



@server

iSeries

Networking

Directory Services (LDAP)





@server

iSeries

Networking

Directory Services (LDAP)

Contents

Part 1. Directory Services (LDAP)	1
Chapter 1. Whats new for V5R2	3
Chapter 2. Print this topic	5
Chapter 3. Get started with Directory Services	7
LDAP basics.	8
Considerations for using LDAP V2 with LDAP V3.	11
Plan your LDAP directory server	11
Migrate to V5R2 from an earlier release of Directory Services	11
Migrate from V4R3 or V4R4 Directory Services to V5R2	12
Install and configure Directory Services	14
Configure the LDAP directory server	14
Default configuration for Directory Services	15
The IBM SecureWay Directory Management Tool.	16
Chapter 4. Administer the LDAP directory server.	17
Start the LDAP directory server	17
Stop the LDAP directory server	18
Check the status of the directory server	18
Check jobs on the LDAP directory server.	18
Enable event notification	18
Specify transaction settings	19
Change the port or IP address.	19
Move LDAP directory data between systems	20
Import an LDIF file	20
Export an LDIF file	20
Set up a new replica of the directory server	20
Publish information to the directory server	24
Specify a server for directory referrals	26
Add suffixes to the LDAP directory server	26
Remove suffixes from the directory server	27
Save and restore Directory Services information	27
Manage ownership and access of directory data	27
Work with the ownership properties of directory objects	27
Work with access control lists (ACLs)	28
Work with ACL Groups	28
Work with administrative access for authorized users	28
Track access and changes to the LDAP directory.	29
Enable object auditing for the directory server	29
Adjust performance of the LDAP directory server	30
Chapter 5. Directory Services concepts and reference information.	31
LDAP access control lists (ACLs)	31
LDAP data interchange format.	32
National language support (NLS) considerations	35
Ownership of LDAP directory objects	35
LDAP directory referrals	35
Transactions	35
Replica LDAP directory servers	36
Directory Services security	36
Use Secure Sockets Layer (SSL) and Translation Layer Security with the LDAP directory server	37

Use Kerberos authentication with the LDAP directory server	37
Operating system projected backend	38
OS/400 user projected directory information tree	39
LDAP operations.	39
Administrator and replica bind DN's	43
OS/400 user-projected schema	43
Directory Services and OS/400 journaling support	44
Chapter 6. LDAP command line utilities	45
ldapmodify and ldapadd utilities	45
Examples: ldapmodify and ldapadd	47
ldapdelete utility	48
Example: ldapdelete	49
ldapsearch utility	50
Examples: ldapsearch	52
ldapmodrdn utility	55
Example: ldapmodrdn	56
Notes about using SSL with the LDAP command line utilities	56
Chapter 7. Troubleshoot Directory Services	59
Basic troubleshooting procedure for Directory Services.	59
Monitor errors and access with the Directory Services job log	60
Use TRCTCPAPP to help find problems	60
Use the LDAP_OPT_DEBUG option to trace errors	60
Common LDAP client errors	61
ldap_search: Timelimit exceeded	62
[Failing LDAP operation]: Operations error	62
ldap_bind: No such object	62
ldap_bind: Inappropriate authentication	62
[Failing LDAP operation]: Insufficient access	62
[failing LDAP operation]: Cannot contact LDAP server	62
[failing LDAP operation]: Failed to connect to ssl server	63

Part 1. Directory Services (LDAP)

Directory Services provides a Lightweight Directory Access Protocol (LDAP) server on the iSeries server. LDAP runs over Transmission Control Protocol/Internet Protocol (TCP/IP) and is popular as a directory service for both Internet and non-Internet applications.

If you are familiar with Directory Services, you might want to start by reading about what's new for this release. If you want, you can print or display a PDF version of the Directory Services information.

The following topics introduce Directory Services and provide you with information to help you administer the LDAP server on your iSeries™ server:

Chapter 3, "Get started with Directory Services" on page 7

Chapter 4, "Administer the LDAP directory server" on page 17

Chapter 5, "Directory Services concepts and reference information" on page 31


Chapter 6, "LDAP command line utilities" on page 45

Chapter 7, "Troubleshoot Directory Services" on page 59

For additional information about Directory Services, visit the Directory Services web page  .



The LDAP server that Directory Services provides is an IBM® SecureWay® Directory  .

Chapter 1. Whats new for V5R2

- Directory Services has the following enhancements and new features.
- Directory Services is part of the base operating system beginning in V5R1. Option 32 is no longer available starting in V5R2.
 - New security enhancements have been made to further protect any data stored on the directory server.
 - LDAP directory server can now be used as a domain controller for an Enterprise Identity Mapping (EIM) domain.
 - A new option is available to administrators that can be used to grant administrator access to the directory server for users who have been given access to the Directory Services Administrator (QIBM_DIRSRV_ADMIN) function identifier (ID) of the operating system through iSeries Navigator application support.
 - You can select to have the directory server use specific IP addresses or you can select to use all configured IP addresses on the server. See “Change the port or IP address” on page 19 for more information.
 - The **ldap_set_option** API has a new debug trace feature for V5R2. The LDAP_OPT_DEBUG option can be used to help diagnose problems with clients that use the LDAP C APIs. For more information, see “Use the LDAP_OPT_DEBUG option to trace errors” on page 60 or see the Directory Services APIs in the iSeries Information Center .

How to see whats new or changed:

To help you see where technical changes have been made, this information uses:





- The  image to mark where new or changed information begins.
- The  image to mark where new or changed information ends.

Chapter 2. Print this topic

To view or download the PDF version, select Directory Services (LDAP) (about 323 KB or 66 pages).

Other information


You can also view or print any of the following PDFs:

- *LDAP Implementation Cookbook*  .
- *Understanding LDAP*  .
- *Using LDAP for Directory Integration: A Look at IBM SecureWay Directory, Active Directory, and Domino™*  .
- | • *Implementation and Practical Use of LDAP on the iSeries Server*  .

To save a PDF on your workstation for viewing or printing:

1. Open the PDF in your browser (click the link above).
2. In the menu of your browser, click **File**.
3. Click **Save As...**
4. Navigate to the directory in which you would like to save the PDF.
5. Click **Save**.

Downloading Adobe Acrobat Reader

If you need Adobe Acrobat Reader to view or print these PDFs, you can download a copy from the Adobe Web site (www.adobe.com/products/acrobat/readstep.html)  .

Chapter 3. Get started with Directory Services

Directory Services provides a Lightweight Directory Access Protocol (LDAP) server on the iSeries server. LDAP runs over Transmission Control Protocol/Internet Protocol (TCP/IP) and is gaining popularity as a directory service for both Internet and non-Internet applications. You perform most setup and administering tasks of the OS/400-based LDAP directory server through the graphical user interface (GUI) of iSeries Navigator. To administer Directory Services, you must have iSeries Navigator installed on a PC that is connected to your iSeries server. You can use Directory Services with LDAP-enabled applications, such as mail applications that look up e-mail addresses from LDAP servers.

Besides the LDAP server, Directory Services also includes:

- An OS/400-based LDAP client. This client includes a set of application program interfaces (APIs) that you can use in OS/400® programs to create your own client applications. For information about these APIs, see the Directory Services topic under Programming in the iSeries Information Center.
- Version 3.2 of the IBM SecureWay Directory Client Software Development Kit (SDK). The SDK includes a Windows® LDAP client and the following tools:
 - The IBM SecureWay Directory Management Tool, which provides you with a graphical user interface for managing directory content.
 - command line utilities (ldapsearch, ldapadd, etc.)
 - C LDAP APIs (library files, header files, and sample source code)
 - IBM JNDI LDAP service provider (ibmjndi.jar)
 - online documentation for all of the above items. See the readme file for the location and names of these HTML files.

If you have used Directory Services with an earlier release of OS/400, see “Migrate to V5R2 from an earlier release of Directory Services” on page 11.





For an introduction to LDAP, see “LDAP basics” on page 8. If you have used LDAP servers on other platforms you should take a few minutes to read this topic as it contains some OS/400-specific information.

When you have familiarized yourself with the basic information, proceed to “Plan your LDAP directory server” on page 11.


For information on installing and configuring your directory server, see “Install and configure Directory Services” on page 14.

Documentation

| The Directory Services Information Center topic provides an overview of LDAP and concentrates
| specifically on managing the LDAP directory server on OS/400. This documentation also provides full
| documentation for the SecureWay Directory Client SDK. For additional LDAP information, consult LDAP
| references such as the following:

- | • *LDAP Implementation Cookbook*  .
- | • *Understanding LDAP*  .
- | • *Using LDAP for Directory Integration: A Look at IBM SecureWay Directory, Active Directory, and Domino*  .
- | • *Implementation and Practical Use of LDAP on the iSeries server*  .
- | • *LDAP: Programming Directory-enabled Applications with Lightweight Directory Access Protocol* by Tim
| Howes and Mark Smith.

- *Understanding and Deploying LDAP Directory Services* by Mark C. Smith, Gordon S. Good, and Tim Howes.

Additional information about Directory Services on the iSeries server is available at the iSeries server Directory Services home page .

Note: Some of the material contained in this document is a derivative of LDAP documentation provided by the University of Michigan. Copyright © 1992-1996, Regents of the University of Michigan, All Rights Reserved.

LDAP basics

The Lightweight Directory Access Protocol (LDAP) is a directory service protocol that runs over Transmission Control Protocol/Internet Protocol (TCP/IP). LDAP version 2 is formally defined in Internet Engineering Task Force (IETF) Request for Comments (RFC) 1777, *Lightweight Directory Access Protocol*. LDAP version 3 is formally defined in IETF RFC 2251, *Lightweight Directory Access Protocol (v3)*. You can view these RFCs on the Internet at the following URL:

[!\[\]\(950a62bbddad88d64435fd35607dfc42_img.jpg\)http://www.ietf.org](http://www.ietf.org)

The LDAP directory service follows a client/server model. One or more LDAP servers contain the directory data. An LDAP client connects to an LDAP Server and makes a request. The server responds with a reply, or with a pointer (a referral) to another LDAP server.

Uses of LDAP:

Because LDAP is a directory service rather than a database, the information in an LDAP directory is usually descriptive, attribute-based information. LDAP users generally read the information in the directory much more often than they change it. Updates are typically simple all-or-nothing changes. Common uses of LDAP directories include online telephone directories and e-mail directories.

LDAP directory structure:

The LDAP directory service model is based on **entries** (which are also referred to as **objects**). Each entry consists of one or more **attributes**, such as a name or address, and a **type**. The types typically consist of mnemonic strings, such as `cn` for common name or `mail` for e-mail address.

The example directory in Figure 1 on page 10 shows an entry for Tim Jones that includes *mail* and *telephoneNumber* attributes. Some other possible attributes include *fax*, *title*, *sn* (for surname), and *jpegPhoto*.

Each directory has a **schema**, which is a set of rules that determine the structure and contents of the directory. You should use the IBM SecureWay Directory Management Tool (DMT) to edit the schema files for your LDAP server. After you install Directory Services, the files are located on your system at `/QIBM/UserData/OS400/DirSrv`.

Note: Original copies of the default schema files are located at `/QIBM/ProdData/OS400/DirSrv`. If you need to replace the files in the UserData directory, you can copy these files to the `/QIBM/ProdData/OS400/DirSrv` directory.

Each directory entry has a special attribute called **objectClass**. This attribute controls which attributes are required and allowed in an entry. In other words, the values of the objectClass attribute determine the schema rules the entry must obey.

Each directory entry also has the following **operational attributes**, which the LDAP server automatically maintains:

- `CreatorsName`, which contains the bind DN used when creating the entry.
- `CreateTimestamp`, which contains the time at which the entry was created.
- `modifiersName`, which contains the bind DN used when the entry was last modified (initially this is the same as `CreatorsName`).
- `modifyTimestamp`, which contains the time at which the entry was last modified (initially this is the same as `CreateTimestamp`).

Traditionally, LDAP directory entries are arranged in a hierarchical structure that reflects political, geographic, or organizational boundaries (see Figure 1 on page 10). Entries that represent countries appear at the top of the hierarchy. Entries representing states or national organizations occupy the second level down in the hierarchy. The entries below that can then represent people, organizational units, printers, documents, or other items.

You are not limited to the traditional hierarchy when structuring your directory. The domain component structure, for example, is gaining popularity. With this structure, entries are composed of the parts of TCP/IP domain names. For example, `dc=ibm,dc=com` may be preferable to `o=ibm,c=us`.

LDAP refers to entries with **Distinguished Names (DNs)**. Distinguished names consist of the name of the entry itself as well as the names, in order from bottom to top, of the objects above it in the directory. For example, the complete DN for the entry in the bottom left corner of Figure 1 on page 10 is `cn=Tim Jones, o=IBM, c=US`. Each entry has at least one attribute that is used to name the entry. This naming attribute is called the **Relative Distinguished Name (RDN)** of the entry. The entry above a given RDN is called its **parent Distinguished Name**. In the example above, `cn=Tim Jones` names the entry, so it is the RDN. `o=IBM, c=US` is the parent DN for `cn=Tim Jones`.

To give an LDAP server the capability to manage part of an LDAP directory, you specify the highest level parent distinguished names in the configuration of the server. These distinguished names are called **suffixes**. The server can access all objects in the directory that are below the specified suffix in the directory hierarchy. For example, if an LDAP server contained the directory shown in Figure 1 on page 10, it would need to have the suffix `o=ibm, c=us` specified in its configuration in order to be able to answer client queries regarding Tim Jones.

LDAP Directory Structure

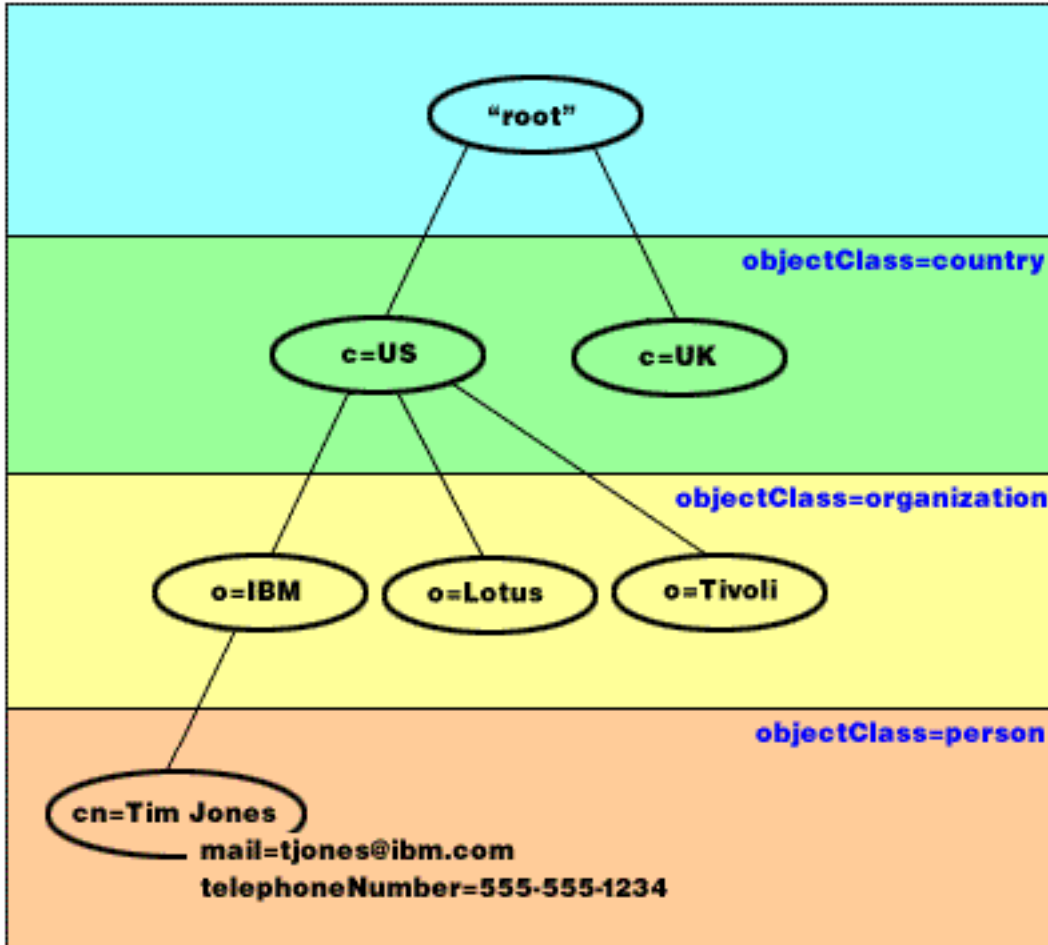


Figure 1. Basic LDAP directory structure

Notes™ about LDAP and Directory Services:

- Beginning with V4R5, both the OS/400 LDAP server and the OS/400 LDAP client are based on LDAP Version 3. You can use a V2 client with a V3 server. However, you cannot use a V3 client with a V2 server unless you bind as a V2 client and use only V2 APIs. See LDAP V2/V3 considerations for more details.
- The Windows LDAP client is also based on LDAP Version 3.
- Because LDAP is a standard, all LDAP servers share many basic characteristics. However, due to implementation differences, they are not all completely compatible with each other. The LDAP server provided by Directory Services is closely compatible with other LDAP directory servers in the IBM SecureWay Directory and IBM Directory product group. However, it may not be as compatible with other LDAP servers.
- The data for the LDAP server that Directory Services provides resides in an OS/400 database.

More information:

- | For examples of using LDAP directories, see the following:
 - | • Section 1.6 The Quick Start: A Public LDAP Example, in the redbook *Understanding LDAP*.
 - | • Section 3.3 Example Scenarios, in the redbook *Understanding LDAP*.

To learn about more LDAP concepts, see Chapter 5, “Directory Services concepts and reference information” on page 31.

Considerations for using LDAP V2 with LDAP V3

Beginning with V4R5, both the OS/400 LDAP server and the OS/400 LDAP client are based on LDAP Version 3. You cannot use a V3 client with a V2 server. However, you can use the `ldap_set_option()` API to change the version of a V3 client to V2. Then you can successfully send in client requests to a V2 server.

You can use a V2 client with a V3 server. Be aware that on a search request, however, the V3 server may send back data in the full range of UTF-8 format, while a V2 client may be only able to handle data in the IA5 character set.

Note: LDAP version 2 is formally defined in Internet Engineering Task Force (IETF) Request for Comments (RFC) 1777, *Lightweight Directory Access Protocol*. LDAP version 3 is formally defined in IETF RFC 2251, *Lightweight Directory Access Protocol (v3)*. You can view these RFCs on the Internet at the following URL:

<http://www.ietf.org> 

Plan your LDAP directory server

Before you install Directory Services and begin to configure your LDAP directory, you should take a few minutes to plan the directory. Important things to consider include the following:

- **Organize the directory.** Plan the structure of your directory and determine what suffixes and attributes your server will require.
- **Decide how large your directory will be.** You can then estimate how much storage you need. The size of the directory depends on the following:
 - The number of attributes in the servers schema.
 - The number of entries on the server.
 - The type of information that you store on the server.

For example, an empty directory that uses the default Directory Services schema requires approximately 10 MB of storage space. A directory that uses the default schema and which contains 1000 entries of typical employee information requires about 30 MB of storage space. This number will vary depending on the exact attributes that you used. It will also increase greatly if you stored large objects, such as pictures, in the directory.

- **Decide what security measures you will take.** Directory Services supports the use of Secure Sockets Layer (SSL) and Digital Certificates as well as Translation Layer Security (TLS) for communication security. Beginning with V5R1, Kerberos authentication is also supported.
- Directory Services allows you to control access to directory objects with access control lists (ACLs). You can also use OS/400 security auditing to protect the directory.

Migrate to V5R2 from an earlier release of Directory Services

| V5R2 of OS/400 introduces new features and capabilities to Directory Services. These changes affect both
| the LDAP directory server and the graphical user interface (GUI) of iSeries Navigator. To take advantage
| of the new GUI features, you need to install iSeries Navigator on a PC that can communicate over TCP/IP
| to your iSeries server. iSeries Navigator is a component of iSeries Access for Windows. If you have an
| earlier version of iSeries Navigator installed, you should upgrade to V5R2.

| V5R2 of OS/400 supports upgrades from V4R5 and V5R1. When you upgrade to V5R2 of OS/400, both
| the LDAP directory data and the directory schema files are automatically migrated to conform to V5R2
| formats. If you have a Directory Services LDAP server running under V4R3 or V4R4 of OS/400 and want
| to migrate the server to V5R2, you need to perform some additional migration tasks.

When you upgrade to V5R2 of OS/400, you should be aware of some migration issues:

- When you upgrade to V5R2, Directory Services automatically migrates your schema files to V5R2 and deletes the old schema files. However, if you have deleted or renamed the schema files, Directory Services cannot migrate them. You may receive an error or Directory Services may assume that the files have already been migrated.
- Directory Services migrates directory data to the V5R2 format the first time that you start the server or import an LDIF file. Plan to allow some time for this migration to complete. If you are upgrading to V5R2 from V4R4 or earlier, be aware that the directory data will require approximately twice as much storage space in V5R2 than it required previously. This is because in V4R4 or earlier versions, Directory Services supported only the IA5 character set and saved data in ccsid 37 (single byte format). Directory Services supports the full ISO 10646 character set.
After you upgrade to V5R2, you should start your server once to migrate existing data before importing new data. If you try to import data before starting the server once and you do not have enough authority, the import may fail.
- V4R4 and earlier releases of Directory Services did not take time zones into account when creating time stamp entries. Beginning with V4R5, the time zone is used in all additions and modifications to the directory. Therefore, if you upgrade to V5R2 from V4R4 or earlier, Directory Services adjusts existing `createtimestamp` and `modifytimestamp` attributes to reflect the correct time zone. It does this by subtracting the time zone that is currently defined on the iSeries system from the time stamps that are stored in the directory. Note that if the current time zone is not the same time zone that was active when the entries were originally created or modified, the new time stamp values will not reflect the original time zone.
- Following migration, the LDAP directory server will automatically start when TCP/IP starts. If you do not want the directory server to start automatically, use iSeries Navigator to change the setting.

Migrate from V4R3 or V4R4 Directory Services to V5R2

V5R2 of OS/400 does not support direct upgrades from V4R3. If you want to migrate a V4R3 or V4R4 Directory Services LDAP server to V5R2, you can follow either of the following procedures:


- Slip installing OS/400 from V4R3 or V4R4 to an interim release
- Saving the database library and scratch installing OS/400 from V4R3 or V4R4 to V5R2

Slip installing OS/400 from V4R3 or V4R4 to an interim release

Though upgrades from V4R3 and V4R4 of OS/400 to V5R2 are not supported, the following upgrades are supported:

- V4R3 and V4R4 upgraded to V4R5
- V4R4 and V4R5 upgraded to V5R1
- V4R5 and V5R1 upgraded to V5R2

One way to migrate your Directory Services server is to upgrade to an interim release (V4R5 or V5R1),

then to V5R2. For detailed information on OS/400 installation procedures, see *Software Installation* . Follow these general steps to perform the migration:

1. Note any changes that you have made to the schema files in the `/QIBM/UserData/OS400/DirSrv` directory. The schema files are migrated automatically.
2. For V4R4 or V4R3, do the slip install of V4R5 or V5R1 of OS/400.
3. Do the slip install to V5R2 of OS/400.
4. Start the Directory Services server if not already started.
5. Use the Directory Management Tool to modify the schema files for any user changes that you noted in step 1.
6. Restart the Directory Services server.

Saving the database library and scratch installing OS/400 from V4R3 or V4R4 to V5R2

The other way to migrate your Directory Services server is to save the database library that Directory Services uses in V4R3 or V4R4, then restore it after scratch installing V5R2. This saves you the step of installing an interim release. However, the servers settings are not migrated, so you must reconfigure the

server settings. For detailed information on OS/400 installation procedures, see *Software Installation* .

Follow these general steps to perform the migration:

1. Note any changes that you have made to the schema files in the /QIBM/UserData/OS400/DirSrv directory. The schema files are not migrated automatically, so if you want to keep your changes you will need to manually implement them again.
2. Note the various configuration settings in the Directory Services servers properties, including the database library name.
3. Save the database library that is specified in the Directory Services servers configuration.
4. Note the publishing configuration.
5. Scratch install the system to V5R2 of OS/400.
6. Use EZ-Setup to configure the Directory Services server.
7. Restore the database library that you saved in step 3.
8. Use the Directory Management Tool to modify the schema files for any user changes that you noted in step 1.
9. Use iSeries Navigator to reconfigure Directory Services. Specify the database library that you saved and restored.
10. Use iSeries Navigator to reconfigure publishing.
11. Restart the Directory Services server.

Upgrade Issues

When you upgrade from V4R3 to any later release, you should be aware of the following issues:

- **Migrating the key ring file to a key database:**

V3R2 Client Access used key ring files to establish Secure Sockets Layer (SSL) connections to the LDAP directory server. iSeries Access for Windows uses certificate stores, which are sometimes called key databases, to establish SSL connections. If you used a key ring file with your LDAP directory server previously, the key ring file must be converted to a key database in order to continue using SSL. The first time that you attempt to start an SSL connection to the LDAP directory server, iSeries Navigator will alert you to this change. If you choose to convert the key, you are prompted to specify some information for the key database before the conversion is made.

The LDAP directory server also used a key ring file for its own SSL connections in V4R3. Beginning with V4R4 it uses the system certificate store. If your server was set up to use SSL in V4R3, the contents of the key ring file will be migrated to the system certificate store.

- **Two stream files have been removed:**

The following stream files used by Directory Services in V4R3 are no longer needed and are automatically removed when you install a later release:

```
/QIBM/ProdData/OS400/DirSrv/qgldcert.kyr  
/QIBM/ProdData/OS400/DirSrv/qgldcert.sth
```

You do not need to take any action with these files. This is mentioned only so that you are not concerned if you notice that they are no longer present on your system.

Also be aware that there may be additional issues associated with upgrading to the current release from other releases.

Install and configure Directory Services

Directory Services (LDAP) is automatically installed when you install OS/400. The directory server includes a default configuration that automatically starts the directory server when TCP/IP is started. The directory server also starts publishing computer information from OS/400 to the directory server. To customize the LDAP directory servers settings, run the Directory Services Configuration Wizard. You must have *ALLOBJ and *IOSYSCFG special authorities in order to use the wizard.

Directory Services is integrated into the base operating system beginning with V5R1 and Option 32 is no longer available starting in V5R2.

Configure the LDAP directory server

If your system has not been configured to publish information to another LDAP server and no LDAP servers are known to the TCP/IP DNS server, then Directory Services is automatically installed with a limited default configuration. Directory Services provides a wizard to assist you in configuring the LDAP directory server for your specific needs. You may run this wizard as part of EZ-Setup or run the wizard later from iSeries Navigator. Use this wizard when you initially configure the directory server. You may also use the wizard to reconfigure the directory server.

Note: When you use the wizard to reconfigure the directory server, you start configuring from scratch. The original configuration is deleted rather than changed. However, the directory data is not deleted, but instead remains stored in the library that you selected upon installation (QUSRDIRDB by default). The change log also remains intact, in the QUSRDIRCL library by default.

If you want to start completely from scratch, clear those two libraries before starting the wizard.

If you want to change the directory server configuration, but not clear it completely, right-click **Directory** and select **Properties**. This does not delete the original configuration. You must have *ALLOBJ and *IOSYSCFG special authorities to configure the server. If you want to configure OS/400 security auditing, you must also have *AUDIT special authority.

To start the Directory Services Configuration Wizard, take these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Configure**.

Note: If you have already configured the directory server, click **Reconfigure** rather than **Configure**.

Follow the instructions in the Configure Directory Server wizard to configure your LDAP directory server.

Note: You may also want to put the library that stores the directory data in a user auxiliary storage pool (ASP) rather than the system ASP. However, this library cannot be stored in an Independent ASP and any attempt to configure, reconfigure, or start the server with a library that exists in an Independent ASP will fail.

When the wizard is finished, your LDAP directory server has a basic configuration. If you are running Lotus® Domino on your system, then port 389 (the default port for the LDAP server) may already be in use by Dominos LDAP function. You must do one of the following:

- Change the port that Lotus Domino uses
- Change the port that Directory Services uses
- Use specific IP addresses

You may start the server at this point. Before starting the server, however, you may want to do some or all of the following:

- Import data to the server
- Enable Secure Sockets Layer (SSL) security
- Enable Kerberos authentication
- Set up a referral

Enable SSL on the LDAP directory server

If you have Digital Certificate Manager installed on your system, you can use Secure Sockets Layer (SSL) security to protect access to your LDAP directory server. Before enabling SSL on the directory server, you may find it helpful to read an overview on using SSL with Directory Services.

To use an SSL connection when you administer your LDAP directory server from iSeries Navigator, or to use SSL with the Windows LDAP client, you must have one of the Client Encryptions products (5722CE2 or 5722CE3) installed on your PC.

To enable SSL on your LDAP server, use the Digital Certificate Manager interface. You can launch Digital Certificate Manager from the **Internet** folder in iSeries Navigator, or from the **Network** page of the directory servers **Properties** dialog.

To launch the Digital Certificate Interface from the **Network** page, follow these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Properties**.
5. Click the **Network** tab.
6. Click **Digital Certificate Manager**.

Digital Certificate Manager will launch in your default Internet browser.

See Securing the LDAP directory server for the specific steps that you need to follow in order to assign a digital certificate to the directory server.

After SSL is enabled, you can change the port that the LDAP directory server uses for secured connections.

Enable Kerberos authentication on the LDAP directory server

If you have Network Authentication Service configured on your system, you can set up your LDAP directory server to use Kerberos authentication. Before enabling Kerberos on the directory server, you may find it helpful to read an overview on using Kerberos with Directory Services.

To enable Kerberos authentication, follow these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Properties**.
5. Click the **Kerberos** tab.
6. Check **Enable Kerberos authentication**.
7. Specify other settings on the **Kerberos** page as appropriate to your situation. See the pages online help for information about individual fields.

Default configuration for Directory Services

The LDAP directory server is automatically installed when you install OS/400. This installation includes a default configuration. The directory server uses the default configuration when all of the following are true:

- Administrators have not run the Directory Services Configuration Wizard or changed directory settings with the properties pages.
- Directory Services publishing is not configured.
- The LDAP directory server cannot find any LDAP DNS information.

- If the LDAP directory server uses the default configuration, then the following occur:
- The LDAP directory server automatically starts when TCP/IP starts.
 - The system creates a default administrator, cn=Administrator. It also generates a password that is used internally. If you need to use an administrator password later, you can set a new one from the Directory Services property page.
 - A default suffix is created that is based on the systems IP name. A system objects' suffix is also created based on the system name. For example, if your systems IP name is mary.acme.com, the suffix is dc=mary,dc=acme,dc=com.
 - The LDAP directory server uses the default data library QUSRDIRDB. The system creates it in the system ASP.
 - The server uses port 389 for non-secure communications. If a digital certificate has been configured for LDAP, secure sockets layer (SSL) is enabled and port 636 is used for secure communications.

The following defaults then exist for Directory Services publishing:

- The system publishes information to the local LDAP directory server
- Publishing does not use SSL
- Publishing uses containers under the default suffix
- For authentication to the directory server, OS/400 uses the cn=Administrator ID and the system-generated password.
- The system publishes only system information

The IBM SecureWay Directory Management Tool

The IBM SecureWay Directory Management Tool (DMT) provides you with a graphical user interface for managing LDAP directory content. The tasks that you can perform with the DMT include the following:

- Browsing directory schema
- Adding, editing, and deleting object classes
- Adding, editing, and deleting attributes
- Browsing and searching the directory tree
- Adding, editing, viewing, and deleting entries
- Editing entry RDNs
- Managing ACLs

The DMT is part of the Windows LDAP client that is included with Directory Services. The client is shipped in an integrated file system directory.

To install the Windows LDAP client, including the DMT, onto a PC, follow these steps:

1. In iSeries Navigator, expand **File Systems**.
2. Expand **File Shares**.
3. Double-click **Qdirsrv**.
4. Double-click **UserTools**.
5. Double-click **Windows**.
6. Double-click **setup.exe** to start installing the DMT. Follow the on-screen instructions to complete the installation.

Documentation for the IBM SecureWay Directory Management Tool (DMT) is located in the file dparent.htm. This file is copied to the IBM SecureWay Directory folder on your PC when you install the client.

Chapter 4. Administer the LDAP directory server

To administer the LDAP directory server, you must have the following authority sets:

- To configure the server or change the server configuration: All Object (*ALLOBJ) and I/O System Configuration (*IOSYSCFG) special authorities
- To start or stop the server: Job Control (*JOBCTL) authority and object authority to the End TCP/IP (ENDTCP), Start TCP/IP (STRTCP), Start TCP/IP Server (STRTCPSVR), and End TCP/IP Server (ENDTCPSVR) commands
- To set auditing behavior for the directory server: Audit (*AUDIT) special authority
- To view the server job log: Spool Control (*SPLCTL) special authority

To manage directory objects (including access control lists, object ownership, and replicas), connect to the directory with either the administrator DN or another DN that has the proper LDAP authority. If authority integration is being used, an administrator can also be a projected user that has authority to the Directory Services Administrator function ID.

Administering the directory server includes the following tasks:

- “Start the LDAP directory server”
- “Stop the LDAP directory server” on page 18
- “Check the status of the directory server” on page 18
- “Check jobs on the LDAP directory server” on page 18
- “Enable event notification” on page 18
- “Specify transaction settings” on page 19
- “Change the port or IP address” on page 19
- “Move LDAP directory data between systems” on page 20
- “Specify a server for directory referrals” on page 26
- “Add suffixes to the LDAP directory server” on page 26
- “Remove suffixes from the directory server” on page 27
- “Save and restore Directory Services information” on page 27
- “Manage ownership and access of directory data” on page 27
- “Track access and changes to the LDAP directory” on page 29
- “Enable object auditing for the directory server” on page 29
- “Adjust performance of the LDAP directory server” on page 30

Start the LDAP directory server

To start the LDAP directory server, take these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Start**.

The directory server may take several minutes to start, depending on the speed of your server and the amount of available memory. The first time you start the directory server may take several minutes longer than usual because the server must create new files. Similarly, when starting the directory server for the first time after upgrading from an earlier version of Directory Services, it may take several minutes longer than usual because the server must migrate files. You can check the status of the server periodically to see if it has started yet.

Note: The directory server can also be started from a 5250 session by entering the command STRTCPSVR *DIRSRV.

Additionally, if you have your directory server configured to start when TCP/IP starts, you can also start it by entering the STRTCP command.

Stop the LDAP directory server

Stopping the directory server affects all applications using the server at the time it is stopped. This includes Enterprise Identity Mapping (EIM) applications that are currently using the directory server for EIM operations. All applications are disconnected from the directory server, however, they are not prevented from attempting to reconnect to the server.

To stop the LDAP directory server, take these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Stop**.

The directory server may take several minutes to stop, depending on the speed of your system, the amount of server activity, and the amount of available memory. You can check the status of the server periodically to see if it has stopped yet.

Note: The directory server can also be stopped from a 5250 session by entering the commands `ENDTCPSVR *DIRSRV`, `ENDTCPSVR *ALL`, or `ENDTCP`. `ENDTCPSVR *ALL` and `ENDTCP` also affect any other TCP/IP servers that run on your system. `ENDTCP` will also end TCP/IP itself.

Check the status of the directory server

iSeries Navigator displays the status of the directory server in the **Status** column in the right frame.

To check the status of the directory server, take these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**. iSeries Navigator displays the status of all TCP/IP servers, including the directory server, in the **Status** column. To update the status of the servers, click the **View** menu and select **Refresh**.
4. To view more information about the status of the directory server, right-click **Directory** and select **Status**. This will show you the number of active connections, as well as other information such as past and current activity levels.

Besides providing additional information, viewing status through this option can save time. You can refresh the status of the directory server without taking the additional time that is required to check the status of the other TCP/IP servers.

Check jobs on the LDAP directory server

At times you may want to monitor specific jobs on the LDAP directory server. To check server jobs, take these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Server Jobs**.

Enable event notification


Directory Services supports event notification, which allows clients to register with the LDAP server to be notified when a specified event, such as something being added to the directory, occurs.

To enable event notification for your server, follow these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Properties**.

5. Click **Events**.
6. Select **Allow clients to register for event notification**.

You can also specify the maximum registrations allowed for each connection and the maximum total registrations that the server allows.

For additional information on event notification, see the Appendix C: Event Notification in the manual IBM SecureWay Directory Version 3.2: Client SDK Programming Reference .

Specify transaction settings

Directory Services supports transactions, which allows a group of LDAP directory operations to be treated as one unit. For more information, see “Transactions” on page 35.

To configure your servers transaction settings, follow these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Properties**.
5. Click **Transactions**.
6. Specify your transaction settings.

Note: Transaction settings can impact your LDAP servers performance, so you may want to experiment with different settings.

Change the port or IP address

The LDAP directory server enabled by Directory Services uses the following default ports:

- 389 for unsecured connections.
- 636 for secured connections (if you have used Digital Certificate Manager to enable Directory Services as an application that can use a secure port).

Note: By default, all IP addresses defined on the local system are bound to the server.

If you are already using these ports for another application, you can either assign a different port to Directory Services, or you can use different IP addresses for the two servers, if the applications support binding to a specific IP address.

For an example of the Domino LDAP server conflicting with the iSeries Directory Services LDAP server, see Host Domino LDAP and Directory Services on the same iSeries

To change the ports that the LDAP directory server uses, take these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Properties**.
5. Click the **Network** tab.
6. Enter the appropriate port numbers, then click **OK**.

To change the IP address on which the directory server accepts connections, take these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Properties**.

- | 5. Click the **Network** tab.
- | 6. Click the **IP Addresses...** button.
- | 7. Select **Use selected IP addresses** and select the IP addresses for the server to use when accepting connections.

Move LDAP directory data between systems

Your Directory Services LDAP server can run independently of other servers. However, you may find it useful to have it work with other servers. This can include:

- “Import an LDIF file”
- “Export an LDIF file”
- “Set up a new replica of the directory server”
- “Publish information to the directory server” on page 24

Import an LDIF file

You can transfer information between different LDAP directory servers by using LDAP Data Interchange Format (LDIF) files. Before you begin this procedure, transfer the LDIF file to your iSeries server as a stream file.

To import an LDIF file to the LDAP directory server, take these steps:

1. If the directory server is started, stop it. See “Stop the LDAP directory server” on page 18 for information on stopping the directory server.
2. In iSeries Navigator, expand **Network**.
3. Expand **Servers**.
4. Click **TCP/IP**.
5. Right-click **Directory** and select **Tools**, then **Import File**.

Note: You can also use the `ldapadd` utility to import LDIF files.

Export an LDIF file

You can transfer information between different LDAP directory servers by using LDAP Data Interchange Format (LDIF) files, see “LDAP data interchange format” on page 32. You can export all or part of your LDAP directory to an LDIF file.

To export an LDIF file from the directory server, take these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Tools**, then **Export File**.

Note: If you do not specify a location for the LDIF file to be exported to, it will be saved to the default directory specified in your OS/400 user profile. If you have not changed your default directory, the default directory is the root directory.

Notes:

1. Be sure to set authority to the LDIF file to prevent unauthorized access to directory data. To do this, right-click on the file in iSeries Navigator, then select **Permissions**.
2. You can also create a full or partial LDIF file with the `ldapsearch` utility, see “`ldapsearch` utility” on page 50. Use the `-L` option and redirect the output to a file.

Set up a new replica of the directory server

You can set up replicas of the LDAP directory server to directory servers on other iSeries servers. Directory Services uses the standard LDAP version 3 protocol to replicate.

Notes:

1. You cannot replicate between LDAP version 3 and LDAP version 2 servers. Therefore, the system that you replicate to must be using the same version of LDAP as the system from which you replicate. V4R3 and V4R4 of OS/400 support LDAP version 2. V4R5 and later releases support LDAP version 3.
2. You can replicate the Directory Services directory to IBM SecureWay V3.2 or later servers on other platforms. To do this, your OS/400 directory server must be configured to use the 3.2 ACI mechanism. If the server encounters a problem when it is trying to replicate, it will stop replicating. If that happens, your replica will be incomplete.

Follow these steps to set up a new replica of the directory server:

1. If you have not already done so, configure both the master server and the replica server.

Note: Make sure that the schema and suffixes match on both servers.

2. Stop the master server.
3. (optional) Set up LDAP data for initial replication. You can skip this step if you do not have any initial data that you want to transfer to the replica server from the master server.
4. (optional) Move LDAP data to the master server. Skip this step if one of the following applies to your replica server:
 - It is a new LDAP directory server.
 - It does not contain data that you want to continue to maintain.
5. Set up the new replica server.
6. Set up the master server to have a new replica.
7. Make sure the master server is allowing updates:
 - a. In iSeries Navigator, expand the system on which the master directory server runs.
 - b. Expand **Network**.
 - c. Expand **Servers**.
 - d. Click **TCP/IP**.
 - e. Right-click **Directory** and select **Properties**.
 - f. If it is not already checked, check **Allow directory updates**.

Note: These instructions assume that both the master server and the replica servers are on systems that you manage from iSeries Navigator on the same PC. If you are managing your systems from separate PCs, you can move between two PCs to perform this task. If either the master or replica server is running on an IBM operating system other than OS/400, refer to the documentation for that platform to set up that server.

Set up LDAP data for initial replication

You may have existing data on your master LDAP directory server that you want to add to a new replica server. To do this, you first need to export the directory to an LDIF file. While the LDIF file is exporting, you must prevent the master server from being updated. You may do this in one of the following ways:

- Stop the LDAP directory server. Depending on the amount of data in your directory, this may require that your server stay stopped for an extended period of time.
- Change the server properties so that updates are not allowed. This allows the server to continue answering search requests while the LDIF file is being exported. To take this option, follow these steps:
 1. In iSeries Navigator, expand the system on which the master directory server runs.
 2. Expand **Network**.
 3. Expand **Servers**.
 4. Click **TCP/IP**.
 5. Right-click **Directory** and select **Properties**.
 6. If **Allow directory updates** is checked, uncheck it. This will prevent updates to the directory until replication is completely set up.
 7. Click **OK**.
 8. Stop, then restart, the LDAP directory server.

After you have stopped the server or changed the server properties to disallow directory updates, perform these tasks:

1. Export the directory to an LDIF file.
2. Transfer the LDIF file to the system on which the replica server will run.

After the LDIF file is transferred to the system on which the replica server will run, you need to import the data to the replica server:

1. In iSeries Navigator, expand the system on which the replica directory server runs.
2. If the replica server is not already stopped, stop it now. Refresh the status of the servers until the status is **Stopped**.
3. Expand **Network**.
4. Expand **Servers**.
5. Click **TCP/IP**.
6. Right-click **Directory** and select **Properties**.
7. If **Allow directory updates** is unchecked, check it. This will allow the data to be imported.
8. Click **OK**.
9. Import the LDIF file that you transferred in step 2.
10. Right-click **Directory** and select **Properties**.
11. Uncheck **Allow directory updates**.

Move LDAP data to the master server

Once you make an LDAP directory server into a replica server, you can no longer update the data on it. If you have existing data on the server that you are configuring to be a replica LDAP directory server, you will probably want to move it to the master server so that it can continue to be maintained. To do this, follow these steps:

1. In iSeries Navigator, expand the system on which the replica directory server runs.
2. Expand **Network**.
3. Expand **Servers**.
4. Click **TCP/IP**.
5. Right-click **Directory** and select **Properties**.
6. If **Allow directory updates** is checked, uncheck it. This will prevent updates to the directory until replication is completely set up.
7. Click **OK**.
8. Stop the LDAP directory server.
9. Export the directory to an LDIF file.
10. Transfer the LDIF file to the system on which the master server will run.

After the LDIF file is transferred to the system on which the master server will run, you need to import the data to the master server:

1. In iSeries Navigator, expand the system on which the master directory server runs.
2. If the master directory server is not already stopped, stop it now. Refresh the status of the servers until the status is **Stopped**.
3. Expand **Network**.
4. Expand **Servers**.
5. Click **TCP/IP**.
6. Right-click **Directory** and select **Properties**.
7. If **Allow directory updates** is unchecked, check it. This will allow the data to be imported.
8. Click **OK**.
9. Import the LDIF file that you transferred in step 10 of the previous procedure.
10. Right-click **Directory** and select **Properties**.
11. Uncheck **Allow directory updates**.

Set up the new replica

Follow these steps to set up the new replica server.

Note: The replica server must be configured and stopped before you perform this procedure.

1. In iSeries Navigator, expand the system on which the replica directory server runs.
2. Expand **Network**.

3. Expand **Servers**.
4. Click **TCP/IP**.
5. If the server is not already stopped, stop the server now. Refresh the status of the servers until the status is **Stopped**.
6. Right-click **Directory** and select **Properties**.
7. Click the **Replication** tab.
8. Select **Use as a replica server**.
9. In the **Name used by master server for updates** field, select a name for the master server to use when it logs on to the replica server when it performs updates. This may be a distinguished name (DN) or a Kerberos user.

If you select a DN:

- Click the **Password** button next to the **Name used by master server for updates** field. Enter a password for the master server to use when it logs on to the replica server to perform updates.

Note: You should make note of this password and the name you entered in step 9. You will need them when you set up the master server for replication.

If you select **Add Kerberos User** :

- You will be prompted to enter the Kerberos name (in the format LDAP/*hostname*, where *hostname* is the fully qualified hostname of the master server) and the default realm (such as ACME.COM) of the master server.

Note: To use Kerberos, you must have Kerberos enabled on both the master and replica servers.

10. In the **Master server URL** field, enter the name of the master server in URL format. If your master server uses a port other than the default, enter this port number as part of the URL.
11. Click the **Database/Suffixes** tab. If the suffix that you want to replicate is not on the list, add it.
12. (optional) If you want to use Secure Sockets Layer (SSL) when replicating, use Digital Certificate Manager to enable SSL for the server. You can start Digital Certificate Manger from the **Network** tab. For additional information on enabling SSL on a directory server, see “Enable SSL on the LDAP directory server” on page 15.
13. Click **OK**.

Set up the master server to have a new replica

Follow these steps to set up the master server to have a new replica.

Note: You must have configured and started your master server before you perform this procedure.

1. In iSeries Navigator, expand the system on which the master directory server runs.
2. Expand **Network**.
3. Expand **Servers**.
4. Click **TCP/IP**.
5. Right-click **Directory** and select **Properties**.
6. If it is not already checked, check **Allow directory updates**.
7. Click **OK**.
8. Stop, then restart the LDAP directory server. Refresh the status of the servers until the status is **Started**.
9. Again, right-click **Directory** and select **Properties**.
10. Click the **Replication** tab. iSeries Navigator may prompt you to enter connection information. Enter this information, then click **OK**.
11. Click **Add**.
12. In the **Server** field, enter the name of the replica server in URL format.
13. Select your authentication method.

To use a distinguished name (DN) and password:

- a. Select **Use DN and password**.
- b. In the **Connect as** field, enter the name you specified in step 9 when you set up the replica server.

- c. Click **Password** and enter the password you specified in step 9 on page 23 when you set up the replica server.

To use Kerberos:

- Select **Use master servers Kerberos account**. The master server will use its Kerberos principal name to authenticate.

Note: To use Kerberos, you must have Kerberos enabled on both the master and replica servers.

14. If you want to use Secure Sockets Layer (SSL) when replicating, use Digital Certificate Manager to enable SSL for the server. You can start Digital Certificate Manager from the **Network** tab. For additional information on enabling SSL on a directory server, see “Enable SSL on the LDAP directory server” on page 15.
15. If the replica server does not use the default port, specify the port number in the **Port** field.
16. If you do not want to update the replica server every time an entry on the master server changes, select **Time**. Then specify how often you want the master server to update the replica.
17. Click **OK**.
18. Click the **Database/Suffixes** tab. If the suffix that you want to replicate is not on the list, add it.
19. Enable directory updates on each replica server:
 - a. In iSeries Navigator, expand the system on which the replica directory server runs.
 - b. Expand **Network**.
 - c. Expand **Servers**.
 - d. Click **TCP/IP**.
 - e. Right-click **Directory** and select **Properties**.
 - f. If **Allow directory updates** is unchecked, check it.
 - g. Click **OK**.
20. If each replica server is not already started, start it now.

Note: A server cannot be both a master server and a replica server.

Publish information to the directory server

You can configure your system to publish certain information into an LDAP directory server on the same system or on a different system. OS/400 automatically publishes this information to the LDAP directory server when you use iSeries Navigator to change this information on OS/400. Information that you can publish includes system (systems and printers), print shares, user information, and TCP/IP Quality of service policies. For more information on Quality of service, see LDAP configuration and QoS .

If the parent DN to which the data is being published does not exist, Directory Services automatically creates it. You may have also installed other OS/400 applications which publish information in an LDAP directory. Additionally, you can call application program interfaces (APIs) from your own programs to publish other types of information to the LDAP directory.

Notes:

1. When you configure OS/400 to publish the information type Users to the LDAP directory server, it automatically exports entries from the system distribution directory to the LDAP server. It uses the QGLDSSDD application program interface (API) to do this. This also keeps the LDAP directory synchronized with changes that are made in the system distribution directory. For information about the QGLDSSDD API, see the OS/400 Directory Services topic under Programming in the iSeries Information Center. The information available includes the following:
 - How to manually call this API.
 - How to prevent specific users from being exported to the LDAP server.
 - How it exports the system distribution directory fields.
2. When you configure OS/400 to publish the information type System to the LDAP directory server and select one or more printers to publish, the system will automatically keep the LDAP directory synchronized with changes that are made to those printers on the system. Printer information that can be published includes the printers location, its speed in pages per minute, whether it supports duplex

and color, its type and model, and its description. This information comes from the device description on the system being published. In a network environment, users can use this information to help select a printer.

3. You can also publish OS/400 information to an LDAP directory server that is not on an OS/400 if you configure that server to use the IBM schema.

To configure your system to publish OS/400 information into an LDAP directory server, take these steps:

1. In iSeries Navigator, right-click on your system and select **Properties**.
2. Click the **Directory Services** tab.
3. Click on the types of information that you want to publish.

Tip: If you plan to publish more than one type of information to the same location, you can save time by selecting multiple information types to configure at one time. Operations Navigator will then use the values you enter when you configure the one information type as default values when you configure subsequent information types.

4. Click **Details**.
5. Click the **Publish system information** check box.
6. Specify the **Authentication method** that you want the server to use, as well as the appropriate authentication information.
7. Click the **Edit** button next to the **(Active) Directory server** field. In the dialog that pops up, enter the name of the LDAP directory server where you want to publish OS/400 information, then click **OK**.
8. In the **Under DN** field, enter the parent distinguished name (DN) where you want information added on the directory server.
9. Fill in the fields in the **Server connection** frame that are appropriate to your configuration.

Note: To publish OS/400 information to the directory server using SSL or Kerberos, you need to first have your directory server configured to use the appropriate protocol. See “Use Kerberos authentication with the LDAP directory server” on page 37 for more information on SSL and Kerberos.

10. If your directory server does not use the default port, enter the correct port number in the **Port** field.
11. Click **Verify** to ensure that the parent DN exists on the server and that the connection information is correct. If the directory path does not exist, a dialog will prompt you to create it.

Note: If the parent DN does not exist, and you do not create it, then publishing will not be successful.

12. Click **OK**.

Note: You can also publish OS/400 information to an LDAP directory server that is on a different platform. You must publish user and system information to a directory server that uses a schema compatible with the Directory Services schema. The IBM SecureWay Directory schema definitions, which include iSeries Directory Services, can be found on the Directory Services web page.

You must publish print shares to a directory server which supports Microsofts Active Directory schema. Publishing print shares to an Active Directory allows users to configure iSeries printers directly from their Windows 2000 desktop with Windows 2000s Add Printer wizard. In order to do this in the Add Printer wizard, specify that you want to find a printer in the Windows 2000 Active Directory.

APIs for publishing OS/400 information to the directory server

Directory Services provides built-in support for publishing user and system information. These items are listed on the **Directory Services** page of the systems **Properties** dialog. You can use LDAP server configuration and publishing APIs to enable the OS/400 programs that you write to publish other types of information. These types of information then appear on the **Directory Services** page as well. Like users and systems, they are initially disabled, and you configure them using the same procedure. The program that adds the data to the LDAP directory is called the publishing agent. The type of information that is published, as it appears on the **Directory Services** page, is called the agent name.

The following APIs will allow you to incorporate publishing into your own programs:

QgldChgDirSvrA

An application uses the CSV0500 format to initially add an agent name that is marked as a disabled entry. Instructions for users of the application should instruct them to use iSeries Navigator to go to the Directory Services property page to configure the publishing agent. Examples of agent names are the systems and users agent names automatically available on the **Directory Services** page.

QgldLstDirSvrA

Use this APIs LSV0500 format to list what agents are currently available on your system.

QgldPubDirObj

Use this API to do the actual publishing of information.

For detailed information about these APIs, see the Lightweight Directory Access Protocol (LDAP) topic under Programming in the iSeries Information Center.

Specify a server for directory referrals

To assign referral servers for the directory server, take these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory**, then select **Properties**.
5. Click **Add**.
6. At the prompt, specify the name of the referral server in URL format. The following are examples of acceptable LDAP URLs:
 - ldap://test.server.com
 - ldap://test.server.com:400
 - ldap://9.9.99.255

Note: If the referral server does not use the default port, specify the correct port number as part of the URL, as port 400® is specified in the second example above.

7. Click **OK**.

Add suffixes to the LDAP directory server

Adding a suffix to the LDAP directory server allows the server to manage that part of the directory tree.

Note: You cannot add a suffix that is under another suffix already on the server. For example, if o=ibm, c=us were a suffix on your server, you cannot add ou=rochester, o=ibm, c=us.

To add a suffix to the directory server, take these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Properties**.
5. Click the **Database/Suffixes** tab.
6. In the **New suffix** field, type the name of the new suffix.
7. Click **Add**.
8. Click **OK**.

Note: Adding a suffix points the server to a section of the directory, but does not create any objects. If an object that corresponds to the new suffix did not previously exist, you must create it just as you would any other object.

Remove suffixes from the directory server

To remove a suffix from the LDAP directory server, take these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Properties**.
5. Click the **Database/Suffixes** tab.
6. Click on the suffix that you want to remove to select it.
7. Click **Remove**.

Note: You can choose to delete a suffix without deleting the directory objects under it. This makes the data inaccessible from the directory server. However, you can later regain access to the data by adding back the suffix.

Save and restore Directory Services information


Directory Services stores information in the following locations:

- The database library (QUSRDIRDB by default), which contains the directory servers contents.
- The QDIRSRV2 library, which is used to store publishing information.
- The QUSRSYS library, which stores various items in objects beginning with QGLD (specify QUSRSYS/QGLD* to save them).
- If you configure the directory server to log directory changes, a database library called QUSRDIRCL that the change log uses.

If the contents of the directory change regularly, you should save your database library and the objects in it on a regular basis. Configuration data is also stored in the following directory:

/QIBM/UserData/OS400/Dirsrv/

You should also save the files in that directory whenever you change the configuration or apply PTFs.

See Backup and Recovery, SC41-5304  for information on saving and restoring OS/400 data.

Manage ownership and access of directory data

Managing ownership and access of directory data includes the following tasks:

- “Work with the ownership properties of directory objects”
- “Work with access control lists (ACLs)” on page 28
- “Work with ACL Groups” on page 28

Work with the ownership properties of directory objects

To set the ownership properties of directory objects, take these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Authority**.

If you are not already connected to the directory server, the **Connect to Directory Server** dialog appears. Connect as the server administrator or as the owner of the object whose ownership properties you want to work with.

5. From the directory tree, select the object whose ownership properties you want to work with, then click **OK**.

Work with access control lists (ACLs)

Working with access control lists (ACLs) includes assigning explicit and implicit ACLs to directory objects, adding users to ACLs, removing users from ACLs, and browsing directory objects. Note that beginning with V5R1 Directory Services supports a new ACL model, so even if you have used ACLs before you may want to refamiliarize yourself with them.

To work with ACLs, take these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Authority**.

If you are not already connected to the directory server, the **Connect to Directory Server** dialog appears. Connect as the server administrator or as the owner of the object whose ACL you want to work with.

5. From the directory tree, select the object whose ACL you want to work with, then click **OK**.
6. Click the **ACL** tab.

Work with ACL Groups

To work with ACL groups, take these steps:

1. In iSeries Navigator, select **Network**.
2. Select **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **ACL Groups**.

Work with administrative access for authorized users

Beginning in V5R2, you can grant administrator access to user profiles that have been given access to the Directory Services Administrator (QIBM_DIRSRV_ADMIN) function identifier (ID).

For example, if the user profile JOHNSMITH is granted access to the Directory Services Administrator function ID and the Grant administrator access to authorized users option is selected from the Directory property dialog, the JOHNSMITH profile then has LDAP administrator authority. When this profile is used to bind to the directory server using the following DN, os400-profile=JOHNSMTH,cn=accounts,os400-sys=systemA.acme.com, the user has administrator authority. The system objects' suffix in this example is os400-sys=systemA.acme.com. For more information on projected users, see "Operating system projected backend" on page 38.

To select this option, take these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Right-click **Directory** and select **Properties**.
4. On the **General** tab under **Administrator information**, select the **Grant administrator access to authorized users** option.

To set the Directory Services Administrator authority function ID in a user profile, take these steps:

1. In iSeries Navigator, right-click the system name and select **Application Administration**.
2. Click the **Host Applications** tab.
3. Expand **Operating System/400®**.
4. Click **Directory Services Administrator** to highlight the option.
5. Click the **Customize** button.
6. Expand **Users**, **Groups**, or **Uses not in a group**, whichever is appropriate for the user you want.
7. Select a user or group to be added to the **Access allowed** list.

- | 8. Click the **Add** button.
- | 9. Click **OK** to save the changes.
- | 10. Click **OK** on the **Application Administration** dialog.

Track access and changes to the LDAP directory

- | You may want to track access and changes to your LDAP directory. You can use the LDAP directory's change log to keep track of changes to the directory. The change log is located under the special suffix `cn=changelog`. It is stored in the QUSRDIRCL library.

To enable the change log, follow these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Properties**.
5. Click the **Database/Suffixes** tab.
6. Select **Log directory changes**.
7. (optional) In the **Maximum entries** specify the maximum number of entries for the change log to keep.

Note: Though this parameter is optional, you should strongly consider specifying a number of maximum entries. If you do not specify a maximum number of entries, the change log will keep all entries and may become very large.

The `changeLogEntry` object class is used to represent the changes applied to the directory server. The set of changes is given by the ordered set of all entries within the changelog container as defined by `changeNumber`. The change log information is read-only.

Any user who is on the Access Control List for the `cn=changelog` suffix can search on the entries in the change log. You should only execute searches on the change log suffix, `cn=changelog`. Do not attempt to add, change, or delete to the change log suffix, even if you have authority to do so. This will cause unpredictable results.

Example:

The following example uses the `ldapsearch` command line utility to retrieve all change log entries logged on the server:

```
ldapsearch -h ldaphost -D cn=adminiatorator -w password -b cn=changelog (changetype=*)
```

Enable object auditing for the directory server

- | Directory Services supports OS/400 security auditing. If the QAUDCTL system value has `*OBJAUD` specified, you can enable object auditing through iSeries Navigator.

To enable object auditing for Directory Services, follow these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Properties**.
5. Click the **Auditing** tab.
6. Select the auditing setting that you want to use for your server.

Changes to auditing settings will take effect as soon as you click **OK**. There is no need to restart the LDAP directory server. For more information, see "Directory Services security" on page 36

Adjust performance of the LDAP directory server

You can adjust the performance of your LDAP directory server by changing any of the following:

- The size of searches
- The maximum time allowed for searches
- The servers transaction settings
- Number of database connections and server threads

To adjust the performance values of the directory server, take these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Properties**.
5. Click the **Performance** tab.

You can also adjust the performance of the directory server by changing the number of database connections and server threads that the server uses. To change this value, follow these steps:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Properties**.
5. Click the **Database/Suffixes** tab.

Chapter 5. Directory Services concepts and reference information

The following conceptual and reference information will help you to learn about and run your Directory Services LDAP server:

- “LDAP access control lists (ACLs)”
- “LDAP data interchange format” on page 32
- “National language support (NLS) considerations” on page 35
- “Ownership of LDAP directory objects” on page 35
- “LDAP directory referrals” on page 35
- “Transactions” on page 35
- “Replica LDAP directory servers” on page 36
- “Directory Services security” on page 36
- “Operating system projected backend” on page 38
- “Directory Services and OS/400 journaling support” on page 44

For information on LDAP basics and planning your LDAP server, also see Chapter 3, “Get started with Directory Services” on page 7.

LDAP access control lists (ACLs)

| In many cases, you probably would not want to restrict access to data on your LDAP directory server. For example, an LDAP server on your company Intranet might contain a telephone directory of company employees. You would probably want all employees to be able to view the data in this directory.

| However, the president of your company does not want all employees to be able to access her telephone number. In that case, you could create an **access control list (ACL)**. With this ACL, you could restrict access to her server entry to only those employees the president wanted to receive calls from.

With ACLs, you can control who has the authority to add and delete directory objects. You can also specify whether or not users have the ability to read, write, search, and compare directory attributes. ACLs can be either inherited or explicit. That is, you can use ACLs in one of the following ways:

- Explicitly set up an ACL for a specific object.
- Specify that objects inherit ACLs from objects higher up in the LDAP directory hierarchy.

| Perhaps the president in the previous example did not want all employees to be able to access her telephone number. She did, however, want all managers to be able to access it. In such a case, you could make use of an **ACL Group** to simplify granting authority to the managers. ACL groups allow you to grant access to specific groups of users rather than granting authority on an individual basis. This is particularly useful if the same group of people needs access to more than one set of objects. If the same managers that had access to the presidents telephone number, for example, later needed access to salary entries, you could reuse the ACL group.

ACL models

All versions of Directory Services support an access class level permissions model. Under this model, each LDAP attribute type has a classification of Normal, Sensitive, or Critical. The attribute schema files control these classifications. When you add a user to an objects ACL, you specify which classifications the user can read, write, search, and compare. In most schema, the telephone number would be classified as a Normal attribute. Therefore, to give the managers in the above example access to the presidents telephone number, you would give them read access to the Normal attributes in the presidents directory object. They would still not be able to access Sensitive and Critical information. All versions of Directory Services support setting access class level permissions.

| Directory Services also supports an attribute level permissions model. Under this model, you can specify
| read, write, search, and compare authorities for specific attributes, regardless of their access class.
| Consider again the above example. Under the attribute level permissions model, you could give the
| managers read access to the telephoneNumber attribute, even if they did not have access to Normal
| attributes in general.

| The attribute level permission model is compatible only with SecureWay Directory Services version 3.2
| and above servers. By default this is not enabled. You have the option of enabling it when you work with
| ACLs. After it is enabled, the model can be disabled only by reconfiguring the server and restoring the
| directory database. Before you decide to enable this model, be aware that you will not be able to
| administer it from any LDAP V2 client (including pre-V5R1 versions of iSeries Navigator) and that
| attempting to do so may corrupt ACL entries.



Special ACL values

Initially, all objects in the Directory Services directory server have an ACL that contains a special ACL group, CN=Anybody, that includes all directory users. By default this group has read, search, and compare access to normal-class attributes for all objects.

You may want some objects to have the same access permissions for all users who bind to the directory server with a connection that is not anonymous. To do this, use the special access control list (ACL) group cn=Authenticated.

To specify what access permissions an object has for itself, you can use the special DN cn=this. This enables child entries who inherit their ACLs to be automatically authorized to perform operations on their own objects.

Additional information

To administer ACLs through iSeries Navigator, you do not need to know the details of how Directory Services implements ACLs. However, if you want to specify ACL related attributes when using LDIF files or want to use ACLs with the LDAP command line utilities, you will need to familiarize yourself with the attributes that ACLs use. For information on ACL attributes, see the Access Control Lists reference document  of the The IBM SecureWay Directory Management Tool documentation .

For information on setting up and changing ACLs and ACL groups, follow these links:

“Work with access control lists (ACLs)” on page 28

“Work with ACL Groups” on page 28

LDAP data interchange format

The LDAP data interchange format (LDIF) provides you with a simple way to transfer directory information between LDAP directory servers. LDIF files hold LDAP directory entries in a simple text format. The format of LDIF files the directory server uses has changed slightly beginning with V4R5 of Directory Services. LDIF files consist of a sequence of lines that describe a directory entry or a set of changes to a directory entry. They cannot describe both.

The general format of an LDIF entry is:

```
version: 1
dn: distinguished name
attrtype1: attrvalue1
...
```

where:

- *version* shows the version of the LDIF file format. The version number must be 1. If the version number is absent, LDIF file is considered to be in an older LDIF file format. When the LDIF file is version 1, the content MUST be UTF-8 encoded.
- *distinguished name* is the distinguished name of the directory entry
- *attrtype1* is an LDAP attribute type (such as cn or ou)
- *attrvalue1* is value of the attribute

Each entry can have several attributes. Each attribute appears on a separate line. If an attribute value is longer than a single line, it may be continued on the next line, and is preceded by a space or tab character.

Blank lines separate multiple entries within the same LDIF file. Any line that begins with a pound-sign (#) is a comment line, and must be ignored when parsing an LDIF file.

Any distinguished name or attribute value that meets one of the following conditions should be base-64 encoded:

- It contains carriage returns or line feeds.
- It starts with a colon (:), SPACE, or less-than (<).
- It ends with space.

Base-64 encoded attributes are designated by using two colons between the attribute name and the value.

| External references are in the file:// URL format. There should be colon and less than (:<) characters
| between the attribute type and the external reference value.

Here are some examples of LDIF files:

Example 1: A simple LDAP file with two entries

```
version: 1
dn: cn=Barbara Jensen, ou=Rochester, o=Big Company, c=US
objectclass: top
objectclass: person
objectclass: organizationalPerson
cn: Barbara Jensen
cn: Barbara J Jensen
cn: Babs Jensen
sn: Jensen
uid: bjensen
telephonenumber: +1 408 555 1212
description: A big sailing fan.

dn: cn=Bjorn Jensen, ou=Rochester, o=Big Company, c=US
objectclass: top
objectclass: person
objectclass: organizationalPerson
cn: Bjorn Jensen
sn: Jensen
telephonenumber: +1 408 555 1212
description: Babs is a big sailing fan, and travels extensively in
search of perfect sailing conditions.
title: Product Manager, Rod and Reel Division
```

Example 2: A file containing a base-64-encoded value

```
version: 1
dn: cn=Gern Jensen, ou=Rochester, o=Big Company, c=US
objectclass: top
objectclass: person
objectclass: organizationalPerson
cn: Gern Jensen
cn: Gern 0 Jensen
sn: Jensen
```

National language support (NLS) considerations

Beginning with V4R5, both the OS/400 Directory Services LDAP server and the OS/400 LDAP client are based on LDAP Version 3. Be aware of the following NLS considerations:

- Data is transferred between LDAP servers and clients in UTF-8 format. All ISO 10646 characters are allowed.
- The Directory Services LDAP server uses the UTF-16 mapping method to store data in the database.
- The server and the client do case insensitive string comparisons. The uppercase algorithms will not be correct for all languages (locales).

For more information about UCS-2, see the Globalization topic under Planning in the iSeries Information Center.

Ownership of LDAP directory objects

| Each object in your LDAP directory has at least one owner. Object owners have the power to delete the
| object. Owners and the server administrator are the only users that can change the ownership properties
| and the access control list (ACL) attributes of an object. Ownership of objects can be either inherited or
| explicit. That is, to assign ownership you can do one of the following:

- | • Explicitly set up ownership for a specific object.
- | • Specify that objects inherit their owners from objects higher up in the LDAP directory hierarchy.

| Directory Services allows you to specify multiple owners for the same object. You can also specify that an
| object owns itself. To do this you include the special DN `cn=this` in the list of object owners. For example,
| assume that the object `cn=A` has the owner `cn=this`. Any user will have owner access to the `cn=A` object if
| he connects to the server as `cn=A`.

Related procedure:

“Work with the ownership properties of directory objects” on page 27

LDAP directory referrals

Referrals allow LDAP directory servers to work in teams. If the DN that a client requests is not in one directory, the server can automatically send (refer) the request to any other LDAP server.

Directory Services allows you to use two different types of referrals. You can specify default referral servers, where the LDAP server will refer clients whenever any DN is not in the directory. You can also use your LDAP client to add entries to the directory server that have the `objectClass referral`. This allows you to specify referrals that are based on what specific DN a client requests.

Note: With Directory Services, referral objects must contain only a distinguished name (`dn`), an `objectClass (objectClass)`, and a `referral (ref)` attribute. See “`ldapsearch utility`” on page 50 for an example that illustrates this restriction.

Referral servers are closely related to replica servers. Because data on replica servers cannot be changed from clients, the replica refers any requests to change directory data to the master server.

Transactions



| You can configure your systems LDAP directory server to allow clients to use transactions. A transaction is
| a group of LDAP directory operations that are treated as one unit. None of the individual LDAP operations
| that make up a transaction are permanent until all operations in the transaction have completed
| successfully and the transaction has been committed. If any of the operations fail or the transaction is
| cancelled, the other operations are undone. This capability can help users to keep LDAP operations
| organized. For example, a user might set up a transaction on his client that will delete several directory

l entries. If the client loses its connection to the server part way through the transaction, none of the entries
l are deleted. Therefore the user can simply start the transaction over rather than having to check to see
l which entries were successfully deleted.

The following LDAP operations may be part of a transaction:

- add
- modify
- modify RDN
- delete

Note: Do not include changes to the directory schema (the cn=schema suffix) in transactions. Though it is possible to include them, they cannot be backed out if the transaction fails. This could cause your directory server to experience unpredictable problems.

For additional information on transactions, see the Limited Transaction Support appendix  of the IBM SecureWay Directory Client SDK Programming Reference .

Replica LDAP directory servers

The information stored on replica LDAP directory servers is identical to the information on your main, or master, LDAP directory server. There are two principal benefits to having one or more replicas of your LDAP directory:

- Replicas make directory searches faster. Instead of having all clients direct search requests to a single master server, you can split requests between the master server and the replica servers.
- Replicas provide a backup to the master server. If the master server is unavailable, a replica can still fulfill search requests and provide access to directory data.

Replica servers are read-only. When an authorized user attempts to change an entry on a replica server, it refers the request to the master directory server.

Related procedure:

“Set up a new replica of the directory server” on page 20


Directory Services security

Security auditing

Beginning with V5R1, Directory Services supports OS/400 security auditing. Auditable items include the following:

- Binds to and unbinds from the directory server.
- Changes to permissions of LDAP directory objects.
- Changes in ownership of LDAP directory objects.
- Creation of, deletion of, searches of, and changes to LDAP directory objects.
- Changes to the password of administrator and update distinguished names (DNs)
- Changes to the passwords of users.
- File imports and exports.

You may need to make changes to your OS/400 auditing settings before auditing of directory entries will work. If the QAUDCTL system value has *OBJAUD specified, you can enable object auditing through

iSeries Navigator. For more information on auditing, see *Security - Reference*  or the topic Security auditing in the iSeries Information Center.

Connection authentication and security

Directory Services provides the following mechanisms that you can use to enhance the security of communications between LDAP clients and the LDAP directory server:

- Secure Sockets Layer (SSL) connections
- Kerberos authentication
- CRAM-MD5 password encryption

Use Secure Sockets Layer (SSL) and Translation Layer Security with the LDAP directory server

To make communications with your LDAP directory server more secure, Directory Services can use Secure Sockets Layer (SSL) security.

To use SSL with Directory Services, you must have one of the Cryptographic Access Provider products (5722-ACx) installed on your system. If you want to use SSL from iSeries Navigator, you must also have one of the Client Encryption products (5722-CEx) installed on your PC. You need this software if you want to do any of the following:

- To configure and administer Directory Services from your workstation using an SSL connection. This includes tasks that you perform from iSeries Navigator.
- To use an SSL connection with applications that you create with the Windows client application program interfaces (APIs).

| SSL is the standard for Internet security. You can use SSL to communicate with LDAP clients, as well as
| with replica LDAP servers. You can use client authentication in addition to server authentication to provide
| additional security to your SSL connections. Client authentication requires that the LDAP client present a
| digital certificate that confirms the clients identity to the server before a connection is established.

To use SSL, you must have Digital Certificate Manager (DCM), option 34 of OS/400, installed on your system. DCM provides an interface for you to create and manage digital certificates and certificate stores. See the documentation for Digital Certificate Manager for information on digital certificates and on using DCM. For information about SSL on iSeries, see *Securing applications with SSL*. For information about TLS on the iSeries server, see *Supported SSL and Transport Layer Security (TLS) protocols*.

Use Kerberos authentication with the LDAP directory server

| Directory Services allows you to set up the LDAP directory server to use Kerberos authentication.
| Kerberos is a network authentication protocol that uses secret key cryptography to provide strong
| authentication to client/server applications.

To enable Kerberos authentication, you must have one the Cryptographic Service Provider products (5722AC2 or 5722AC3) installed on your system. You must also have network authentication service configured.

The Kerberos support of Directory Services provides support for the GSSAPI SASL mechanism. This enables both SecureWay and Windows 2000 LDAP clients to use Kerberos authentication with the LDAP directory server.

The **Kerberos principal name** that the server uses has the following form:

```
service-name/host-name@realm
```

service-name is LDAP, host-name is the fully qualified TCP/IP name of the system, and realm is the default realm specified in the systems Kerberos configuration.

For example, for a system named `my-as400` in the `acme.com` TCP/IP domain, with a default Kerberos realm of `ACME.COM`, the LDAP server Kerberos principal name would be `LDAP/my-as400.acme.com@ACME.COM`. The default Kerberos realm is specified in the Kerberos configuration file (by default, `/QIBM/UserData/OS400/NetworkAuthentication/krb5.conf`) with the `default_realm` directive (`default_realm = ACME.COM`). By convention, Kerberos realm names use upper case, and host names use lower case. `LDAP/` must be upper case. The directory server cannot be configured to use Kerberos authentication if a default realm has not been configured.

When Kerberos authentication is used, the LDAP directory server associates a distinguished name (DN) with the connection that determines access to directory data. You can choose to have the server DN associated with one of the following methods:

- The server can create a DN based on the Kerberos ID. When you choose this option, a Kerberos identity of the form `principal@realm` generates a DN of the form `ibm-kn=principal@realm`. `ibm-kn=` is equivalent to `ibm-kerberosName=`.
- The server can search the directory for a distinguished name (DN) that contains an entry for the Kerberos principal and realm. When you choose this option, the server searches the directory for an entry that specifies this Kerberos identity as follows:
 - The server searches the directory for a `krbRealm-V2` object that has a `krbRealmName-V2` attribute that matches the Kerberos realm. If it finds such an entry, then it searches the DNs that are specified in the `princSubtree` attribute for an entry with a `krbPrincipalName` attribute that matches the principal name and realm name. If the DN configured in `krbAliasedObjectName` contains the DN of the entry previously found, then the DN configured in `krbAliasedObjectName` is used. Otherwise, the DN of the entry is used. This method is typically used when a Kerberos KDC is storing Kerberos principal information in the LDAP directory.
 - If the search described above fails, then the server searches for a directory entry that uses the `ibm-securityIdentities` auxiliary class and has an `altSecurityIdentities` attribute value of `KERBEROS:principal@realm`. This method can be used to associate Kerberos identities with directory entries when the KDC is not storing principals in the directory.

You must have a key table (keytab) file that contains a key for the LDAP service principal. See the Information Center topic `Network authentication service` under `Security` for more information about Kerberos on the iSeries server. The `Configuring network authentication service` section contains information on adding information to key table files.

Operating system projected backend

The system projected backend has the ability to map OS/400 objects as entries within the LDAP-accessible directory tree. The projected objects are LDAP representations of OS/400 objects instead of actual entries stored in the LDAP server database. With V5R2, OS/400 user profiles are the only objects being mapped or projected as entries within the directory tree. The mapping of user profile objects is referred to as the OS/400 user projected backend.

LDAP operations are mapped to the underlying OS/400 objects and LDAP operations perform operating system functions in order to access these objects. All LDAP operations performed on the user profiles are done under the authority of the user profile associated with the client connection.

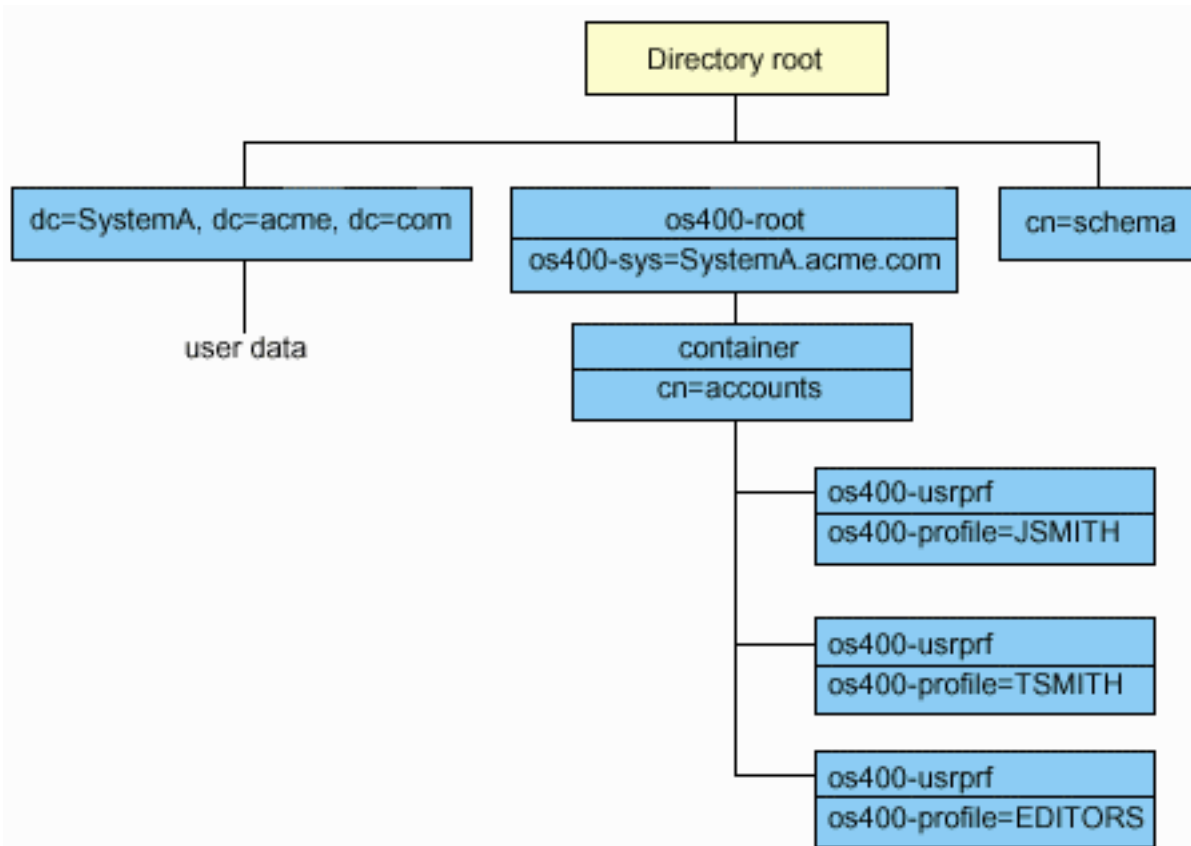
For more detailed information on the operating system projected backend, see the following:

- “OS/400 user projected directory information tree” on page 39
- “LDAP operations” on page 39
- “Administrator and replica bind DNs” on page 43
- “OS/400 user-projected schema” on page 43

OS/400 user projected directory information tree

The figure below shows a sample directory information tree (DIT) for the user projected backend. The figure shows both individual and group profiles. In the figure, JSMITH and TSMITH are user profiles, which is indicated internally by the group identifier (GID), GID=*NONE (or 0); EDITORS is a group profile, which is indicated internally by a non-zero GID.

The suffix `dc=SystemA,dc=acme,dc=com` is included in the figure for reference. This suffix represents the current database backend which is managing other LDAP entries. The suffix `cn=schema` is the current server-wide schema being used.



The root of the tree is a suffix, which defaults to `os400-sys=SystemA.acme.com`, where *SystemA.acme.com* is the name of your system. The objectclass is `os400-root`. Although the DIT cannot be modified or deleted, you may reconfigure the system objects' suffix. However, you must ensure that current suffix is not being used in ACLs or elsewhere on the system where wntries would need to be modified should the suffix be changed.

In the previous figure, the container, `cn=accounts`, is shown below the root. This object cannot be modified. A container is placed at this level in anticipation of other kinds of information or objects that may be projected by the operating system in the future. Below the `cn=accounts` container are the user profiles that are projected as `objectclass=os400-usrprf`. The user profiles are referred to as projected user profiles and are known to LDAP in the form `os400-profile=JSMITH,cn=accounts,os400-sys=SystemA.acme.com`.

LDAP operations

The following are the LDAP operations that can be performed using the projected user profiles.

Bind

An LDAP client can bind (authenticate) to the LDAP server using a projected user profile. This is accomplished by specifying the projected user profile distinguished name (DN) for the bind DN and the correct OS/400 user profile password for authentication. An example of a DN used in a bind request would be `os400-profile=jsmith,cn=accounts,os400-sys=systemA.acme.com`.

A client must bind as a projected user to access information in the system projected backend. The server performs all of the operations using the authority of that user profile. The projected user profile DN can also be used in LDAP ACLs like other LDAP entry DNs. The simple bind method is the only bind method that is allowed when a projected user profile is specified on a bind request.

Search

The system projected backend supports some basic search filters. You can specify the `objectclass`, `os400-profile`, and `os400-gid` attributes in search filters. The `os400-profile` attribute supports wildcards. The `os400-gid` attribute is limited to specifying `(os400-gid=0)`, which is an individual user profile, or `!(os400-gid=0)`, which is a group profile. You can retrieve all attributes of a user profile except the password and similar attributes.

For certain filters, only the DN objectclass and `os400-profile` values are returned. However, subsequent searches can be conducted to return more detailed information.

The following table describes the behavior of the system projected backend for search operations.

Table 1. System projected backend behavior for search operations

Search requested	Search base	Search scope	Search filter	Comments
Return information for <code>os400-sys=SystemA</code> , (optionally) for the containers under it, and (optionally) for the objects in those containers.	<code>os400-sys=SystemA.acme.com</code>	base, sub, or one	<code>objectclass=*</code> <code>objectclass=os400-root</code> <code>objectclass=container</code> <code>objectclass=os400-usrprf</code>	Return the appropriate attributes and their values based on the scope and filter specified. Hardcoded attributes and their values are returned for the system objects' suffix and the container under it.
Return all user profiles.	<code>cn=accounts,os400-sys=SystemA.acme.com</code>	one or sub	<code>os400-gid=0</code>	Only the distinguished name (DN), objectclass, and <code>os400-profile</code> values are returned for projected user profiles. If any other filter is specified, <code>LDAP_UNWILLING_TO_PERFORM</code> is returned.

Table 1. System projected backend behavior for search operations (continued)

Search requested	Search base	Search scope	Search filter	Comments
Return all group profiles.	cn=accounts, os400- sys=SystemA.acme.com	one or sub	(!(os400-gid=0))	Only the distinguished name (DN), objectclass, and os400-profile values are returned for projected user profiles. If any other filter is specified, LDAP_UNWILLING_TO_PERFORM is returned.
Return all user and group profiles.	cn=accounts, os400- sys=SystemA.acme.com	one or sub	os400-profile=*	Only the distinguished name (DN), objectclass, and os400-profile values are returned for projected user profiles. If any other filter is specified, LDAP_UNWILLING_TO_PERFORM is returned.
Return information for a specific user or group profile such as the user profile JSMITH.	cn=accounts, os400- sys=SystemA.acme.com	one or sub	os400-profile=JSMITH	Other attributes to be returned can be specified.
Return information for a specific user or group profile such as the user profile JSMITH.	os400- profile=JSMITH, cn=accounts, os400- sys=SystemA.acme.com	bas, sub, or one	objectclass=os400- usrprf objectclass=* os400-profile=JSMITH	Other attributes to be returned can be specified. Even though a scope of one level can be specified, the search results would return no values because there is nothing below the user profile JSMITH in the DIT.
Return all user and group profiles starting with A.	cn=accounts, os400- sys=SystemA.acme.com	one or sub	os400-profile=A*	Only the distinguished name (DN), objectclass, and os400-profile values are returned for projected user profiles. If any other filter is specified, LDAP_UNWILLING_TO_PERFORM is returned.

Table 1. System projected backend behavior for search operations (continued)

Search requested	Search base	Search scope	Search filter	Comments
Return all group profiles starting with G.	cn=accounts, os400- sys=SystemA.acme.com	one or sub	(&(!(os400-gid=0)) (os400-profile=G*))	Only the distinguished name (DN), objectclass, and os400-profile values are returned for projected user profiles. If any other filter is specified, LDAP_UNWILLING_TO_PERFORM is returned.
Return all user profiles starting with A.	cn=accounts, os400- sys=SystemA.acme.com	one or sub	(&(os400-gid=0) (os400-profile=A*))	Only the distinguished name (DN), objectclass, and os400-profile values are returned for projected user profiles. If any other filter is specified, LDAP_UNWILLING_TO_PERFORM is returned.

Compare

The LDAP compare operation can be used to compare an attribute value of a projected user profile. The os400-aut and os400-docpwd attributes cannot be compared.

Add and modify

You can create user profiles using the LDAP add operation and you can also modify user profiles using the LDAP modify operation.

Delete

User profiles can be deleted using the LDAP delete operation. To specify the behavior of the DLTUSRPRF OWNNOBJOPT and PGPOPT parameters, two LDAP server controls are now provided. These controls can be specified on the LDAP delete operation. Refer to the Delete User Profile (DLTUSRPRF) command for more information on the behavior of these parameters.

The following are the controls and their object identifiers (OIDs) that can be specified on the LDAP delete client operation.

- os400-dltusrprf-ownobjopt 1.3.18.0.2.10.8

The follow is the control value:

- controlValue ::= ownObjOpt [newOwner]
- ownObjOpt ::= *NODLT / *DLT / *CHGOWN

The ownObjOpt control value specifies the action to be taken if the user profile owns any objects. The value of *NODLT indicates not to delete the user profile if the user profile owns any objects. The *DLT value indicates to delete the owned objects and the *CHGOWN value indicates to transfer ownership to another profile.

| The newOwner value specifies the profile to which ownership is transferred. This value is required when
| ownObjOpt is set to *CHGOWN.

| Examples of the control value are the following:

- | – *NODLT: specifies that the profile cannot be deleted if it owns any objects
- | – *CHGOWN SMITH: specifies to transfer the ownership of any objects to the SMITH user profile.
- | • The object identifier (OID) is defined in ldap.h as LDAP_OS400_OWNOBJOPT_CONTROL_OID.
 - | – os400-dltusrprf-pgpopt 1.3.18.0.2.10.9

| The control value is defined as the following:

```
| controlValue ::=pgpOpt [ newPgp [ newPgpAut ] ]  
| pgpOpt ::= *NOCHG / *CHGPGP  
| newPgp ::= *NONE / user-profile-name  
| newPgpAut ::= *OLDPGP / *PRIVATE / *ALL / *CHANGE / *USE / *EXCLUDE
```

| The pgpOpt value specifies the action to be taken if the profile being deleted is the primary group for
| any objects. If *CHGPGP is specified, newPgp must also be specified. The newPgp value specifies
| the primary group profile name or *NONE. If a new primary group profile is specified, the newPgpAut
| value may also be specified. The newPgpAut value specifies the authority to the objects that the new
| primary group is given.

| Examples of the control value are the following:

- | – *NOCHG: specifies that the profile cannot be deleted if it is the primary group for any objects.
- | – *CHGPGP *NONE: specifies to remove the primary group for the objects.
- | – *CHGPGP SMITH *USE: specifies to change the primary group to the SMITH user profile and to
| grant *USE authority to the primary group.

| If either of these controls is not specified on the delete, the defaults currently in effect for the
| QSYS/DLTUSRPRF command are used instead.

| **ModRDN**

| You cannot rename projected user profiles because this is not supported by the operating system.

| **Import and Export APIs**

| The QgldImportLdif and QgldExportLdif APIs do not support importing or exporting data within the system
| projected backend.

| **Administrator and replica bind DN**

| You can specify a projected user profile as the configured administrator or replica bind DN. The password
| of the user profile is used. Projected user profiles can also become LDAP administrators if they are
| authorized to the Directory Server Administrator function identifier (QIBM_DIRSRV_ADMIN). Multiple user
| profiles can be granted administrator access.

| For more information, see “Work with administrative access for authorized users” on page 28.

| **OS/400 user-projected schema**

| The object classes and attributes from the projected backend can be found in the server-wide schema.
| The names of the LDAP attributes are in the format os400-*nnn*, where *nnn* is typically the keyword of the
| attribute (such as CRTUSRPRF or CHGUSRPRF) on the user profile commands. See “OS/400 user
| projected directory information tree” on page 39 for more information.

Directory Services and OS/400 journaling support

Directory Services uses OS/400 database support to store directory information. Directory Services uses commitment control to store directory entries in the database. This requires OS/400 journaling support.

When the server or the LDIF import tool is started for the first time, the following are built:

- A journal
- A journal receiver
- Any database tables needed initially

The journal QSQJRN is built in the database library that you have configured. The journal receiver QSQJRN0001 is initially created in the database library that you have configured.

Your environment, directory size and structure, or save and restore strategy may dictate some differences from the defaults, including how these objects are managed and the size threshold used. You can change journaling command parameters if necessary. LDAP journaling is set up by default to delete old receivers. If the change log is configured and you want to keep old receivers, execute the following command from an OS/400 command line:

```
JRN(QUSRDIRCL/QSQJRN) DLTRCV(*NO)
```

If the change log is configured, you can delete its journal receivers with the following command:

```
CHGJRN JRN(QUSRDIRCL/QSQJRN) DLTRCV(*YES)
```

For information on journaling commands, see the OS/400 commands topic under Programming in the iSeries Information Center.

Chapter 6. LDAP command line utilities

Directory Services includes five utilities that allow you to perform actions on the LDAP directory server from the Qshell command environment on OS/400. These utilities use the LDAP APIs. You can use these utilities from the qsh command line or call them from your programs. You may also find them useful as programming examples. When you install the Windows LDAP client that is included with Directory Services, you also install code that is very similar to the source code for the shell utilities.

The utilities are as follows:

- “Idapmodify and Idapadd utilities”, which add and modify LDAP directory entries.
- “Idapdelete utility” on page 48, which removes entries from the LDAP directory.
- “Idapsearch utility” on page 50, which searches the LDAP directory for entries.
- “Idapmodrdn utility” on page 54, which modifies the Relative Distinguished Name (RDN) of LDAP directory entries.

See “Notes about using SSL with the LDAP command line utilities” on page 56 for information on using SSL with the command line utilities.

Idapmodify and Idapadd utilities

The Idapmodify utility allows you to change entries or add entries to the LDAP directory server from the QSH command shell on your system. It uses the Idap_modify, Idap_add, and Idap_delete application program interfaces (APIs). The Idapadd utility works almost identically to the Idapmodify utility with the exception that the -a flag is turned on automatically.

Format:

Idapmodify [-a] [-V] [-b] [-c] [-r] [-M] [-n] [-v] [-F] [-R] [-C *charsef*] [-d *debuglevel*] [-D *binddn*] [-w *passwd*] [-m *mechanism*] [-O*hopcount*] [-h *ldaphost*] [-p *ldapport*] [-f *file*] [-Z] [-K *keyfile*] [-P *keyfilepw*] [-N *certificatename*]

Idapadd [-V] [-b] [-c] [-r] [-M] [-n] [-v] [-F] [-R] [-C *charsef*] [-d *debuglevel*] [-D *binddn*] [-w *passwd*] [-m *mechanism*] [-O*hopcount*] [-h *ldaphost*] [-p *ldapport*] [-f *file*] [-Z] [-K *keyfile*] [-P *keyfilepw*] [-N *certificatename*]

Note: If you do not provide entry information from *file* through the use of the -f option, the utility will wait to read entries from standard input. To break out of the wait, press the SysReq key, then choose 2. End previous request.

Diagnostics:

The exit status is 0 if no errors occur. Errors result in a non-zero exit status and a diagnostic message being written to standard error.

[Click here](#) to see examples of using these utilities.

Parameters:

-V	Specifies the LDAP version that the utility uses to bind to the LDAP server. By default, it uses an LDAP V3 connection. To explicitly select LDAP V3, specify -V 3. Specify -V 2 to run as an LDAP V2 application.
-a	Only Idapmodify uses this parameter. It indicates that the utility will add entries by default rather than modifying them. Using this parameter is the same as using Idapadd.

-b	Assume that any values that start with a <code>\</code> are binary values and that the actual value is in a file whose path is specified in the place where values normally appear.
-c	Continuous operation mode. Errors are reported, but <code>ldapmodify</code> or <code>ldapadd</code> continues with modifications or adds. The default is to exit after reporting an error.
-r	Replace existing values by default.
-M	Manage referral objects as regular entries.
-n	Show what would be done, but do not actually modify entries. Useful for debugging in conjunction with <code>-v</code> .
-v	Use verbose mode, with many diagnostics written to standard output.
-F	Force application of all changes regardless of the contents of input lines that begin with <code>replica:</code> (by default, <code>replica:</code> lines are compared against the LDAP server host and port in use to decide if a replication log record should actually be applied).
-R	Specifies that referrals are not to be automatically followed.
-C charset	Specifies that strings supplied as input to the utility are represented in a local character set (<i>charset</i>), and must be converted to UTF-8. Use the -C <i>charset</i> option if the input string codepage is different from the job codepage value. Refer to the documentation for the <code>ldap_set_iconv_local_charset()</code> API to see supported <i>charset</i> values.
-d debuglevel	Sets the debug level to <i>debuglevel</i> .
-D binddn	Use <i>binddn</i> to bind to the LDAP directory. <i>binddn</i> should be a string-represented DN.
-w passwd	Use <i>passwd</i> as the password for authentication.
-m mechanism	Use <i>mechanism</i> to specify the SASL mechanism that the client uses to bind to the server. The client uses the <code>ldap_sasl_bind_s()</code> API. Available mechanisms include CRAM-MD5 (encrypts password), EXTERNAL (used with SSL) and GSSAPI (Kerberos). The command ignores the -m parameter if -V 2 is set. If you do not specify -m , simple authentication is used.
-O hopcount	Specify <i>hopcount</i> to set the maximum number of hops that the client library will take when it chases referrals. The default hopcount is 10.
-h ldaphost	Specify an alternate host on which the LDAP server is running.
-p ldapport	Specify an alternate Transmission Control Protocol (TCP) port where the LDAP server is listening. The default LDAP port is 389. If not specified and -Z is specified, the default LDAP SSL port 636 is used.
-f file	Read the entry modification information from an LDIF file instead of from standard input. If an LDIF file is not specified, you must use standard input to specify the update records in LDIF format.
-Z	Use a secure SSL connection to communicate with the LDAP server. The -Z option is only supported by SSL-enabled versions of this tool.
-K keyfile	Specify the name of the SSL key database file. If the key database file is not in the current directory, specify the fully-qualified key database filename. If the utility cannot locate a key database, it will use a hard-coded set of default trusted certificate authority roots. The key database file typically contains one or more certificates of certification authorities (CAs) that are trusted by the client. These types of X.509 certificates are also known as trusted roots. This parameter effectively enables the -Z switch.
-P keyfilepw	Specify the key database password. This password is required to access the encrypted information in the key database file (including the private key). If a password stash file is associated with the key database file, the password is obtained from the stash file and this parameter is not required. This parameter is ignored if neither -Z nor -K are specified.

-N <i>certificatename</i>	Specify the label associated with the client certificate in the key database file. Note that if the LDAP server is configured to perform Server Authentication only, a client certificate is not required. If the LDAP server is configured to perform Client and Server Authentication, a client certificate is required. <i>certificatename</i> is not required if a default certificate/private key pair has been designated as the default. Similarly, <i>certificatename</i> is not required if there is a single certificate/private key pair in the designated key database file. This parameter is ignored if neither -Z nor -K are specified.
----------------------------------	--

Alternative Input Format:

The `ldapmodify` utility supports an alternative input format in order to maintain compatibility with older versions of the utility. This format consists of one or more entries that are separated by blank lines. Each entry has the following format:

```
Distinguished Name (DN)
attr=value
[attr=value ...]
```

where *attr* is the name of the attribute and *value* is the value. By default, values are added. If you give the **-r** command line flag, the default is to replace existing values with the new one. Note that it is permissible for a given attribute to appear more than once (for example, you can add more than one value for an attribute). Also note that you can use a trailing backslash (`\`) to continue values across lines and to preserve new lines in the value itself. To remove a value, precede the *attr* value with a dash (`-`). The equal sign (`=`) and value should be omitted to remove an entire attribute. *attr* should be preceded by a plus sign (`+`) to add a value in the presence of the **-r** flag.

Examples: ldapmodify and ldapadd

Example 1:

If the file `/tmp/entrymods` exists and has the following contents:

```
dn: cn=Modify Me, o=University of Higher Learning, c=US
changetype: modify
replace: mail
mail: modme@student.of.life.edu
-
add: title
title: Grand Poobah
-
add: jpegPhoto
jpegPhoto:< file:///tmp/modme.jpeg
-
delete: description
-
```

The command `ldapmodify -b -r -f /tmp/entrymods` will do the following:

- Replace the contents of the Modify Me entry's mail attribute with the value `modme@student.of.life.edu`.
- Add a title of Grand Poobah.
- Add the contents of the file `/tmp/modme.jpeg` as a jpegPhoto.
- Completely remove the description attribute.

You can perform the same modifications as above with the older `ldapmodify` input format:

```
cn=Modify Me, o=University of Higher Learning, c=US
mail=modme@student.of.life.edu
+title=Grand Poobah
+jpegPhoto=/tmp/modme.jpeg
-description
```

The command for using the old format would be:

```
ldapmodify -b -r -f /tmp/entrymods
```

Example 2:

Assume that the file **/tmp/newentry** exists and has the following contents:

```
dn: cn=John Doe, o=University of Higher Learning, c=US
objectClass: person
cn: John Doe
cn: Johnny
sn: Doe
title: Manager
mail: johndoe@student.of.life.edu
uid: jdoe
```

The command `ldapadd -f /tmp/entrymods` will add a new entry for John Doe, using the values from the file `/tmp/newentry`.

Example 3:

If the file **/tmp/newentry** exists and has the contents:

```
dn: cn=John Doe, o=University of Higher Learning, c=US
changetype: delete
```

The command `ldapmodify -f /tmp/entrymods` will remove the entry for John Doe.

Idapdelete utility

The `Idapdelete` utility allows you to delete one or more entries from an LDAP directory server. It runs through the QSH command shell on OS/400. It uses the `ldap_delete` application program interface (API).

Format:

```
Idapdelete [-V] [-M] [-n] [-v] [-c] [-R] [-C charset] [-d debuglevel] [-f file] [-D binddn] [-w passwd] [-m mechanism] [-O hopcount] [-h ldaphost] [-p ldapport] [-Z] [-K keyfile] [-P keyfilepw] [-N certificatename] [dn]...
```

Note: If you do not provide *dn* arguments, the `Idapdelete` command will wait to read a list of DNs from standard input. To break out of the wait, press the SysReq key, then choose 2. End previous request.

Diagnostics:

The exit status is 0 if no errors occur. Errors result in a non-zero exit status and a diagnostic message being written to standard error.

[Click here to see examples of using the `Idapdelete` utility.](#)

Parameters:

-V	Specifies the LDAP version that the utility uses to bind to the LDAP server. By default, it uses an LDAP V3 connection. To explicitly select LDAP V3, specify <code>-V 3</code> . Specify <code>-V 2</code> to run as an LDAP V2 application.
-M	Manage referral objects as regular entries.
-n	Show what would be done, but do not actually delete entries. Useful for debugging in conjunction with <code>-v</code> .
-v	Use verbose mode, with many diagnostics written to standard output.

-c	Continuous operation mode. Errors are reported, but <code>ldapdelete</code> will continue with deletions. The default is to exit after reporting an error.
-R	Specifies that referrals are not to be automatically followed.
-C charset	Specifies that the distinguished names (DNs) supplied as input to the <code>ldapdelete</code> utility are represented in a local character set (<i>charset</i>). Use -C charset to override the default, where strings must be supplied in UTF-8. Use the -C charset option if the input string codepage is different from the job codepage value. Refer to the documentation for the <code>ldap_set_iconv_local_charset()</code> API to see supported <i>charset</i> values.
-d debuglevel	Sets the debug level to <i>debuglevel</i> .
-f file	Read a series of lines from <i>file</i> , performing one LDAP delete for each line in the file. Each line in the file should contain a single distinguished name (DN).
-D binddn	Use <i>binddn</i> to bind to the LDAP directory. <i>binddn</i> should be a string-represented DN.
-w passwd	Use <i>passwd</i> as the password for authentication.
-m mechanism	Use <i>mechanism</i> specify the SASL mechanism to be used to bind to the server. The <code>ldap_sasl_bind_s()</code> API will be used. Available mechanisms include CRAM-MD5 (encrypts password), EXTERNAL (used with SSL) and GSSAPI (Kerberos). The -m parameter is ignored if -V 2 is set. If -m is not specified, simple authentication is used.
-O hopcount	Specify <i>hopcount</i> to set the maximum number of hops that the client library will take when chasing referrals. The default hopcount is 10.
-h ldaphost	Specify an alternate host on which the LDAP server is running.
-p ldapport	Specify an alternate Transmission Control Protocol (TCP) port where the LDAP server is listening. The default LDAP port is 389. If not specified and -Z is specified, the default LDAP SSL port 636 is used.
-Z	Use a secure SSL connection to communicate with the LDAP server. The -Z option is only supported by SSL-enabled versions of this tool.
-K keyfile	Specify the name of the SSL key database file. If the key database file is not in the current directory, specify the fully-qualified key database filename. If the utility cannot locate a key database, it will use a hard-coded set of default trusted certificate authority roots. The key database file typically contains one or more certificates of certification authorities (CAs) that are trusted by the client. These types of X.509 certificates are also known as trusted roots. This parameter effectively enables the -Z switch.
-P keyfilepw	Specify the key database password. This password is required to access the encrypted information in the key database file (including the private key). If a password stash file is associated with the key database file, the password is obtained from the stash file and this parameter is not required. This parameter is ignored if neither -Z nor -K are specified.
-N certificatename	Specify the label associated with the client certificate in the key database file. Note that if the LDAP server is configured to perform Server Authentication only, a client certificate is not required. If the LDAP server is configured to perform Client and Server Authentication, a client certificate is required. <i>certificatename</i> is not required if a default certificate/private key pair has been designated as the default. Similarly, <i>certificatename</i> is not required if there is a single certificate/private key pair in the designated key database file. This parameter is ignored if neither -Z nor -K are specified.
<i>dn</i>	Specifies one or more <i>dn</i> arguments. Each <i>dn</i> should be a string-represented DN.

Example: ldapdelete

The following command will attempt to delete the entry named with commonName Delete Me directly below the University of Life organizational entry :

```
ldapdelete cn=Delete Me, o=University of Life, c=US
```

It may be necessary to supply a *binddn* and *passwd* (see the **-D** and **-w** options).

ldapsearch utility

The ldapsearch utility allows you to search for an entry on your LDAP directory server from the QSH command shell on OS/400. It uses the ldap_search application programming interface (API).

The search uses a filter that conforms to the string representation for LDAP filters. For more information on LDAP search filters, see the ldap_search API information in the OS/400 Directory Services topic under Programming in the iSeries Information Center.

If the ldapsearch utility finds one or more entries, it retrieves the attributes that are specified by *attrs* and prints the entries and values to standard output. If you do not list any attributes, it returns all attributes.

Format:

ldapsearch [-V] [-n] [-v] [-t] [-A] [-B] [-L] [-M] [-R] [-C *charsef*] [-d *debuglevel*] [-F *sep*] [-f *file*] [-D *binddn*] [-w *bindpasswd*] [-m *mechanism*] [-O *hopcountf*] [-h *ldaphostf*] [-p *ldapporf*] [-Z] [-K *keyfile*] [-P *keyfilepw*] [-N *certificatename*] [-b *searchbase*] [-s *scope*] [-a *deref*] [-l *time limit*] [-z *size limit*] *filter* [*attrs...*]

Diagnostics:

The exit status is 0 if no errors occur. Errors result in a non-zero exit status and a diagnostic message being written to standard error.

Output Format:

If ldapsearch finds one or more entries, it writes each entry to standard output in the form:

```
Distinguished Name (DN)
attributename=value
attributename=value
attributename=value
...
```

Multiple entries are separated with a single blank line. If you use the -F option to specify a separator character, the output displays that character instead of the equal (=) character. If you use the -t option, the name of a temporary file replaces the actual value. If you specify the -A option, only the attributename part is written.

[Click here to see examples of using the ldapsearch utility.](#)

Parameters:

-V	Specifies the LDAP version that the utility uses to bind to the LDAP server. By default, it uses an LDAP V3 connection. To explicitly select LDAP V3, specify -V 3. Specify -V 2 to run as an LDAP V2 application.
-n	Show what would be done, but do not actually perform the search. Useful for debugging in conjunction with -v.
-v	Use verbose mode, with many diagnostics written to standard output.
-t	Write retrieved values to a set of temporary files. This is useful for dealing with binary values such as jpegPhoto or audio.
-A	Retrieve attributes only (no values). This is useful when you just want to see if an attribute is present in an entry and are not interested in the specific values.
-B	Do not suppress display of binary values. This is useful when dealing with values that appear in alternate character sets such as ISO-8859.1. This option is implied by -L.

-L	Display search results in LDIF format. This option also turns on the -B option, and causes the -F option to be ignored.
-M	Manage referral objects as regular entries.
-R	Specifies that referrals are not to be automatically followed.
-C charset	Specifies that strings supplied as input to the <code>ldapsearch</code> utility are represented in a local character set (<i>charset</i>). String input includes the filter, the bind DN, and the base DN. Similarly, when displaying data, <code>ldapsearch</code> will convert data received from the LDAP server to the specified characters. Use the -C <i>charset</i> option if the input string codepage is different from the job codepage value. Refer to the documentation for the <code>ldap_set_iconv_local_charset()</code> API to see supported <i>charset</i> values. Also, if the -C option and the -L option are both specified, input is assumed to be in the specified character set, but output from <code>ldapsearch</code> is always preserved in its UTF-8 representation, or a base 64-encoded representation of the data when non-printable characters are detected. This is the case since standard LDIF files only contain UTF-8 (or base 64-encoded UTF-8) representations of string data.
-d debuglevel	Sets the debug level to <i>debuglevel</i> .
-F sep	Use <i>sep</i> as the field separator between attribute names and values. The default separator is <code>`=</code> , unless the -L flag has been specified, in which case this option is ignored.
-f file	Read a series of lines from file, performing one LDAP search for each line in the file. Each line in the file should contain a single distinguished name (DN).
-D binddn	Use <i>binddn</i> to bind to the LDAP directory. <i>binddn</i> should be a string-represented DN.
-w passwd	Use <i>passwd</i> as the password for authentication.
-m mechanism	Use <i>mechanism</i> to specify the SASL mechanism to be used to bind to the server. The <code>ldap_sasl_bind_s()</code> API will be used. Available mechanisms include CRAM-MD5 (encrypts password), EXTERNAL (used with SSL) and GSSAPI (Kerberos). The -m parameter is ignored if -V 2 is set. If -m is not specified, simple authentication is used.
-O hopcount	Specify <i>hopcount</i> to set the maximum number of hops that the client library will take when chasing referrals. The default hopcount is 10.
-h ldaphost	Specify an alternate host on which the LDAP server is running.
-p ldapport	Specify an alternate Transmission Control Protocol (TCP) port where the LDAP server is listening. The default LDAP port is 389. If not specified and -Z is specified, the default LDAP Secure Sockets Layer (SSL) port 636 is used.
-Z	Use a secure SSL connection to communicate with the LDAP server. The -Z option is only supported by SSL-enabled versions of this tool.
-K keyfile	Specify the name of the SSL key database file. If the key database file is not in the current directory, specify the fully-qualified key database filename. If the utility cannot locate a key database, it will use a hard-coded set of default trusted certificate authority roots. The key database file typically contains one or more certificates of certification authorities (CAs) that are trusted by the client. These types of X.509 certificates are also known as trusted roots. This parameter effectively enables the -Z switch.
-P keyfilepw	Specify the key database password. This password is required to access the encrypted information in the key database file (including the private key). If a password stash file is associated with the key database file, the password is obtained from the stash file and this parameter is not required. This parameter is ignored if neither -Z nor -K are specified.

-N <i>certificatename</i>	Specify the label associated with the client certificate in the key database file. Note that if the LDAP server is configured to perform Server Authentication only, a client certificate is not required. If the LDAP server is configured to perform Client and Server Authentication, a client certificate is required. <i>certificatename</i> is not required if a default certificate/private key pair has been designated as the default. Similarly, <i>certificatename</i> is not required if there is a single certificate/private key pair in the designated key database file. This parameter is ignored if neither -Z nor -K are specified.
-b <i>searchbase</i>	Use <i>searchbase</i> as the starting point for the search instead of the default. If -b is not specified, this utility will examine the LDAP_BASEDN environment variable for a <i>searchbase</i> definition.
-s <i>scope</i>	Specify the scope of the search. <i>scope</i> should be one of base, one, or sub to specify a base object, one-level, or subtree search. The default is sub.
-a <i>deref</i>	Specify how aliases dereferencing is done. <i>deref</i> should be one of never, always, search, or find to specify that aliases are never dereferenced, always dereferenced, dereferenced when searching, or dereferenced only when locating the base object for the search. The default is to never dereference aliases.
-l <i>timelimit</i>	Wait at most <i>timelimit</i> seconds for a search to complete.
-z <i>sizelimit</i>	Limit the results of the search to at most <i>sizelimit</i> entries. This makes it possible to place an upper bound on the number of entries that are returned for a search operation.
<i>filter</i>	Specifies the name of the filter that the search uses.
<i>attrs...</i>	Specifies the attributes that the utility retrieves if the search finds one or more entries. If you do not list any values for <i>attrs</i> , the utility returns all attributes.

Examples: ldapsearch

Example 1:

The command `ldapsearch cn=john doe cn telephoneNumber` performs a subtree search (using the default search base) for entries with a `commonName` of `john doe`. The search retrieves the `commonName` values and the `telephoneNumber` values, and prints them to standard output. If the search finds two entries, the output looks similar to this:

```
cn=John E Doe, ou=College of Literature, Science, and the Arts,
ou=Students, ou=People, o=University of Higher Learning, c=US
cn=John Doe
cn=John Edward Doe
cn=John E Doe 1
cn=John E Doe
telephoneNumber=+1 313 555-5432
```

```
cn=John B Doe, ou=Information Technology Division,
ou=Faculty and Staff,
ou=People, o=University of Higher Learning, c=US
cn=John Doe
cn=John B Doe 1
cn=John B Doe
telephoneNumber=+1 313 555-1111
```

Example 2:

The command `ldapsearch -t uid=jed jpegPhoto audio` performs a subtree search using the default search base for entries with user id of `jed`. The search retrieves the `jpegPhoto` and `audio` values and writes them to temporary files. If the search finds one entry with one value for each of the requested attributes, the output looks similar to this:

```
cn=John E Doe,  
ou=Information Technology Division,  
ou=Faculty and Staff,  
ou=People, o=University of Higher Learning, c=US  
audio=/tmp/ldapsearch-audio-a19924  
jpegPhoto=/tmp/ldapsearch-jpegPhoto-a19924
```

Example 3:

The command `ldapsearch -L -s one -b c=US o=university* o description` performs a one-level search at the `c=US` level. This search looks for all organizations whose `organizationName` begins with `university`. The search displays its results in the LDIF format. It retrieves the `organizationName` attribute value and the `description` attribute values and prints them to standard output that looks similar to this:

```
dn: o=University of Alaska Fairbanks, c=US  
o: University of Alaska Fairbanks  
description: Preparing Alaska for a brave new tomorrow  
description: leaf node only  
  
dn: o=University of Colorado at Boulder, c=US  
o: University of Colorado at Boulder  
description: No personnel information  
description: Institution of education and research  
  
dn: o=University of Colorado at Denver, c=US  
o: University of Colorado at Denver  
o: UCD  
o: CU/Denver  
o: CU-Denver  
description: Institute for Higher Learning and Research  
  
dn: o=University of Florida, c=US  
o: University of Florida  
o: UF1  
description: Shaper of young minds  
...
```

Example 4:

As discussed in “LDAP directory referrals” on page 35, Directory Services LDAP directories may contain referral objects, provided that they contain only the following:

- A distinguished name (`dn`).
- An `objectClass` (`objectClass`).
- A referral (`ref`) attribute.

This example demonstrates searches where a referral object is involved.

Assume that `System_A` holds the referral entry:

```
dn: cn=Barb Jensen, ou=Rochester, o=Big Company, c=US  
ref: ldap://System_B:389/cn=Barb Jensen,  
    ou=Rochester, o=Big Company, c=US  
objectclass: referral
```

All attributes associated with the entry should reside on `System_B`.

`System_B` contains an entry:

```
dn: cn=Barb Jensen, ou=Rochester, o=Big Company, c=US  
cn: Barb Jensen  
objectclass: organizationalPerson  
sn: Jensen  
telephonenumber: (800) 555 1212
```

When a client issues a request to System_A and does not send the manageDsaIT control, then the server returns a referral. For example, by using -M on ldapsearch the LDAP server on System_A responds to the client with the following URL:

```
ldap://System_B:389/cn=Barb Jensen,  
ou=Rochester, o=Big Company, c=US
```

The client uses this information to issue a request to System_B. If the entry on System_A contains attributes in addition to dn, objectclass, and ref, the server ignores those attributes.

When the client receives a referral response from a server, it issues the request again, this time to the server to which the returned URL refers. If the search was done with onelevel scope, the referral request uses base scope. The results of this search vary depending on the value you specify for the scope of the search (-b).

If you specify -s sub, as shown here:

```
ldapsearch -h System_A -b ou=Rochester, o=Big Company, c=US  
-s sub sn=Jensen
```

the search returns all attributes for all entries with sn=Jensen that reside in or below ou=Rochester, o=Big Company, c=US on both System_A and System_B. The client receives a referral from System_A and searches System_B, returning cn=Barb Jensen,ou=Rochester,o=Big Company,c=US.

If you specify -s one, as shown here:

```
ldapsearch -h System_A -b ou=Rochester, o=Big Company, c=US  
-s one sn=Jensen
```

the search returns no entries on either system. Instead, the server returns the referral URL to the client:

```
ldap://System_B:389/cn=Barb Jensen,  
ou=Rochester, o=Big Company, c=US??base
```

The client in turn submits a request:

```
ldapsearch -h System_B -b cn=Barb Jensen, ou=Rochester, o=Big Company, c=US  
-s base sn=Jensen
```

This returns the entry cn=Barb Jensen,ou=Rochester,o=Big Company,c=US.

Idapmodrdn utility

The Idapmodrdn utility allows you to change the Relative Distinguished Name (RDN) of entries on the LDAP directory server. You use it from the QSH command shell on OS/400. It uses the ldap_modrdn application program interface (API).

Format:

```
Idapmodrdn [-V] [-r] [-M] [-n] [-v] [-c] [-R] [-C charset] [-d debuglevel] [-D binddn] [-w passwd] [-m  
mechanism] [-O hopcount] [-h ldaphost] [-p ldapport] [-Z] [-K keyfile] [-P keyfilepw] [-N certificatename]  
[-f file] [dn rdn]
```

Notes:

1. If you give the command-line arguments *dn* and *rdn*, *rdn* will replace the RDN of the entry that is specified by the DN, *dn*. Otherwise, the contents of the file (or of the standard input if you do not give the -f flag) should consist of one or more entries.

Distinguished Name (DN)

Relative Distinguished Name (RDN)

One or more blank lines separate each DN/RDN pair.

- If you do not supply entry information from *file* through the use of the **-f** option (or from the command-line pair *dn* and *rdn*), the `ldapmodrdn` command will wait to read entries from standard input. To break out of the wait, press the SysReq key, then choose 2. End previous request.

Diagnostics:

The exit status is 0 if no errors occur. Errors result in a non-zero exit status and a diagnostic message being written to standard error.

[Click here to see an example of using the `ldapmodrdn` utility.](#)

Parameters:

-V	Specifies the LDAP version that the utility uses to bind to the LDAP server. By default, it uses an LDAP V3 connection. To explicitly select LDAP V3, specify -V 3 . Specify -V 2 to run as an LDAP V2 application.
-r	Remove old relative distinguished name (RDN) values from the entry. Default is to keep old values.
-M	Manage referral objects as regular entries.
-n	Show what would be done, but do not actually change entries. Useful for debugging in conjunction with -v .
-v	Use verbose mode, with many diagnostics written to standard output.
-c	Continuous operation mode. Errors are reported, but <code>ldapmodrdn</code> will continue with modifies. The default is to exit after reporting an error.
-R	Specifies that referrals are not to be automatically followed.
-C charset	Specifies that strings supplied as input to the utility are represented in a local character set (<i>charset</i>), and must be converted to UTF-8. Use the -C charset option if the input string codepage is different from the job codepage value. Refer to the documentation for the <code>ldap_set_iconv_local_charset()</code> API to see supported <i>charset</i> values.
-d debuglevel	Sets the debug level to <i>debuglevel</i> .
-D binddn	Use <i>binddn</i> to bind to the LDAP directory. <i>binddn</i> should be a string-represented DN.
-w passwd	Use <i>passwd</i> as the password for authentication.
-m mechanism	Use <i>mechanism</i> specify the SASL mechanism to be used to bind to the server. The <code>ldap_sasl_bind_s()</code> API will be used. Available mechanisms include CRAM-MD5 (encrypts password), EXTERNAL (used with SSL) and GSSAPI (Kerberos). The -m parameter is ignored if -V 2 is set. If -m is not specified, simple authentication is used.
-O hopcount	Specify <i>hopcount</i> to set the maximum number of hops that the client library will take when chasing referrals. The default hopcount is 10.
-h ldaphost	Specify an alternate host on which the LDAP server is running.
-p ldapport	Specify an alternate Transmission Control Protocol (TCP) port where the LDAP server is listening. The default LDAP port is 389. If not specified and -Z is specified, the default LDAP SSL port 636 is used.
-Z	Use a secure SSL connection to communicate with the LDAP server. The -Z option is only supported by SSL-enabled versions of this tool.
-K keyfile	Specify the name of the SSL key database file. If the key database file is not in the current directory, specify the fully-qualified key database filename. If the utility cannot locate a key database, it will use a hard-coded set of default trusted certificate authority roots. The key database file typically contains one or more certificates of certification authorities (CAs) that are trusted by the client. These types of X.509 certificates are also known as trusted roots. This parameter effectively enables the -Z switch.

-P <i>keyfilepw</i>	Specify the key database password. This password is required to access the encrypted information in the key database file (including the private key). If a password stash file is associated with the key database file, the password is obtained from the stash file and this parameter is not required. This parameter is ignored if neither -Z nor -K are specified.
-N <i>certificatename</i>	Specify the label associated with the client certificate in the key database file. Note that if the LDAP server is configured to perform Server Authentication only, a client certificate is not required. If the LDAP server is configured to perform Client and Server Authentication, a client certificate is required. <i>certificatename</i> is not required if a default certificate/private key pair has been designated as the default. Similarly, <i>certificatename</i> is not required if there is a single certificate/private key pair in the designated key database file. This parameter is ignored if neither -Z nor -K are specified.
-f <i>file</i>	Read the entry modification information from an LDIF file instead of from standard input or the command-line (by specifying <i>dn</i> and the new <i>rdn</i>). Standard input can also be supplied from a file (< file).
<i>dn rdn</i>	Specify the distinguished name of an entry to rename and the new relative distinguished name for the entry.

Example: ldapmodrdn

Assume that you have already created the text file **/tmp/entrymods** and that it has the following contents:

```
cn=Modify Me, o=University of Life, c=US
cn=The New Me
```

The following command:

```
ldapmodrdn -r -f /tmp/entrymods
```

will change the RDN of the Modify Me entry from Modify Me to The New Me. The old cn, Modify Me will be removed.

Notes about using SSL with the LDAP command line utilities

To use the Secure Sockets Layer (SSL) features of the command line utilities, you must have installed one of the Cryptographic Access Provider Products (5722-ACx).

“Use Secure Sockets Layer (SSL) and Translation Layer Security with the LDAP directory server” on page 37 discusses using SSL with the Directory Services LDAP server. This information includes managing and creating trusted Certificate Authorities with Digital Certificate Manager.

Some of the LDAP servers accessed by the client use server authentication only. For these servers, you only need to define one or more trusted root certificates in the certificate store. With server authentication, the client can be assured that the target LDAP server has been issued a certificate by one of the trusted Certificate Authorities (CAs). In addition, all LDAP transactions that flow over the SSL connection with the server are encrypted. This includes the LDAP credentials that are supplied on application program interfaces (APIs) that are used to bind to the directory server. For example, if the LDAP server is using a high-assurance Verisign certificate, you should do the following:

1. Obtain a CA certificate from Verisign.
2. Use DCM to import it into your certificate store.
3. Use DCM to mark it as trusted.

If the LDAP server is using a privately issued server certificate, the servers administrator can supply you with a copy of the servers certificate request file. Import the certificate request file into your certificate store and mark it as trusted.


If you use the shell utilities to access LDAP servers that use both client authentication and server authentication, you must do the following:

- Define one or more trusted root certificates in the system certificate store. This allows the client to be assured that the target LDAP server has been issued a certificate by one of the trusted CAs. In addition, all LDAP transactions that flow over the SSL connection with the server are encrypted. This includes the LDAP credentials that are supplied on application program interfaces (APIs) that are used to bind to the directory server.
- Create a key pair and request a client certificate from a CA. After receiving the signed certificate from the CA, receive the certificate into the key ring file on the client.

Chapter 7. Troubleshoot Directory Services

Unfortunately, even reliable servers such as the Directory Services LDAP server sometimes have problems. When your LDAP directory server has problems, the following information can help you figure out what is wrong and how to fix the trouble.

- “Basic troubleshooting procedure for Directory Services”
- “Common LDAP client errors” on page 61

For additional information about common Directory Services problems, see the Directory Services home page  at the following URL:

<http://www.iseries.ibm.com/ldap>

Basic troubleshooting procedure for Directory Services

You can find return codes for LDAP errors in the ldap.h file, which is located on your system in QSYSINC/H.LDAP.

When you get an error on your LDAP directory server and want more details, another action to take is to view the QDIRSRV job log. For reproduceable errors, you can use Trace TCP/IP Application (TRCTCPAPP APP(*DIRSRV)) command to run a trace of the errors. See “Use TRCTCPAPP to help find problems” on page 60 for more information.

Directory Services uses several Structured Query Language (SQL) servers. When an SQL error occurs, the QDIRSRV job log will usually contain the following message:

```
SQL error -1 occurred
```

In these instances the QDIRSRV job log will refer you to the SQL server job logs. However, in some cases QDIRSRV may not contain this message and this referral, even if an SQL server is the cause of the problem. In these instances, it will help you to know what SQL servers should be started, and what Directory Services uses them for.

When the LDAP directory server starts normally, it generates messages similar to the following:

Note: The messages and the number of SQL server jobs started may differ in any of the following cases:

- You are starting your server for the first time.
- Migration needs to occur.
- Your server is using the change log.
- Your server is set to allow a higher number of database connections.

```
Job . . . : QDIRSRV      User . . . : QDIRSRV      System:  WARMERS
Number . . . : 174440
```

```
>> CALL PGM(QSYS/QGLDSVR)
Job 057448/QUSER/QSQSRVR used for SQL server mode processing.
Job 057340/QUSER/QSQSRVR used for SQL server mode processing.
Job 057448/QUSER/QSQSRVR used for SQL server mode processing.
Job 057166/QUSER/QSQSRVR used for SQL server mode processing.
Job 057279/QUSER/QSQSRVR used for SQL server mode processing.
Job 057288/QUSER/QSQSRVR used for SQL server mode processing.
Directory Services server started successfully.
```

Directory Services uses the first SQL server, 057448/QUSER/QSQSRVR, during LDAP server startup. Directory Services may start additional SQL servers during LDAP server startup as necessary if you are starting your server for the first time, if migration needs to occur, or if your server is using the change log. After startup, these SQL servers are dropped.

- | In this example, no additional SQL servers were used for migration or server startup and change log is not configured. Directory Services uses the next SQL server (057340/QUSER/QSQSRVR) for replication.
- | The very last connection in this example (057288/QUSER/QSQSRVR) is used for add, modify, modrdn, and delete operations. The other connections are used for search, bind, and compare.

On the directory servers **Database/Suffixes** Properties page in iSeries Navigator you specify the total number of SQL servers that Directory Services uses for directory operations after server startup. In addition, one SQL server is always configured for replication.

Monitor errors and access with the Directory Services job log

Viewing the job log for your LDAP server can alert you to errors and help you to monitor server access.

If your server is started, take these steps to view the QDIRSRV job log:

1. In iSeries Navigator, expand **Network**.
2. Expand **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Server Jobs..**
5. From the **File** menu, choose **Job Log**.

If your server is stopped, take these steps to view the QDIRSRV job log:

1. In iSeries Navigator, expand **Basic Operations**.
2. Click **Printer Output**.
3. QDIRSRV appears in the **User** column of iSeries Navigators right panel. To view the job log, double-click on **Qpjoblog** to the left of QDIRSRV in the same row.

Note: iSeries Navigator may be configured to show only spooled files. If QDIRSRV does not appear on the list, click **Printer Output**, then choose **Include** from the **Options** menu. Specify **All** in the **User** field, then click **OK**.

Note: Directory Services uses other system resources to perform some tasks. If an error occurs with one of those resources, the job log will indicate where to go for information. In some cases Directory Services may not be able to determine where to look. In those cases, look in the Structured Query Language (SQL) servers job log to see if the problem was related to SQL servers.

Use TRCTCPAPP to help find problems

Your server provides a communication trace to collect data on a communications line, such as a local area network (LAN) or a wide area network (WAN) interface. The average user may not understand the entire contents of the trace data. However, you can use the trace entries to determine whether a data exchange between two points actually took place.

The Trace TCP/IP Application (TRCTCPAPP) command with the *DIRSRV option can be used on the LDAP directory server to aid in finding problems with clients or applications.

For more detailed information on the uses of the TRCTCPAPP command with LDAP as well as the restrictions on required authorities, see TRCTCPAPP (Trace TCP/IP Application) Command Description.

For general information on using communications trace, see Communications trace.

Use the LDAP_OPT_DEBUG option to trace errors

- | Beginning with V5R2, you can use the LDAP_OPT_DEBUG option of the **ldap_set_option()** API to trace problems with clients that are using the LDAP C APIs. The debug option has multiple debug level setting that you can use to aid in troubleshooting problems with these applications.

| The following is an example of enabling the client trace debug option.

```
| int debugvalue= LDAP_DEBUG_TRACE | LDAP_DEBUG_PACKETS;  
| ldap_set_option( ld, LDAP_OPT_DEBUG, &debugvalue);
```

| An alternate way of setting the debug level is to configure the numerical value of the LDAP_DEBUG environment variable, for the job in which the client application runs, to the same numerical value that the debugvalue would be if the **ldap_set_option()** API is used.

| An example of enabling the client trace using the LDAP_DEBUG environment variable is the following:

```
| ADDENVVAR ENVVAR(LDAP_DEBUG) VALUE(0x0003)
```

| After running the client that produces the problem you are having, type the following at the iSeries prompt:

```
| DMPUSRTRC ClientJobNumber
```

| where ClientJobNumber is the number of the client job.

| To display this information interactively, type the following at the iSeries prompt:

```
| DSPPFM QAPOZDMP QPOZnnnnnn
```

| where nnnnnn is the job number.

| To save this information in order to send the information to service, take the following steps:

- | 1. Create a SAVF file using the create SAVF (CRTSAVF) command.
- | 2. Type the following at the iSeries command prompt.

```
| SAVOBJ OBJ(QAPOZDMP LIB(QTEMP) DEV(*SAVF) SAVF(xxx)
```

| where xxx is the name that you specified for the SAVF file.

Common LDAP client errors

Knowing the causes of common LDAP client errors can help you to solve problems with your server. For a complete list of LDAP client error conditions, see the OS/400 Directory Services topic under Programming in the iSeries Information Center.

The client error messages have the following format:

```
[Failing LDAP operation]:[LDAP client API error conditions]
```

Note: The explanation of these errors assumes that the client is communicating with an LDAP server on OS/400. A client communicating with a server on a different platform might get similar errors, but the causes and resolutions would most likely be different.

Common messages include the following:

- “ldap_search: Timelimit exceeded” on page 62
- “[Failing LDAP operation]: Operations error” on page 62
- “ldap_bind: No such object” on page 62
- “ldap_bind: Inappropriate authentication” on page 62
- “[Failing LDAP operation]: Insufficient access” on page 62
- “[failing LDAP operation]: Cannot contact LDAP server” on page 62

- “[failing LDAP operation]: Failed to connect to ssl server” on page 63

ldap_search: Timelimit exceeded

This error occurs when ldapsearches are performing slowly. To correct this error, you can do one or both of the following:

- Increase the search time limit for the LDAP directory server. See “Adjust performance of the LDAP directory server” on page 30 for information on doing this.
- Reduce the activity on your system. You can also reduce the number of active LDAP client jobs running.

[Failing LDAP operation]: Operations error

Several things can generate this error. To get information about the cause of this error for a particular instance, look at the QDIRSRV and Structured Query Language (SQL) server job logs as described in “Basic troubleshooting procedure for Directory Services” on page 59.

ldap_bind: No such object

A common cause of this error is that a user makes a typing mistake when performing an operation. Another common cause is when the LDAP client attempts to bind with a DN that does not exist. This often occurs when the user specifies what he or she mistakenly thinks is the administrator DN. For example, the user may specify QSECOFR or Administrator, when the actual administrator DN may be something like cn=Administrator.

For details about the error, look at the QDIRSRV job log as described in “Basic troubleshooting procedure for Directory Services” on page 59.

ldap_bind: Inappropriate authentication

- | The server returns Invalid credentials when the password or bind DN is incorrect. The server returns
- | inappropriate authentication when the client attempts to bind as one of the following:
- | • An entry that does not have a userpassword attribute
- | • An entry that represents an OS/400 user, which has a UID attribute and not a userpassword attribute.
- | This causes a compare to be done between the password specified and the OS/400 user password,
- | which do not match.
- | • An entry that represents a projected user and a bind method other than simple has been requested.
- | This error is usually generated when the client attempts to bind with a password that is not valid. To obtain
- | details about the error, look at the QDIRSRV job log as described in “Basic troubleshooting procedure for
- | Directory Services” on page 59.

[Failing LDAP operation]: Insufficient access

This error is usually generated when the binding DN does not have authority to do the operation (such as an add or delete) that the client requests. To get information about the error, look at the QDIRSRV job log as described in “Basic troubleshooting procedure for Directory Services” on page 59.

[failing LDAP operation]: Cannot contact LDAP server

The most common causes of this error include the following:

- An LDAP client makes a request before the LDAP server on the specified system is up and in select wait status.
- The user specifies a port number that is not valid. For example, the server is listening on port 386, but the client request attempts to use port 387.

To get information about the error, look at the QDIRSRV job log as described in “Basic troubleshooting procedure for Directory Services” on page 59. If the Directory Services server started successfully, the message Directory Services server started successfully will be in the QDIRSRV job log.

[failing LDAP operation]: Failed to connect to ssl server

This error occurs when the LDAP server rejects the client connection because a secure socket connection cannot be established. This can be caused by any of the following:

- The Certificate Management support rejects the clients attempt to connect to the server. Use Digital Certificate Manager to make sure that your certificates are set up properly, then restart the server and try to connect again.
- The user may not have read access to the *SYSTEM certificate store (by default /QIBM/userdata/ICSS/Cert/Server/default.kdb).

For OS/400 C applications, additional SSL error information is available. See the documentation for the individual Directory Services APIs for details.



Printed in U.S.A.