'IA5String' SINGLE-VALUE )

( 1.3.6.1.4.1.42.2.27.5.1.14 NAME 'SolarisReserved2'
  DESC 'Reserved for future use'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 'IA5String' SINGLE-VALUE )

( 2.16.840.1.113894.1009.2.100.1.1 NAME 'SolarisUserAttrEntry'
  DESC 'user_attr file format without username'
  EQUALITY caseExactIA5Match
  SYNTAX  'IA5String'  )


( 2.16.840.1.113894.1009.2.100.1.2 NAME 'SolarisUserType'
  DESC 'specifies whether a normal user or a role'
  EQUALITY caseExactIA5Match
  SYNTAX  'IA5String'  SINGLE-VALUE )
```

The role based access control `objectClasses` are as follows:

```
( 1.3.6.1.4.1.42.2.27.5.2.3 NAME 'SolarisUserAttr' SUP top AUXILIARY
  DESC 'User attributes'
  MAY ( SolarisUserQualifier $ SolarisAttrReserved1 $ \
        SolarisAttrReserved2 $ SolarisAttrKeyValue ) )

( 1.3.6.1.4.1.42.2.27.5.2.4 NAME 'SolarisAuthAttr' SUP top STRUCTURAL
  DESC 'Authorizations data'
  MUST cn
  MAY ( SolarisAttrReserved1 $ SolarisAttrReserved2 $ \
        SolarisAttrShortDesc $ SolarisAttrLongDesc $ \
        SolarisAttrKeyValue ) )

( 1.3.6.1.4.1.42.2.27.5.2.5 NAME 'SolarisProfAttr' SUP top STRUCTURAL
  DESC 'Profiles data'
  MUST cn
  MAY ( SolarisAttrReserved1 $ SolarisAttrReserved2 $ \
        SolarisAttrLongDesc $ SolarisAttrKeyValue ) )

( 1.3.6.1.4.1.42.2.27.5.2.6 NAME 'SolarisExecAttr' SUP top AUXILIARY
  DESC 'Profiles execution attributes'
  MAY ( SolarisKernelSecurityPolicy $ SolarisProfileType $ \
        SolarisAttrReserved1 $ SolarisAttrReserved2 $ \
        SolarisProfileId $ SolarisAttrKeyValue ) )

( 2.16.840.1.113894.1009.2.100.2.1 NAME 'SolarisQualifiedUserAttr'
  SUP top AUXILIARY
  DESC 'Host or netgroup qualified user attributes'
  MAY ( SolarisUserAttrEntry $ SolarisUserType ) )
```

Internet Print Protocol Information for LDAP

This section provides information about the attributes and object classes for the internet print protocol and the printer.

Internet Print Protocol Attributes

```
( 1.3.18.0.2.4.1140
NAME 'printer-uri'
DESC 'A URI supported by this printer.
This URI SHOULD be used as a relative distinguished name (RDN).
```

```
If printer-xri-supported is implemented, then this URI value
MUST be listed in a member value of printer-xri-supported.'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )

( 1.3.18.0.2.4.1107
NAME 'printer-xri-supported'
DESC 'The unordered list of XRI (extended resource identifiers) supported
by this printer.
Each member of the list consists of a URI (uniform resource identifier)
followed by optional authentication and security metaparameters.'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

( 1.3.18.0.2.4.1135
NAME 'printer-name'
DESC 'The site-specific administrative name of this printer, more end-user
friendly than a URI.'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127}  SINGLE-VALUE )

( 1.3.18.0.2.4.1119
NAME 'printer-natural-language-configured'
DESC 'The configured language in which error and status messages will be
generated (by default) by this printer.
Also, a possible language for printer string attributes set by operator,
system administrator, or manufacturer.
Also, the (declared) language of the "printer-name", "printer-location",
"printer-info", and "printer-make-and-model" attributes of this printer.
For example: "en-us" (US English) or "fr-fr" (French in France) Legal values of
language tags conform to [RFC3066] "Tags for the Identification of Languages".'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127}  SINGLE-VALUE )

( 1.3.18.0.2.4.1136
NAME 'printer-location'
DESC 'Identifies the location of the printer. This could include
things like: "in Room 123A", "second floor of building XYZ".'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch
```

```
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127} SINGLE-VALUE )

( 1.3.18.0.2.4.1139
NAME 'printer-info'
DESC 'Identifies the descriptive information about this printer.
This could include things like: "This printer can be used for
printing color transparencies for HR presentations", or
"Out of courtesy for others, please print only small (1-5 page)
jobs at this printer", or even "This printer is going away on July 1, 1997,
please find a new printer".'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127}
SINGLE-VALUE )

( 1.3.18.0.2.4.1134
NAME 'printer-more-info'
DESC 'A URI used to obtain more information about this specific printer.
For example, this could be an HTTP type URI referencing an HTML page
accessible to a Web Browser.
The information obtained from this URI is intended for end user consumption.'
EQUALITY caseIgnoreMatch ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )

( 1.3.18.0.2.4.1138
NAME 'printer-make-and-model'
DESC 'Identifies the make and model of the device.
The device manufacturer MAY initially populate this attribute.'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{127}  SINGLE-VALUE )

( 1.3.18.0.2.4.1133
NAME 'printer-ipp-versions-supported'
DESC 'Identifies the IPP protocol version(s) that this printer supports,
including major and minor versions,
i.e., the version numbers for which this Printer implementation meets
the conformance requirements.'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127} )

( 1.3.18.0.2.4.1132
NAME 'printer-multiple-document-jobs-supported'
DESC 'Indicates whether or not the printer supports more than one
document per job, i.e., more than one Send-Document or Send-Data
operation with document data.'
```

```
EQUALITY booleanMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE )

( 1.3.18.0.2.4.1109
NAME 'printer-charset-configured'
DESC 'The configured charset in which error and status messages will be
generated (by default) by this printer.
Also, a possible charset for printer string attributes set by operator,
system administrator, or manufacturer.
For example: "utf-8" (ISO 10646/Unicode) or "iso-8859-1" (Latin1).
Legal values are defined by the IANA Registry of Coded Character Sets and
the "(preferred MIME name)" SHALL be used as the tag.
For coherence with IPP Model, charset tags in this attribute SHALL be
lowercase normalized.
This attribute SHOULD be static (time of registration) and SHOULD NOT be
dynamically refreshed attributetypes: (subsequently).'
EQUALITY caseIgnoreMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{63} SINGLE-VALUE )

( 1.3.18.0.2.4.1131
NAME 'printer-charset-supported'
DESC 'Identifies the set of charsets supported for attribute type values of
type Directory String for this directory entry.
For example: "utf-8" (ISO 10646/Unicode) or "iso-8859-1" (Latin1).
Legal values are defined by the IANA Registry of Coded Character Sets and
the preferred MIME name.'
EQUALITY caseIgnoreMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{63} )

( 1.3.18.0.2.4.1137
NAME 'printer-generated-natural-language-supported'
DESC 'Identifies the natural language(s) supported for this directory entry.
For example: "en-us" (US English) or "fr-fr" (French in France).
Legal values conform to [RFC3066], Tags for the Identification of Languages.'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch SUBSTR caseIgnoreSubstringsMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{63} )

( 1.3.18.0.2.4.1130
NAME 'printer-document-format-supported'
DESC 'The possible document formats in which data may be interpreted
and printed by this printer.
Legal values are MIME types come from the IANA Registry of Internet Media Types.'
EQUALITY caseIgnoreMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127} )

( 1.3.18.0.2.4.1129
NAME 'printer-color-supported'
DESC 'Indicates whether this printer is capable of any type of color printing
at all, including highlight color.'
```

```
EQUALITY booleanMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.7  SINGLE-VALUE )

( 1.3.18.0.2.4.1128
NAME 'printer-compression-supported'
DESC 'Compression algorithms supported by this printer.
For example: "deflate, gzip".  Legal values include; "none", "deflate"
attributetypes: (public domain ZIP), "gzip" (GNU ZIP), "compress" (UNIX).'
EQUALITY caseIgnoreMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{255} )

( 1.3.18.0.2.4.1127
NAME 'printer-pages-per-minute'
DESC 'The nominal number of pages per minute which may be output by this
printer (e.g., a simplex or black-and-white printer).
This attribute is informative, NOT a service guarantee.
Typically, it is the value used in marketing literature to describe this printer.'
EQUALITY integerMatch
ORDERING integerOrderingMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.27  SINGLE-VALUE )

( 1.3.18.0.2.4.1126 NAME 'printer-pages-per-minute-color'
DESC 'The nominal number of color pages per minute which may be output by this
printer (e.g., a simplex or color printer).
This attribute is informative, NOT a service guarantee.
Typically, it is the value used in marketing literature to describe this printer.'
EQUALITY integerMatch
ORDERING integerOrderingMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.27  SINGLE-VALUE )

( 1.3.18.0.2.4.1125 NAME 'printer-finishings-supported'
DESC 'The possible finishing operations supported by this printer.
Legal values include; "none", "staple", "punch", "cover", "bind", "saddle-stitch",
"edge-stitch", "staple-top-left", "staple-bottom-left", "staple-top-right",
"staple-bottom-right", "edge-stitch-left", "edge-stitch-top", "edge-stitch-right",
"edge-stitch-bottom", "staple-dual-left", "staple-dual-top", "staple-dual-right",
"staple-dual-bottom".'
EQUALITY caseIgnoreMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{255} )

( 1.3.18.0.2.4.1124 NAME 'printer-number-up-supported'
DESC 'The possible numbers of print-stream pages to impose upon a single side of
an instance of a selected medium. Legal values include; 1, 2, and 4.
Implementations may support other values.'
EQUALITY integerMatch
ORDERING integerOrderingMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.27 )

( 1.3.18.0.2.4.1123 NAME 'printer-sides-supported'
```

```
DESC 'The number of impression sides (one or two) and the two-sided impression
rotations supported by this printer.
Legal values include; "one-sided", "two-sided-long-edge", "two-sided-short-edge".'
EQUALITY caseIgnoreMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127} )

( 1.3.18.0.2.4.1122 NAME 'printer-media-supported'
DESC 'The standard names/types/sizes (and optional color suffixes) of the media
supported by this printer.
For example: "iso-a4",  "envelope", or "na-letter-white".
Legal values  conform to ISO 10175, Document Printing Application (DPA), and any
IANA registered extensions.'
EQUALITY caseIgnoreMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{255} )

( 1.3.18.0.2.4.1117 NAME 'printer-media-local-supported'
DESC 'Site-specific names of media supported by this printer, in the language in
"printer-natural-language-configured".
For example: "purchasing-form" (site-specific name) as opposed to
(in "printer-media-supported"): "na-letter" (standard keyword from ISO 10175).'
EQUALITY caseIgnoreMatch SUBSTR caseIgnoreSubstringsMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{255} )

( 1.3.18.0.2.4.1121 NAME 'printer-resolution-supported'
DESC 'List of resolutions supported for printing documents by this printer.
Each resolution value is a string with 3 fields:
1) Cross feed direction resolution (positive integer), 2) Feed direction
resolution (positive integer), 3) Resolution unit.
Legal values are "dpi" (dots per inch) and "dpcm" (dots per centimeter).
Each resolution field is delimited by ">".  For example:  "300> 300> dpi>".'
EQUALITY caseIgnoreMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{255} )

( 1.3.18.0.2.4.1120 NAME 'printer-print-quality-supported'
DESC 'List of print qualities supported for printing documents on this printer.
For example: "draft, normal".  Legal values include; "unknown", "draft", "normal",
"high".'
EQUALITY caseIgnoreMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127} )

( 1.3.18.0.2.4.1110 NAME 'printer-job-priority-supported'
DESC 'Indicates the number of job priority levels supported.
An IPP conformant printer which supports job priority must always support a
full range of priorities from "1" to "100"
(to ensure consistent behavior), therefore this attribute describes the
"granularity".
 Legal values of this attribute are from "1" to "100".'
EQUALITY integerMatch
```

```
ORDERING integerOrderingMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.27  SINGLE-VALUE )

( 1.3.18.0.2.4.1118
NAME 'printer-copies-supported'
DESC 'The maximum number of copies of a document that may be printed as a single job.
A value of "0" indicates no maximum limit.
A value of "-1" indicates unknown.'
EQUALITY integerMatch
ORDERING integerOrderingMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.27  SINGLE-VALUE )

( 1.3.18.0.2.4.1111
NAME 'printer-job-k-octets-supported'
DESC 'The maximum size in kilobytes (1,024 octets actually) incoming print job that
this printer will accept.
A value of "0" indicates no maximum limit.  A value of "-1" indicates unknown.'
EQUALITY integerMatch
ORDERING integerOrderingMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.27  SINGLE-VALUE )

( 1.3.18.0.2.4.1113
NAME 'printer-service-person'
DESC 'The name of the current human service person responsible for servicing this
printer.
It is suggested that this string include information that would enable other humans
to reach the service person, such as a phone number.'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127}
SINGLE-VALUE )

( 1.3.18.0.2.4.1114
NAME 'printer-delivery-orientation-supported'
DESC 'The possible delivery orientations of pages as they are printed and ejected
from this printer.
Legal values include; "unknown", "face-up", and "face-down".'
EQUALITY caseIgnoreMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127} )

( 1.3.18.0.2.4.1115
NAME 'printer-stacking-order-supported'
DESC 'The possible stacking order of pages as they are printed and ejected from
this printer.
Legal values include; "unknown", "first-to-last", "last-to-first".'
EQUALITY caseIgnoreMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127} )

( 1.3.18.0.2.4.1116
NAME 'printer-output-features-supported'
```

```
DESC 'The possible output features supported by this printer.
Legal values include; "unknown", "bursting", "decollating", "page-collating",
"offset-stacking".'
EQUALITY caseIgnoreMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127} )

( 1.3.18.0.2.4.1108
NAME 'printer-aliases'
DESC 'Site-specific administrative names of this printer in addition the printer
name specified for printer-name.'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127} )

( 1.3.6.1.4.1.42.2.27.5.1.63
NAME 'sun-printer-bsdaddr'
DESC 'Sets the server, print queue destination name and whether the client generates
protocol extensions.
"Solaris" specifies a Solaris print server extension. The value is represented b the
following value: server "," destination ", Solaris".'
SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' SINGLE-VALUE )

( 1.3.6.1.4.1.42.2.27.5.1.64
NAME 'sun-printer-kvp'
DESC 'This attribute contains a set of key value pairs which may have meaning to the
print subsystem or may be user defined.
Each value is represented by the following: key "=" value.'
SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )
```

Internet Print Protocol `ObjectClasses`

```
objectclasses: ( 1.3.18.0.2.6.2549
NAME 'slpService'
DESC 'DUMMY definition'
SUP 'top' MUST (objectclass) MAY ())

objectclasses: ( 1.3.18.0.2.6.254
NAME 'slpServicePrinter'
DESC 'Service Location Protocol (SLP) information.'
AUXILIARY SUP 'slpService')

objectclasses: ( 1.3.18.0.2.6.258
NAME 'printerAbstract'
DESC 'Printer related information.'
ABSTRACT SUP 'top' MAY ( printer-name
$ printer-natural-language-configured
```

```
$ printer-location
$ printer-info
$ printer-more-info
$ printer-make-and-model
$ printer-multiple-document-jobs-supported
$ printer-charset-configured
$ printer-charset-supported
$ printer-generated-natural-language-supported
$ printer-document-format-supported
$ printer-color-supported
$ printer-compression-supported
$ printer-pages-per-minute
$ printer-pages-per-minute-color
$ printer-finishings-supported
$ printer-number-up-supported
$ printer-sides-supported
$ printer-media-supported
$ printer-media-local-supported
$ printer-resolution-supported
$ printer-print-quality-supported
$ printer-job-priority-supported
$ printer-copies-supported
$ printer-job-k-octets-supported
$ printer-current-operator
$ printer-service-person
$ printer-delivery-orientation-supported
$ printer-stacking-order-supported $ printer! -output-features-supported ))

objectclasses: ( 1.3.18.0.2.6.255
NAME 'printerService'
DESC 'Printer information.'
STRUCTURAL SUP 'printerAbstract' MAY ( printer-uri
$ printer-xri-supported ))

objectclasses: ( 1.3.18.0.2.6.257
NAME 'printerServiceAuxClass'
DESC 'Printer information.'
AUXILIARY SUP 'printerAbstract' MAY ( printer-uri $ printer-xri-supported ))

objectclasses: ( 1.3.18.0.2.6.256
NAME 'printerIPP'
DESC 'Internet Printing Protocol (IPP) information.'
AUXILIARY SUP 'top' MAY   ( printer-ipp-versions-supported $
printer-multiple-document-jobs-supported ))

objectclasses: ( 1.3.18.0.2.6.253
NAME 'printerLPR'
DESC 'LPR information.'
AUXILIARY SUP 'top' MUST ( printer-name ) MAY ( printer-aliases))
```

```
objectclasses: ( 1.3.6.1.4.1.42.2.27.5.2.14
NAME 'sunPrinter'
DESC 'Sun printer information'
SUP 'top' AUXILIARY MUST (objectclass $ printer-name)  MAY
(sun-printer-bsdaddr $ sun-printer-kvp))
```

Printer Attributes

```
ATTRIBUTE ( 1.3.6.1.4.1.42.2.27.5.1.63
NAME sun-printer-bsdaddr
DESC 'Sets the server, print queue destination name and whether the
     client generates protocol extensions. "Solaris" specifies a
     Solaris print server extension.  The value is represented by
     the following value: server "," destination ", Solaris".'
EQUALITY caseIgnoreIA5Match
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
)


ATTRIBUTE ( 1.3.6.1.4.1.42.2.27.5.1.64
NAME sun-printer-kvp
DESC 'This attribute contains a set of key value pairs which may have
      meaning to the print subsystem or may be user defined.  Each
      value is represented by the following: key "=" value.'
EQUALITY caseIgnoreIA5Match
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15  )
```

Sun Printer `ObjectClasses`

```
OBJECTCLASS ( 1.3.6.1.4.1.42.2.27.5.2.14
NAME sunPrinter
DESC 'Sun printer information'
SUP  top
AUXILIARY
MUST ( printer-name )
MAY  ( sun-printer-bsdaddr $ sun-printer-kvp ))
```

♦ ♦ ♦ CHAPTER 8

Transitioning From NIS to LDAP

This chapter describes how to enable support of NIS clients that use naming information stored in the LDAP directory. By following the procedures in this chapter, you can transition from using an NIS naming service to using the LDAP naming service.

For information about the benefits of transitioning to LDAP, see “Overview of the LDAP Naming Service” on page 11.

Note - The recommended and more cost effective procedure is to avoid using NIS-to-LDAP. First, set up your LDAP infrastructure; then, reconfigure your systems to directly use the LDAP infrastructure.

See the announcement about "Network Information Name Service (NIS)" in End of Features (EOF) Planned for Future Releases of Oracle Solaris (`https://www.oracle.com/solaris/technologies/end-of-feature-notices-solaris11.html#futurereleases`).

This chapter covers the following topics:

About the NIS-to-LDAP Service

The NIS-to-LDAP ("N2L" transition service replaces existing NIS daemons on the NIS master server with N2L transition daemons. The N2L service also creates an N2L mapping file on the NIS server. The mapping file specifies the mapping between NIS map entries and equivalent DIT entries in LDAP. An NIS master server that has gone through this transition is known as an N2L server. The slave servers do not have an `NISLDAPmapping` file, so they continue to function in the usual manner. The slave servers periodically update their data from the N2L server as if it were a regular NIS master.

The behavior of the N2L service is controlled by the `ypserv` and `NISLDAPmapping` configuration files. The `inityp2l` script assists the server with the initial setup of these configuration files. Once the N2L server has been established, you can edit the configuration file to maintain the N2L service.

The N2L service supports the following:

- Import of NIS maps into the LDAP DIT
- Client access to DIT information with the speed and extensibility of NIS

In the context of the N2L service, the term "map" is used in the following ways:

- To refer to a database file in which NIS stores a specific type of information
- To describe the process of mapping NIS information to or from the LDAP DIT

In any naming system, only one source of information can be the authoritative source. In traditional NIS, NIS sources are the authoritative information. When you use the N2L service, the source of authoritative data is the LDAP directory. The directory is managed by using directory management tools. For more information about directory management tools, see Chapter 1, “Introduction to the LDAP Naming Service”.

NIS sources are retained for emergency backup or backout only. After you use the N2L service, you must phase out NIS clients. Eventually, all NIS clients should be replaced by LDAP naming service clients.

The Service Management Facility (SMF) manages the NIS and LDAP services. You can perform administrative actions on these services, such as enabling, disabling, or restarting, by using the `svcadm` command. You can query the status of services by using the `svcs` command. For more information about using SMF with LDAP and NIS, see “LDAP and the Service Management Facility” on page 72 and “NIS and the Service Management Facility” in *Working With Oracle Solaris 11.4 Directory and Naming Services: DNS and NIS*. For information about SMF, refer to *Managing System Services in Oracle Solaris 11.4*. Also refer to the `svcadm`(8) and `svcs`(1) man pages for more details.

You need to be familiar with NIS and LDAP concepts, terminology, and IDs to perform the procedures in this chapter. For more information about the NIS and LDAP naming service, see the following sections:

- Chapter 5, “About the Network Information Service” in *Working With Oracle Solaris 11.4 Directory and Naming Services: DNS and NIS*, for an overview of NIS
- Chapter 1, “Introduction to the LDAP Naming Service”, for an overview of LDAP

Additional overview information is provided in the following sections:

- “When Not to Use the NIS-to-LDAP Service” on page 115
- “Effect of Installing the NIS-to-LDAP Service” on page 115
- “NIS-to-LDAP Commands, Files, and Maps” on page 116
- “Supported Standard Mappings” on page 117

When Not to Use the NIS-to-LDAP Service

The intent of the N2L service is to serve as a transition tool from using NIS to using LDAP. Do not use the N2L service in the following situations:

- In an environment where you do not plan to share data between NIS and LDAP naming service clients

  In such an environment, an N2L server would serve as an excessively complex NIS master server.
- In an environment where NIS maps are managed by tools that modify the NIS source files (other than `yppasswd`)

  Regeneration of NIS sources from DIT maps is an imprecise task that requires manual checking of the resulting maps. Once the N2L service is used, regeneration of NIS sources is provided only for backout or reverting to NIS.
- In an environment with no NIS clients

  In such an environment, use LDAP naming service clients and their corresponding tools.

Effect of Installing the NIS-to-LDAP Service

Installing the files that are related to the N2L service does not change the NIS server's default behavior. While installing the N2L service, you may see some changes to the NIS man pages and the addition of N2L helper scripts, `inityp2l` and `ypmap2src`, on the servers. However, as long as `inityp2l` is not run or the N2L configuration files are not created manually on the NIS server, the NIS components continue to start in traditional NIS mode and function as usual.

After `inityp2l` is run, users see some changes in server and client behavior. The following table lists the NIS and LDAP user types and a description of what each type of user should notice after the N2L service is deployed.

User Type	Effect of N2L Service
NIS master server administrators	The NIS master server is converted to an N2L server. The `NISLDAPmapping` and `ypserv` configuration files are installed on the N2L server. After the N2L server is established, you can use LDAP commands to administer your naming information.
NIS slave server administrators	After the N2L transition, an NIS slave server continues to run NIS in the usual manner. The N2L server pushes updated NIS maps to the slave server when `yppush` is called by `ypmake`. For more information, see the `ypmake`(8) man page.
NIS clients	NIS read operations are similar to traditional NIS. When an LDAP naming service client changes information in the DIT, the information is copied into the NIS maps. The copy operation is complete after a configurable timeout expires. This behavior is similar to the behavior of a normal NIS client when the client is connected to an NIS slave server. If an N2L server cannot bind to the LDAP server for a read, the N2L server returns the information from its own cached copy. Alternatively, the N2L server can return an internal server error. You can configure the N2L server to respond either way. For more information, see the `ypserv`(8) man page.
All users	When an NIS client makes a password change request, the change is immediately visible on the N2L master server and to native LDAP clients. If you attempt to change a password on the NIS client and the LDAP server is unavailable, then the change is refused and the N2L server returns an internal server error. This behavior prevents incorrect information from being written into the cache.

NIS-to-LDAP Commands, Files, and Maps

This section describes the utilities, configuration files, and mapping associated with the N2L transition.

The N2L transition uses the following utilities:

- `/usr/lib/netsvc/yp/inityp2l` – Assists with the creation of the `NISLDAPmapping` and `ypserv` configuration files but not the management of these files. If you are familiar with the technologies, you can maintain the N2L configuration files or create custom mappings by using a text editor to examine and customize the `inityp2l` output. For more information, see the `inityp2l`(8) man page.
- `/usr/lib/netsvc/yp/ypmap2src` – Converts standard NIS maps to approximations of the equivalent NIS source files. You can use the `ypmap2src` utility to convert from an N2L transition server to traditional NIS. For more information, see the `ypmap2src`(8) man page.

The N2L service uses the following files to transition from NIS to LDAP:

- `/var/yp/NISLDAPmapping` – Specifies the mapping between NIS map entries and equivalent DIT entries in LDAP. See the `NISLDAPmapping(5)` man page.
- `/var/yp/ypserv` – Specifies configuration information for the N2L transition daemons. For more information, see the `ypserv(5)` man page.

When the N2L transition is implemented, the `yppasswdd` command uses the `ageing.byname` mapping to read and write password aging information to the DIT.

Supported Standard Mappings

By default, the N2L service supports mappings between its standard maps and LDAP entries based on RFC 2307, RFC 2307bis, and later standards. Standard maps do not require manual modification of the mapping file. Any maps on your system that are not standard N2L service maps are considered custom maps and require manual modification.

The N2L service also supports automatic mapping of the `auto.*` maps. However, because most `auto.*` file names and contents are specific to each network configuration, those files are not specified in the list of standard maps. The exceptions are the `auto.home` and `auto.master` maps, which are supported as standard maps.

The N2L service supports the following standard maps:

```
audit_user
auth_attr
auto.home
auto.master
bootparams
ethers.byaddr ethers.byname
exec_attr
group.bygid group.byname group.adjunct.byname
hosts.byaddr hosts.byname
ipnodes.byaddr ipnodes.byname
mail.byaddr mail.aliases
netgroup netgroup.byprojid netgroup.byuser netgroup.byhost
netid.byname
netmasks.byaddr
networks.byaddr networks.byname
passwd.byname passwd.byuid passwd.adjunct.byname
prof_attr
project.byname project.byprojectid
protocols.byname protocols.bynumber
publickey.byname
rpc.bynumber
```

```
services.byname services.byservicename
timezone.byname
user_attr
```

During the N2L transition, the `yppasswdd` daemon uses the `ageing.byname` map to read and write password aging information to the DIT. If you are not using password aging, then the `ageing.byname` mapping is ignored.

Transitioning From NIS to LDAP Task Map

The following table identifies the procedures needed to install and manage the N2L service with standard and custom N2L mappings.

Task	Description	For Instructions
Complete all prerequisites.	Be sure that you have properly configured your NIS server and OUD (LDAP server).	"Prerequisites for the NIS-to-LDAP Transition" on page 118
Set up the N2L service.	Uses the `inityp2l` command on the NIS master server to set up either standard mappings or custom or nonstandard mappings.	"How to Set Up the N2L Service With Standard Mappings" on page 120 "How to Set Up the N2L Service With Custom or Nonstandard Mappings" on page 122
Customize a map.	Displays examples of custom maps for the N2L transition.	"Examples of Custom Maps" on page 124
Configure OUD with N2L.	Configures OUD as your LDAP server for the N2L transition.	"NIS-to-LDAP Best Practices With Oracle Unified Directory" on page 126
Troubleshoot the system.	Identifies and resolves common N2L issues.	"NIS-to-LDAP Troubleshooting" on page 129
Revert to NIS.	Revert to NIS using the appropriate map: Maps based on NIS source files Maps based on the DIT	 "How to Revert to Maps Based on NIS Source Files" on page 135 "How to Revert to Maps Based on DIT Contents" on page 136

Prerequisites for the NIS-to-LDAP Transition

Before implementing the N2L service, you must ensure the following items:

- Make sure that the system is set up as a working traditional NIS server before running the `inityp2l` script to enable N2L mode.
- Configure the LDAP directory server on your system.

  The N2L migration tools support OUD and compatible versions of directory servers offered by Oracle. The N2L migration tools also support OpenLDAP. If you use an OUD or OpenLDAP directory server, use the `ldapservercfg` command to configure the server *before* you set up the N2L service. For more information about the `ldapservercfg` command, see Chapter 4, “Setting Up an Oracle Unified Directory Server or OpenLDAP Server” and the `ldapservercfg(8)` man page.

  Although other third-party LDAP servers might work with the N2L service, they are not supported by Oracle. If you are using an LDAP server other than OpenLDAP directory server or OUD or compatible Oracle servers, you must manually configure the server to support the schemas of RFC 2307bis, RFC 4876, or later standards *before* you set up the N2L service.
- Use `files` before `dns` for the `config/host` property.
- Ensure that the addresses of the N2L master server and the LDAP server are present in the `hosts` file on the N2L master server.

  An alternative solution is to list the LDAP server address in `ypserv`, rather than its host name. Because the LDAP server address is listed in another place, changing the address of either the LDAP server or the N2L master server requires additional file modifications.

Setting Up the NIS-to-LDAP Service

You can use the standard mappings or custom mappings to set up the N2L service, as described in the procedures in this section.

As part of the NIS-to -LDAP conversion, you need to run the `inityp2l` command. This command runs an interactive script for which you must provide configuration information. For more information about the types of information you need to provide for configuration, see the `ypserv(8)` man page. This information typically includes:

- The name of the configuration file being created. The default configuration file is `/etc/default/ypserv`.
- The DN that stores configuration information in LDAP. The default value is `ypserv`.
- Preferred server list for mapping data to LDAP.
- Preferred server list for mapping data from LDAP.
- Authentication method for mapping data to LDAP.
- Authentication method for mapping data from LDAP.

- TLS method for mapping data to LDAP.
- TLS method for mapping data from LDAP.
- Proxy user bind DN to read or write data from LDAP.
- Proxy user bind DN to read or write data to LDAP.
- Proxy user password to read or write data from LDAP.
- Proxy user password to read or write data to LDAP.
- Timeout value (in seconds) for an LDAP bind operation.
- Timeout value (in seconds) for an LDAP search operation.
- Timeout value (in seconds) for an LDAP modify operation.
- Timeout value (in seconds) for an LDAP add operation.
- Timeout value (in seconds) for an LDAP delete operation.
- Time limit (in seconds) for search operation on the LDAP server.
- Size limit (in bytes) for search operation on the LDAP server.
- Whether N2L should follow LDAP referrals.
- LDAP retrieval error action, number of retrieval attempts, and timeout (in seconds) between each attempt.
- Store error action, number of attempts, and timeout (in seconds) between each attempt.
- Mapping file name.
- Whether to generate mapping information for `auto_direct` map.

  The script places relevant information regarding custom maps at appropriate places in the mapping file.
- The naming context.
- Whether to enable password changes.
- Whether to change the default TTL values for any map.

Note - Many LDAP servers do not support `sasl/cram-md5` authentication.

How to Set Up the N2L Service With Standard Mappings

Use this procedure if you are transitioning the maps listed in “Supported Standard Mappings” on page 117. If you are using custom or nonstandard maps, see “How to Set Up the N2L Service With Custom or Nonstandard Mappings” on page 122.

Before You Begin Complete the prerequisite steps that are listed in “Prerequisites for the NIS-to-LDAP Transition” on page 118.

1. **Become an administrator on the NIS master server.**
   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

2. **Convert the NIS master server into an N2L server.**

   ```
   # inityp2l
   ```

   Run the `inityp2l` script on the NIS master server and follow the prompts. `inityp2l` sets up the configuration and mapping files for standard and `auto.*` maps. For information about the list of the information you need to provide, see "Setting Up the NIS-to-LDAP Service" on page 119.

3. **Determine whether the LDAP DIT is fully initialized for the transition from the NIS source files.**
   The DIT is fully initialized if it already contains the information necessary to populate all the maps that are listed in the `NISLDAPmapping` file.

   - If the LDAP DIT is fully initialized, initialize the NIS maps.

     1. Stop the NIS service.

        ```
        # svcadm disable network/nis/server:default
        ```

     2. Initialize the NIS maps from information in the DIT.

        ```
        # ypserv -r
        ```

        Wait for `ypserv` to exit.

   - If the LDAP DIT is not fully initialized, initialize it.

     1. Make sure the NIS maps are up to date.

        ```
        # cd /var/yp
        # make
        ```

        For more information, see the `ypmake`(8) man page.

     2. Stop the NIS service.

        ```
        # svcadm disable network/nis/server:default
        ```

     3. Copy the NIS maps to the DIT and then initialize N2L support for the maps.

        ```
        # ypserv -Ir
        ```

        Wait for `ypserv` to exit.

4. **Start the DNS and NIS services to ensure that they use the new maps.**

```
# svcadm enable network/dns/client:default
# svcadm enable network/nis/server:default
```

▼ How to Set Up the N2L Service With Custom or Nonstandard Mappings

Use this procedure if the following circumstances apply:

- The maps you want to use are not listed in "Supported Standard Mappings" on page 117.
- Standard NIS maps need to be mapped to non-RFC 2307 LDAP mappings.

Before You Begin Complete the prerequisite steps that are listed in "Prerequisites for the NIS-to-LDAP Transition" on page 118.

1. **Become an administrator on the NIS master server.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

2. **Configure the NIS master server into the N2L server.**

   ```
   # inityp2l
   ```

   Run the `inityp2l` script on the NIS master server and follow the prompts. For the list of the information that you need to provide, see "Setting Up the NIS-to-LDAP Service" on page 119.

3. **Modify the `/var/yp/NISLDAPmapping` file.**

   For examples of how to modify the mapping file, see "Examples of Custom Maps" on page 124.

4. **Determine whether the LDAP DIT is fully initialized.**

   The DIT is fully initialized if it already contains the information necessary to populate all the maps that are listed in the `NISLDAPmapping` file. If the DIT is fully initialized, skip Step 5.

5. **Initialize the DIT for the transition from the NIS source files.**

   a. **Make sure that the old NIS maps are up-to-date.**

      ```
      # cd /var/yp
      ```

```
# make
```

For more information, see the ypmake(8) man page.

b. Stop the NIS daemons.

```
# svcadm disable network/nis/server:default
```

c. Copy the NIS maps to the DIT, then initialize N2L support for the maps.

```
# ypserv -Ir
```

Wait for ypserv to exit.

Tip - The original NIS dbm files are not overwritten. You can recover these files if needed.

---

d. Start the DNS and NIS service to ensure that they use the new maps.

```
# svcadm enable network/dns/client:default
# svcadm enable network/nis/server:default
```

e. Skip Step 6 and continue with Step 7.

6. Initialize the NIS maps.

Perform this step only if the DIT is fully initialized.

a. Stop the NIS daemons.

```
# svcadm disable network/nis/server:default
```

b. Initialize the NIS maps from information in the DIT.

```
# ypserv -r
```

Wait for ypserv to exit.

Tip - The original NIS dbm files are not overwritten. You can recover these files if needed.

---

c. Start the DNS and NIS service to ensure that they use the new maps.

```
# svcadm enable network/dns/client:default
# svcadm enable network/nis/server:default
```

7. Verify whether the LDAP entries are correct.

If the entries are incorrect, then the entries cannot be found by LDAP naming service clients.

```
# ldapsearch -H ldapuri -s sub \
    -b "ou=servdates, dc=..." "objectclass=servDates"
```

8. **Verify the contents of the LDAP maps.**

   The following sample output shows how to use the `makedbm` command to verify the contents of the `hosts.byaddr` map.

```
# makedbm -u LDAP_servdate.bynumber
plato: 1/3/2001
johnson: 2/4/2003,1/3/2001
yeats: 4/4/2002
poe: 3/3/2002,3/4/2000
```

   If the contents are as expected, the transition from NIS to LDAP was successful.

Examples of Custom Maps

Examples in this section show how you might customize maps. Use your preferred text editor to modify the `/var/yp/NISLDAPmapping` file as needed. For more information about file attributes and syntax, see the `NISLDAPmapping(5)` man page. For more information about the LDAP naming service, see Chapter 1, “Introduction to the LDAP Naming Service”.

EXAMPLE 1 Moving Host Entries

This example shows how to move host entries from the default location to another location in the DIT by changing the `nisLDAPobjectDN` attribute in the `NISLDAPmapping` file to the new base LDAP distinguished name (DN). For this example, the internal structure of the LDAP objects is unchanged, so `objectClass` entries are also unchanged..

Change:

```
nisLDAPobjectDN hosts: \
ou=hosts,?one?, \
objectClass=device, \
objectClass=ipHost
```

to:

```
nisLDAPobjectDN hosts: \
ou=newHosts,?one?, \
objectClass=device, \
```

```
objectClass=ipHost
```

This change causes entries to be mapped under `dn: ou=newHosts, dom=domain1, dc=sun, dc=com`, instead of `dn: ou=hosts, dom=domain1, dc=sun, dc=com`.

EXAMPLE 2 Implementing a Custom Map

This example shows how to implement a custom map.

In this example the `servdate.bynumber` map contains information about the servicing dates for systems. This map is indexed by the system's serial number, which in this example is `123`. Each entry consists of the system owner's name, a colon, and a comma-separated list of service dates, such as `John Smith:1/3/2001,4/5/2003`.

The old map structure is to be mapped onto LDAP entries of the following form:

```
dn: number=123,ou=servdates,dc=... \
number: 123 \
userName: John Smith \
date: 1/3/2001 \
date: 4/5/2003 \
.
.
.
objectClass: servDates
```

By examining the `NISLDAPmapping` file, you can see that the mapping closest to the required pattern is `group`. The custom mappings can be modeled on the `group` mapping. Because there is only one map, no `nisLDAPdatabaseIdMapping` attribute is required. The attributes to be added to `NISLDAPmapping` are as follows:

```
nisLDAPentryTtl servdate.bynumber:1800:5400:3600

nisLDAPnameFields servdate.bynumber: \
("%s:%s", uname, dates)

nisLDAPobjectDN servdate.bynumber: \
ou=servdates, ?one? \
objectClass=servDates:

nisLDAPattributeFromField servdate.bynumber: \
dn=("number=%s,", rf_key), \
number=rf_key, \
userName=uname, \
(date)=(dates, ",")

nisLDAPfieldFromAttribute servdate.bynumber: \
```

```
rf_key=number, \
uname=userName, \
dates=("%s,", (date), ",")
```

NIS-to-LDAP Best Practices With Oracle Unified Directory

The N2L service supports OUD. Although other third-party LDAP servers might work with the N2L service, they are not supported by Oracle. If you are using an LDAP server other than an OUD server or compatible Oracle servers, you must manually configure the server to support the schemas of RFC 2307, RFC 2307bis and RFC 4876, or later standards.

If you are using OUD, you can enhance the directory server to improve performance. To make these enhancements, you must have LDAP administrator privileges on the OUD server. In addition, you must coordinate with the LDAP clients if the directory server need to be rebooted. The OUD documentation is available at Oracle Unified Directory documentation.

Creating Virtual List View Indexes With Oracle Unified Directory

For large maps, you must use the LDAP virtual list view (VLV) indexes to ensure that LDAP searches return complete results. For information about setting up VLV indexes on OUD, see the Oracle Unified Directory documentation.

VLV search results use a fixed page size of 50000. If you are using VLVs with OUD, ensure that both the LDAP server and N2L server are able to handle transfers of this size. If all of your maps are known to be smaller than this limit, you do not need to use VLV indexes. However, if your maps are larger than the size limit or you are unsure of the size of all maps, use VLV indexes to avoid incomplete returns.

If you are using VLV indexes, set up the appropriate size limits as follows:

- On the OUD server, ensure that the `nsslapd-sizelimit` attribute is set to greater than or equal to 50000 or `-1`. For more information, see the `ldapservercfg(8)` man page.
- On the N2L server, ensure that the `nisLDAPsearchSizelimit` attribute is set to either greater than or equal to 50000 or zero. For more information, see the `NISLDAPmapping(5)` man page.

After VLV indexes have been created, activate them by running `dsadm` with the `vlvindex` option on the OUD server. For more information, see the `dsadm`(8) man page.

VLVs for Standard Maps

Use the OUD `ldapservercfg` command to set up VLVs if the following conditions apply:

- You are using OUD.
- You are mapping standard maps to RFC 2307bis LDAP entries.

VLVs are domain specific, so each time `ldapservercfg` is run, VLVs are created for one NIS domain. Therefore, during the N2L transition, you must run `ldapservercfg` once for *each* `nisLDAPdomainContext` attribute included in the `NISLDAPmapping` file.

VLVs for Custom and Nonstandard Maps

You must manually create new OUD VLVs for maps, or copy and modify existing VLV indexes, if the following conditions apply:

- You are using OUD.
- You have large custom maps or have standard maps that are mapped to nonstandard DIT locations.

To view existing VLV indexes, type the following command:

```
% ldapsearch -H ldapuri -s sub -b "cn=ldbm database,cn=plugins,cn=config"
 "objectclass=vlvSearch"
```

Avoiding Server Timeouts With Oracle Unified Directory

When the N2L server refreshes a map, the result might require a lengthy LDAP directory access. If OUD is not correctly configured, the refresh operation might time out before completion. To avoid directory server timeouts, modify OUD attributes manually or by running the `ldapservercfg` command.

For example, you might want to modify the following attributes to increase the minimum amount of time in seconds that the server should spend performing the search request:

```
dn: cn=config
nsslapd-timelimit: -1
```

For testing purposes, you can use an attribute value of `-1`, which indicates no limit. When you have determined the optimum limit value, change the attribute value. Do *not* maintain any

attribute settings at -1 on a production server. With no limits, the server might be vulnerable to Denial of Service attacks.

For more information about configuring OUD with LDAP, see Chapter 4, “Setting Up an Oracle Unified Directory Server or OpenLDAP Server”.

Avoiding Buffer Overruns With Oracle Unified Directory

To avoid buffer overruns, modify the following attributes manually or by running the `ldapservercfg` command.

EXAMPLE 3 Increase the Maximum Number of Entries Returned

This example shows how to set attributes to increase the maximum number of entries that are returned for a client search query.

The following example is specific to OpenLDAP:

```
dn: cn=config
changetype: modify
replace: olcSizeLimit
olcSizeLimit: -1
```

The following example is specific to OUD:

```
dn: cn=MyRootUser,cn=Root DNs,cn=config
changetype: modify
add: ds-rlim-lookthrough-limit
ds-rlim-size-limit: -1
```

The attribute value `-1` indicates no limit. A value of `-1` can be used for testing purposes. When you have determined the optimum limit value, change the attribute value.

Note - Do *not* maintain any attribute settings at `-1` on a production server. With no limits, the server might be vulnerable to Denial of Service attacks.

If VLVs are being used, the `sizelimit` attribute values should be set as defined in “Creating Virtual List View Indexes With Oracle Unified Directory” on page 126. If VLVs are not being used, the size limit should be set large enough to accommodate the largest container.

EXAMPLE 4 Increase the Maximum Number of Entries Verified

This example shows how to set attributes to increase the maximum number of entries that are verified for a client search query.

The following example is specific to OUD. OpenLDAP does not have an equivalent to the `lookthrough-limit` attribute.

```
dn: cn=MyRootUser,cn=Root DNs,cn=config
changetype: modify
add: ds-rlim-lookthrough-limit
ds-rlim-lookthrough-limit: -1
```

NIS-to-LDAP Restrictions

When the N2L server has been set up, the NIS source files are no longer used. Therefore, do not run `ypmake` on an N2L server. If `ypmake` is accidentally run, such as for an existing `cron` job, the N2L service is unaffected. However, a warning is logged suggesting that `yppush` should be called explicitly.

NIS-to-LDAP Troubleshooting

This section covers the following areas of troubleshooting:

- “Common LDAP Error Messages” on page 129
- “NIS-to-LDAP Issues” on page 131

Common LDAP Error Messages

The N2L server might log errors that relate to internal LDAP problems, resulting in LDAP-related error messages. Although the errors are nonfatal, they indicate that you need to investigate the problems. The N2L server might continue to operate but provide out-of-date or incomplete results.

This section describes some of the common LDAP error messages that you might encounter when implementing the N2L service. It also includes error descriptions and possible causes and solutions for the errors.

```
Administrative limit exceeded
Error Number: 11
```

Cause: An LDAP search was larger than the limit allowed by the directory server's `nsslapd-sizelimit` attribute. The search returns partial information.

Solution: Increase the value of the `nsslapd-sizelimit` attribute or implement a VLV index for the failing search.

```
Invalid DN Syntax
Error Number: 34
```

Cause: An attempt has been made to write an LDAP entry with a DN that contains illegal characters. The N2L server attempts to escape illegal characters, such as the + symbol, that are generated in DNs.

Solution: Check the LDAP server error log to find out which illegal DNs were written and modify the `NISLDAPmapping` file that generated the illegal DNs.

```
Object class violation
Error Number: 65
```

Cause: An attempt has been made to write an LDAP entry that is invalid. Generally, this error is due to missing `MUST` attributes that can be caused by either of the following circumstances:

- Bugs in the `NISLDAPmapping` file that create entries with missing attributes
- Attempts to add an `AUXILIARY` attribute to an object that does not exist

  For example, if a user name has not yet been created from the `passwd.byxxx` map, an attempt to add auxiliary information to that user will fail.

Solution: For bugs in the `NISLDAPmapping` file, check the information in the server error log to determine the nature of the problem.

```
Can't contact LDAP server
Error Number: 81
```

Cause: The `ypserv` file might be incorrectly configured to point to the wrong LDAP directory server. Alternatively, the directory server might not be running.

Solution: Perform the following actions to resolve the issue:.

- Reconfigure the `ypserv` file to point to the correct LDAP directory server.

- Type the following command to confirm that the LDAP server is running:

  ```
  % ping hostname 5 | grep "no answer" || \
          (ldapsearch -H ldapuri -s base -b ""  \
          "objectclass=*" >/dev/null && echo Directory accessible)
  ```

  If the server is unavailable, the following message is displayed:

  ```
  no answer from hostname
  ```

  If there are problems with the LDAP server, the following message is displayed:

  ```
  ldap_search: Can't connect to the LDAP server - Connection refused
  ```

  If everything is working, the following message is displayed:

  ```
  Directory accessible
  ```

```
Timeout
Error Number: 85
```

Cause: An LDAP operation timed out while updating a map from the DIT. The map might now contain out-of-date information.

Solution: Increase the `nisLDAPxxxTimeout` attributes in the `ypserv` configuration file.

NIS-to-LDAP Issues

This section describes problems that could occur while running the N2L server and provides possible causes and solutions.

Debugging the `NISLDAPmapping` File

The mapping file, `NISLDAPmapping`, is complex. Different issues might cause the mapping to behave in unexpected ways. Use the described techniques to resolve such problems.

```
Console Message Displays When ypserv -ir (or -Ir) Runs
```

Description: A simple message is displayed on the console and the server exits (a detailed description is written to `syslog`).

Cause: The syntax of the mapping file might be incorrect.

Solution: Check and correct the syntax in the `NISLDAPmapping` file.

`NIS Daemon Exits at Startup`

Description: When `ypserv` or other NIS daemons run, an LDAP-related error message is logged and the daemon exits.

Cause: The cause might be one of the following:

- The LDAP server cannot be contacted.
- An entry found in an NIS map or in the DIT is incompatible with the mapping specified.
- An attempt to read or write to the LDAP server returns an error.

Solution: Examine the error log on the LDAP server. For the information about LDAP errors, see "Common LDAP Error Messages" on page 129.

`Unexpected Results From NIS Operations`

Description: NIS operations do not return the expected results but no errors are logged.

Cause: Incorrect entries might exist in the LDAP or NIS maps, which results in mappings not completing as intended.

Solution: Check and correct entries in the LDAP DIT and in the N2L versions of the NIS maps.

1. Check that the correct entries exist in the LDAP DIT, and fix the entries as needed.

   If you are using OUD, start the management console by running the `dsadm startconsole` command.
2. Check that the N2L versions of the NIS maps in the `/var/yp` directory contain the expected entries by comparing the newly generated map to the original map. Fix entries as needed.

   ```
   # cd /var/yp/domain-name
   # makedbm -u test.byname
   ```

   Be aware of the following when checking the output for the maps:

   - The order of entries might not be the same in both files.

     Use the `sort` command before comparing output.
   - The use of white space might not be the same in both files.

     Use the `diff -b` command when comparing output.

```
Processing Order of NIS Maps
```

Description: Object class violations have occurred.

Cause: When the `ypserv -i` command is run, each NIS map is read and its contents are written into the DIT. Several maps might contribute attributes to the same DIT object. Generally, one map creates most of the object, including all of the object's `MUST` attributes. Other maps contribute additional `MAY` attributes.

Maps are processed in the same order that `nisLDAPobjectDN` attributes appear in the `NISLDAPmapping` file. If maps containing `MAY` attributes get processed before maps containing `MUST` attributes, then object class violations occur. For more information about this error, see "Common LDAP Error Messages" on page 129.

Solution: Reorder the `nisLDAPobjectDN` attributes so that maps are processed in the correct order.

As a temporary fix, rerun the `ypserv -i` command several times. Each time the command is executed, the LDAP entry approaches a complete state.

Note - Mapping in such a way that all of an object's `MUST` attributes cannot be created from at least one map is *not* supported.

N2L Server Timeout Issue

```
The server times out.
```

Cause: When the N2L server refreshes a map, the result might require a single lengthy access of a large LDAP directory. If OUD is not correctly configured, this operation might time out before completion.

Solution: To avoid directory server timeouts, modify the OUD attributes manually or by running the `ldapservercfg` command. For more information, see "Common LDAP Error Messages" on page 129 and "NIS-to-LDAP Best Practices With Oracle Unified Directory" on page 126.

N2L Lock File Issue

```
The ypserv command starts but does not respond to NIS requests.
```

Cause: The N2L server lock files are not correctly synchronizing access to the NIS maps.

Solution: Type the following commands on the N2L server:

1. Stop the NIS server.

   ```
   # svcadm disable network/nis/server:default
   ```

2. Remove the lock files.

   ```
   # rm /var/run/yp_maplock /var/run/yp_mapupdate
   ```

3. Restart the NIS server.

   ```
   # svcadm enable network/nis/server:default
   ```

N2L Deadlock Issue

```
The N2L server deadlocks.
```

Cause: If the addresses of the N2L master server and the LDAP server are not listed properly in the `hosts`, `ipnodes`, or `ypserv` files, a deadlock might result. For more information about address configuration for N2L, see “Prerequisites for the NIS-to-LDAP Transition” on page 118.

For an example of a deadlock scenario, consider the following sequence of events:

1. An NIS client tries to look up an IP address.
2. The N2L server finds that the `hosts` entry is out of date.
3. The N2L server tries to update the `hosts` entry from LDAP.
4. The N2L server gets the name of its LDAP server from `ypserv`, then does a search by using the LDAP client library.
5. The LDAP client library tries to convert the LDAP server's name to an IP address by making a call to the name service switch.
6. The name service switch might make an NIS call to the N2L server, which deadlocks.

Solution: List the addresses of the N2L master server and the LDAP server in the `hosts` or `ipnodes` files on the N2L master server. Whether the server addresses must be listed in `hosts`, `ipnodes`, or both files depends on how these files are configured to resolve local host names. Also, check that the `config/hosts` property of the `svc:/network/name-service/switch` service lists `files` before `nis` in the lookup order.

An alternative solution to this deadlock problem is to list the LDAP server address, not its host name, in the `ypserv` file. Because the LDAP server address would be listed in another

place, changing the address of either the LDAP server or the N2L server would require slightly more effort.

Reverting to NIS

A site that has transitioned from NIS to LDAP using the N2L service is expected to gradually replace all NIS clients with LDAP naming services clients. Support for NIS clients eventually becomes redundant. However, if required, the N2L service provides two ways to return to NIS, as explained in the procedures in this section.

Tip - Because traditional NIS ignores the N2L versions of the NIS maps if those maps are present, you can safely leave the N2L versions of the maps on the server. Keeping the N2L maps might be useful in case you later decide to re-enable N2L.

▼ How to Revert to Maps Based on NIS Source Files

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

2. **Stop the `NIS` daemons.**

   ```
   # svcadm disable network/nis/server:default
   ```

3. **Disable N2L.**

   ```
   # mv /var/yp/NISLDAPmapping backup-filename
   ```

   This command backs up and moves the N2L mapping file.

4. **Set the `NOPUSH` environment variable so the new maps are not pushed by `ypmake`.**

   ```
   # NOPUSH=1
   ```

5. **Make a new set of NIS maps that are based on the NIS sources.**

   ```
   # cd /var/yp
   # make
   ```

6. **(Optional) Remove the N2L versions of the NIS maps.**

```
# rm /var/yp/domain-name/LDAP_*
```

7. **Start the DNS and the NIS service.**

```
# svcadm enable network/dns/client:default
# svcadm enable network/nis/server:default
```

How to Revert to Maps Based on DIT Contents

Back up the old NIS source files before performing this procedure.

1. **Become an administrator.**

   For more information, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

2. **Stop the `NIS` daemons.**

```
# svcadm disable network/nis/server:default
```

3. **Update the maps from the DIT.**

```
# ypserv -r
```

   Wait for `ypserv` to exit.

4. **Disable N2L.**

```
# mv /var/yp/NISLDAPmapping backup-filename
```

   This command backs up and moves the N2L mapping file.

5. **Regenerate the NIS source files.**

```
# ypmap2src
```

6. **Manually check that the regenerated NIS source files have the correct content and structure.**

7. **Move the regenerated NIS source files to the appropriate directories.**

8. **(Optional) Remove the N2L versions of the mapping files.**

```
# rm /var/yp/domain-name/LDAP_*
```

9. **Start the DNS and NIS service.**

```
# svcadm enable network/dns/client:default
# svcadm enable network/nis/server:default
```

LDAP Glossary

attribute Each LDAP entry consists of a number of named attributes, each of which has one or more values.

Also, the N2L service mapping and configuration files each consist of a number of named attributes. Each attribute has one or more values.

baseDN The DN where part of the DIT is rooted. When this is the baseDN for an NIS domains entries it is also referred to as a *context*.

context For the N2L service, a context is something under which a NIS domain is generally mapped. See also *baseDN*.

custom map Any map that is not a standard map and therefore requires manual modifications to the mapping file when transitioning from NIS to LDAP.

directory An LDAP directory is a container for LDAP objects. In UNIX, a container for files and subdirectories.

directory cache A local file used to store data associated with directory objects.

directory information tree (DIT) The DIT is the distributed directory structure for a given network. By default, clients access the information assuming that the DIT has a given structure. For each domain supported by the LDAP server, there is an assumed subtree with an assumed structure.

DIT See directory information tree.

entry A single row of data in a database table, such as an LDAP element in a DIT.

LDAP Lightweight Directory Access Protocol is a standard, extensible directory access protocol used by LDAP naming service clients and servers to communicate with each other.

mapping file The `NISLDAPmapping` file that establishes how to map entries between NIS and LDAP files.

searchTriple A description of where to look for a given attribute in the DIT. The searchTriple is composed of a base dn, scope, and filter. This is part of the LDAP URL format as defined in RFC 2255.

Index

N

O

P