IDDS Transition Guide

iPlanet Directory Server

Version 5.0

June 2001

Contents

About This Guide

Welcome to iPlanet Directory Server. This manual describes how to transition your deployment from Innosoft Distributed Directory Server (IDDS) to iPlanet Directory Server.

This chapter contains the following sections:

- Prerequisite Reading
- Conventions Used In This Guide
- Related Information

Prerequisite Reading

Before you install Directory Server, we recommend that you read the *iPlanet Directory Server Deployment Guide*. The *Deployment Guide* covers key concepts on how to design and plan your directory service.

After you finish planning your directory service, follow the steps in the *iPlanet Directory Server Installation Guide* to install iPlanet Directory Server and its related software components.

You may also want to review the *iPlanet Directory Server Administrator's Guide*.

Conventions Used In This Guide

This section explains the conventions used in this book.

`Monospaced font`—This typeface is used for any text that appears on the computer screen or text that you should type. It is also used for filenames, functions, and examples.

NOTE	Notes and Cautions mark important information. Make sure you read the information before continuing with a task.

The greater than symbol (>) is used as a separator for successive menu selections. For example, Object > New > User means that you should pull down the Object menu, drag the mouse down to highlight New, and drag the mouse across to the New submenu in which you must select User.

Throughout this book you will see path references of the form:

`/usr/iplanet/servers/slapd-`*`serverID`*`/...`

The `/usr/iplanet/servers` directory is the default installation directory. If you have installed iPlanet Directory Server in a different location, you should adapt the path accordingly. *serverID* represents the server identifier you gave the server when you installed it. For example, if you gave the server an identifier of **phonebook**, then the actual path would be:

`/usr/iplanet/servers/slapd-phonebook/. . .`

All paths specified in this manual are in UNIX format. If you are using a Windows NT-based directory server, you should assume the NT equivalent file paths whenever UNIX file paths are shown in this guide.

Related Information

The document set for iPlanet Directory Server contains the following guides:

iPlanet Directory Server Installation Guide. Procedures for installing your Directory Server as well as procedures for migrating your Netscape Directory Server to iPlanet Directory Server.

iPlanet Directory Server Administrator's Guide. Procedures for the day-to-day maintenance of your directory service. Includes information on configuring server-side plug-ins.

iPlanet Directory Server Deployment Guide. Procedures for the day-to-day maintenance of your directory service. Includes information on configuring server-side plug-ins.

iPlanet Directory Server Configuration, Command, and File Reference. Information about using the command-line scripts shipped with Directory Server.

iPlanet Schema Reference. Information about all the schema used in the iPlanet suite of products.

Other useful iPlanet information can be found at the following Internet locations:

- iPlanet release notes and other documentation:
  http://docs.iplanet.com/docs/manuals/
- iPlanet product status:
  http://www.iplanet.com/support/technical_resources/
- iPlanet Professional Services information:
  http://www.iplanet.com/services/pro_serv/index.html
- iPlanet developer information:
  http://developer.iplanet.com/
- iPlanet learning solutions:
  http://www.iplanet.com/learning/index.html
- iPlanet product data sheets:
  http://www.iplanet.com/products/index.html
- iPlanet product technical support
  http://www.iplanet.com/support

Chapter 1

Major Differences

This chapter highlights the major differences between the Innosoft Distributed Directory Server (IDDS) version 4.5.2 and the iPlanet Directory Server version 5.0. It contains the following sections:

- Feature Comparison
- Supported Platforms

Feature Comparison

Table 1-1 Feature Comparison

Feature	IDDS	iPlanet Directory Server
LDAPv3	Y	Y
• Change log	Y	Y
• Change notification	Y	Y
• Schema publication	Y	Y
• Paging / virtual list view	Y	Y
• Server side sorting	N	Y
Database		
• Multiple databases	Y	Y
• Bulk load / export tools	Y	Y
• Referential integrity	Y	Y

Table 1-1 Feature Comparison *(Continued)*

Feature	IDDS	iPlanet Directory Server
• Collective attributes / CoS	Y	Y
• On-line backup / restore	N	Y
• Transaction log	N	Y
Distribution		
• Chaining	Y	Y
• Integrated replication	N	Y
• Multi-Master replication	N	Y
Security		
• SSL	Y	Y
• SSL client authentication	Y	Y
• Group-based access control	Y	Y
• Password hashing	Y	Y
• Password expiration	N	Y
• DIGEST-MD5	N	Y
• Macro ACIs	N	Y
• Role-based access control	N	Y
• Proxy authorization	N	Y
Administration		
• Command-line tools	Y	Y
• Graphical administration interface	Y	Y
• SNMP monitoring	Y	Y
• Schema updates over LDAP	N	Y

Supported Platforms

Customers running IDDS on versions of Solaris earlier than Solaris 8 are encouraged to upgrade their operating system to Solaris 8 before upgrading to iPlanet Directory Server.

Table 1-2 Platform Comparison

Platform	IDDS	iPlanet Directory Server
• Solaris SPARC 2.5.1	Y	N
• Solaris SPARC 2.6	Y	Y
• Solaris SPARC 7	Y	N
• Solaris SPARC 8	N	Y
• Solaris x86	Y	N
• Windows NT 4.0 Server	Y	Y
• Windows 2000 Server	Y	Y
• Windows 2000 Advanced Server	N	Y
• HP-UX 11.00	Y	Y
• AIX 4.3.3	Y	Y
• Tru64 UNIX 4.0D	Y	N
• Red Hat Linux	Y	N
• OpenVMS	N	N

Chapter 2

Transition Procedure

This chapter outlines the procedure for transitioning from the Innosoft Distributed Directory Server (IDDS) to iPlanet Directory Server.

Steps to Transition

1. Stop the IDDS process.
2. Export all of the databases to LDIF files, using `ldbmexport`.
3. Back up the installation directory: `/opt/IIIdds` or `[Program Files]\Innosoft` on Windows NT. This ensures that the directory server can be restored if a problem occurs during the transition procedure.
4. On UNIX platforms, remove the `symlink /etc/rc2.d/S93IIIdds` to ensure that the directory server does not start automatically. On Windows NT, use the Control Panel to configure that the directory server service is started manually.
5. If necessary, upgrade the operating system to the latest supported version.
6. Apply the required operating system patches as documented in the *iPlanet Directory Server Installation Guide*.
7. Choose an installation directory in the local file system for the iPlanet Directory Server. Do not install in the same directory as IDDS was installed, as this will prevent IDDS from being uninstalled properly.
8. Install iPlanet Directory Server 5.0 as described in the *iPlanet Directory Server Installation Guide.*
9. Use the iPlanet Console to configure the databases and other tuning settings for the iPlanet Directory Server. The following chapters provide more information on the differences between the configuration options of IDDS and iPlanet Directory Server.

10. Stop the iPlanet Directory Server.
11. On UNIX platforms, run the `idsktune` utility, located in the `/bin/slapd/server` directory. This utility will provide operating system tuning recommendations.
12. Export the `userRoot` database of the iPlanet Directory Server, using the `db2ldif` program. This will provide an LDIF file containing a sample access control specification in the `aci` attribute.
13. Edit the LDIF files that were obtained from the IDDS. You may need to add access control information to your LDIF files, because by default no `aci` attribute is present.
14. Create a schema file in the `slapd-<serverid>/config/schema` directory, to contain any attribute and object class definitions needed by the entries. See the `00core.ldif` file as an example of the new schema definition format.
15. Load the LDIF files for each of the database into the iPlanet Directory Server, using the `ldif2db` tools. Be aware that it may be necessary to make some modifications to the schema or LDIF file in order to ensure that the files can be loaded.
16. If you want to have the server start automatically on UNIX platforms, create a file in the `/etc/rc2.d` to invoke the `start-slapd` script to have the directory server started automatically.
17. At this point you can uninstall IDDS.
18. Reboot the system. This will ensure that any operating system patches and tuning settings take effect.
19. Ensure that the iPlanet Administration Server and iPlanet Directory Server can be started properly, and that the directory manager can access all the entries.
20. You may need to make additional configuration and tuning changes to the iPlanet Directory Server, in particular, access control and indexing. More information is available in the *iPlanet Directory Server Administrator's Guide*.

The following chapters provide more information on transitioning from the `sls.conf` file to equivalent settings in the iPlanet Directory Server configuration.

Product Installation and Contents

This chapter describes the differences in installation between Innosoft Distributed Directory Server and iPlanet Directory Server, and the layout of the files on the installation disk. It contains the following sections:

- Product Installation
- Package Contents

Product Installation

On Windows NT, both products are installed using a graphical installer framework.

On UNIX platforms, IDDS uses native operating system packaging tools. iPlanet Directory Server uses its own packaging system.

Package Contents

This section summarizes the changes in the package layout from IDDS to iPlanet Directory Server.

NOTE All directories in this section are given relative to `/usr/iplanet/servers/slapd-<`*server-id*`>`.

/opt/IIIdds/bin Directory

The LDAP command-line tools are present in the `shared/bin` subdirectory.

/opt/IIIdds/bin/ldapsearch

The `ldapsearch` is a shell-accessible interface to the LDAP search operation. In iPlanet Directory Server, this program is located in the `shared/bin` directory.

The `ldapsearch` program provided with iPlanet Directory Server uses a different SSL key and certificate database technology from IDDS. As a result, there is no direct equivalent for the `-c` (certificate filename), `-k` (private key filename), `-C` (certificate filename), `-P` (certificate directory pathname), `-Z` (start TLS), or `-g` (start TLS and bind) options.

/opt/IIIdds/bin/ldapadd

In iPlanet Directory Server, the `ldapmodify` program provides the same function as the `ldapadd` command, so there is no separate `ldapadd` program.

/opt/IIIdds/bin/ldapmodify

The `ldapmodify` utility is a shell-accessible interface to the LDAP Add, Delete, Modify, and ModifyDN operations. In iPlanet Directory Server, the `ldapmodify` program is located in the `shared/bin` directory.

/opt/IIIdds/bin/ldapdelete

`ldapdelete` is a shell-accessible interface to the LDAP Delete operation. The `ldapdelete` program is now located in the `shared/bin` directory.

/opt/IIIdds/bin/tclsnmpc

This program is not provided with iPlanet Directory Server.

/opt/IIIdds/bin/show_mib.tcl

This script is not provided with iPlanet Directory Server as it does not support RFC 2248, *Network Services Monitoring MIB*.

/opt/IIIdds/bin/searchrate

The `searchrate` program can be used to measure search response by generating a simulated client load. This program is not provided with iPlanet Directory Server.

/opt/IIIdds/sbin Directory

The command-line administrative tools and daemons are present in the `bin` subdirectory.

/opt/IIIdds/sbin/idds

In iPlanet Directory Server, the server is started using the `slapd-<server-id>/start-slapd` script.

/opt/IIIdds/sbin/ldbmtest

This program is not provided with iPlanet Directory Server.

/opt/IIIdds/sbin/ldbmexport

In IDDS, the `ldbmexport` program is used to convert an `LDBM` database back into its LDIF text format. In iPlanet Directory Server, it is replaced by the `db2ldif` script.

/opt/IIIdds/sbin/ldbmcheck

There is no equivalent for this program in iPlanet Directory Server.

/opt/IIIdds/sbin/ldbmindex

The `ldbmindex` program is replaced by the `db2index.pl` script in iPlanet Directory Server.

/opt/IIIdds/sbin/ldif2ldbm

The `ldif2ldbm` program is replaced by the `ldif2db` script.

The `-e` argument (write to error file rejected entries) and `-m` argument (merge) are not available in iPlanet Directory Server.

/opt/IIIdds/sbin/ldif

The `ldif` program is used to convert arbitrary data values to LDIF format. `ldif` takes an attribute name as an argument, and reads the attribute value(s) from standard input. It produces the LDIF formatted attribute line(s) on standard output.

In iPlanet Directory Server, this program is located in the `bin/slapd/server` directory.

/opt/IIIdds/sbin/snmpd

An SNMP subagent is provided with iPlanet Directory Server. Consult the *iPlanet Directory Server Configuration, Command, and File Reference Guide* for more information.

/opt/IIIdds/sbin/passcode

This program is not provided with iPlanet Directory Server.

/opt/IIIdds/sbin/replmove

There is no equivalent for this program in iPlanet Directory Server, as the replication mechanism is different.

/opt/IIIdds/sbin/selfsign

The `selfsign` program is not applicable to iPlanet Directory Server. Instead, server key pair generation is done using the console. Note that the console does not generate and iPlanet Directory Server does not support self-signed certificates.

/opt/IIIdds/sbin/idsktune

In iPlanet Directory Server, this program is located in the `bin/slapd/server` subdirectory.

/opt/IIIdds/sbin/ldifxform

The `ldifxform` program can be used to rewrite and reformat the contents of an LDIF file. This program is not available in iPlanet Directory Server.

/opt/IIIdds/sbin/tcpdmatch

This program is not provided with iPlanet Directory Server, because the TCP wrappers functionality is not supported.

/opt/IIIdds/sbin/db_dump

This program is not provided with iPlanet Directory Server.

/opt/IIIdds/sbin/db_stat

This program is not provided with iPlanet Directory Server.

/opt/IIIdds/sbin/viewldbg

This program is not provided with iPlanet Directory Server.

/opt/IIIdds/sbin/snapshot.tcl

This program is not provided with iPlanet Directory Server.

/opt/IIIdds/data Directory

The data files for iPlanet Directory Server are provided in the `slapd-<serverID>` subdirectory.

/opt/IIIdds/data/sls.at.conf and sls.oc.conf

In iPlanet Directory Server, sample schema is located in the `slapd-<serverID>/config/schema` subdirectory.

/opt/IIIdds/data/sls.conf

In iPlanet Directory Server, the configuration is stored in the `slapd-<serverID>/config/dse.ldif` file. This file must not be modified while the server is running.

/opt/IIIdds/tmp Directory

The log files are present in the `slapd-<serverID>/logs` subdirectory.

/opt/IIIdds/tmp/sls.pid

This file is called `pid` in iPlanet Directory Server.

/opt/IIIdds/tmp/sls.args

This file is not generated by iPlanet Directory Server.

/opt/IIIdds/tmp/sls.log

This file is called `access` in iPlanet Directory Server.

/opt/IIIdds/tmp/debug.log

This file is called `errors` in iPlanet Directory Server.

/opt/IIIdds/tmp/iddserr.log

This information is in the `errors` file in iPlanet Directory Server.

/opt/IIIdds/doc Directory

The on-line documentation is present in the `manual` subdirectory.

/opt/IIIdds/lib Directory

The library files are present in the `lib` subdirectory.

/opt/IIIdds/lib/ldbm-backup.sh

The `db2bak` script can be used to back up the iPlanet Directory Server databases. The backup is done to an internal file format which is much quicker to produce and restore than and ldif file. Backing up databases to LDIF can be done with the `db2ldif` script.

/opt/IIIdds/lib/mibs Directory

iPlanet Directory Server does not support RFC 2248 MIBs.

/opt/IIIdds/samples Directory

Sample LDIF files are provided in the `slapd-<`*serverID*`>/ldif` subdirectory. The Innosoft LDAP Proxy Server and dirportal samples, and performance test tools, are not provided with iPlanet Directory Server.

/etc/init.d/S93IIIdds

This file is not provided with iPlanet Directory Server.

Chapter 4

Configuration File Options

This chapter separates the IDDS configuration file options into global and backend-specific categories, describing each option and equivalent (if any) in iPlanet Directory Server. It is composed of the following sections:

- Global Options
- General Backend Options
- LDBM Backend-Specific Options
- Chaining Options

Global Options

Options described in this section apply to all backends of IDDS.

`access to <what> [ by <who> <accesslevel> ]+`

This option grants access (specified by `<accesslevel>`) to a set of entries and/or attributes (specified by `<what>`) by one or more requesters (specified by `<who>`). See "Migrating Access Control" in the next chapter for more details on how this is represented in iPlanet Directory Server.

`adminaddr <address>`

The value of this option is provided to clients in the `administratorsAddress` attribute of the root DSE. There is no direct equivalent in iPlanet Directory Server.

`approximation {soundex|metaphone}`

This option specifies the algorithm by which IDDS performs approximate matching. This option is not available in iPlanet Directory Server, which only supports metaphone.

`attribute <name> [<oid>] <syntax>`

This option associates a syntax with an attribute name. In iPlanet Directory Server, this information is provided in the schema files.

`attribute <name> [<oid>] sup <attrname>`

This option defines an attribute as a subtype of another attribute type. In iPlanet Directory Server, this information is provided in the schema files.

`disable_locking { yes | no }`

This option specifies whether the underlying disk databases should disable the file locking service. This option has no equivalent in iPlanet Directory Server.

`http_port <portno>`

This option specifies the TCP port for processing HTTP requests. There is no equivalent in iPlanet Directory Server as server administration is performed through the iPlanet Console.

`https_port <portno>`

This option specifies the TCP port for processing HTTPS (HTTP protected by SSL) requests. There is no equivalent in iPlanet Directory Server as server administration is performed through the iPlanet Console.

`include <filename>`

This option specifies that IDDS should read additional schema configuration information from the given file before continuing with the next line of the current file. There is no equivalent in iPlanet Directory Server; schema files are read from the schema directory.

`install_path <pathname>`

This option specifies an alternate installation directory. There is no equivalent in iPlanet Directory Server.

`ldaps_port <portno>`

This option specifies the port number for the obsolete alternate LDAP/SSL mapping. In iPlanet Directory Server, it is controlled by the `nsslapd-secureport` attribute.

`listen_backlog<integer>`

This option specifies the maximum number of pending connections that should be queued. This option is not available in iPlanet Directory Server.

`loglevel <integer>`

This option specifies the level at which debugging statements and operation statistics should be logged on UNIX. This option is not available in iPlanet Directory Server.

`logpriority <loglevel>`

This option specifies the priority level to which log messages should be syslogged on UNIX. This option is not available in iPlanet Directory Server.

`maxconns <integer>`

This option specifies the maximum number of simultaneous connections that IDDS will permit. On UNIX platforms this is also limited by the number of file descriptors permitted to each process. In iPlanet Directory Server, this is controlled by the `nsslapd-maxdescriptors` attribute.

`mgrdn  <dn>`

This option specifies the Distinguished Name of the global manager who is exempt from access control restrictions. In iPlanet Directory Server, this is represented by the `nsslapd-rootdn` attribute of `cn=config`.

`mgrpw  <string>`

This option specifies a password for the DN given above that will always work, regardless of whether an entry exists with the given DN. In iPlanet Directory Server, this is represented by the `nsslapd-rootpw` attribute of `cn=config`.

`nodelay { on | off }`

This option specifies whether TCP packet delays should be disabled for returning responses. In iPlanet Directory Server, this is controlled by the `nsslapd-nagle` attribute.

`notimeout <dn>`

This option specifies the identity of a user whose connections should not be timed out. In iPlanet Directory Server, this is represented by operational attributes of that user's entry.

`objectclass <name> ...`

In iPlanet Directory Server, this information is provided in the schema files.

`ops_unthreaded { yes | no }`

This option specifies whether operations should be run by the same thread as the connection. This feature is not available in iPlanet Directory Server.

`port <portnumber>`

This option specifies the TCP port number on which IDDS should listen for incoming connections. In iPlanet Directory Server, this is represented by the `nsslapd-port` attribute of `cn=config`.

`referral <url> [ <url> ...]`

This option specifies the referral to pass back when IDDS cannot find a local database to handle a request. In iPlanet Directory Server, this is represented by the `nsslapd-referral` attribute of `cn=config`.

`replogfile <filename>`

This option tells IDDS to write a log file of updates made by clients. This feature is not available in iPlanet Directory Server.

`reverse_lookup { yes | no }`

This feature is not available in iPlanet Directory Server.

`rewrite_rfc1274 { yes | no }`

When this option is set to yes, RFC 1274 (*The COSINE and Internet X.500 Schema*) attribute type names provided by clients in add and modify requests are converted to LDAPv3 compatible names as described in RFC 2256 (*A Summary of the X.500(96) User Schema for use with LDAPv3*). The option is provided for compatibility with LDAPv2 applications, such as the Entrust CA. If this option is set, then multiple attributes with the same values will be returned to clients. In iPlanet Directory Server, this is controlled by the `nsslapd-rewrite-rfc1274` attribute of `cn=config`.

`schemacheck { on | off }`

This option turns schema checking on or off. In iPlanet Directory Server, this is controlled by the `nsslapd-schemacheck` attribute of `cn=config`.

`serverdn <dn>`

This option has no equivalent in iPlanet Directory Server.

`serveruid <integer>`

This option specifies a UNIX `userid` to which IDDS should `setuid()` after it has started running. In iPlanet Directory Server, this is controlled by the `nsslapd-localuser` attribute of `cn=config`.

`sizelimit <integer>`

This option specifies the maximum number of entries to return from a search operation. In iPlanet Directory Server, this is represented by the `nsslapd-sizelimit` attribute of `cn=config`.

`snmp_port <integer>`

This option specifies the UDP port snmpd should use. There is no direct equivalent in iPlanet Directory Server.

`ssl_capath <pathname>`

This option specifies the name of a directory containing trusted CA certificates in PEM format. iPlanet Directory Server uses a different SSL certificate and key database technology, so this option has no direct equivalent. It will be necessary to import any CA certificates into the new database using the iPlanet Console.

`ssl_cafile <filename>`

This option specifies the name of a file containing trusted CA certificates in PEM format. iPlanet Directory Server uses a different SSL certificate and key database technology, so this option has no direct equivalent. It will be necessary to import any CA certificates into the new database using the iPlanet Console.

`ssl_cert <filename>`

This option specifies the location on disk of the file containing IDDS's own certificate. iPlanet Directory Server uses a different SSL certificate and key database technology, so this option has no direct equivalent. It will be necessary to generate a new key pair and recertify any servers when transitioning to iPlanet Directory Server.

`ssl_cert_required { yes | no }`

This option specifies whether a client must provide a certificate when negotiating SSLv3. Consult the *iPlanet Directory Server Administration Guide* for additional information on SSL configuration.

`ssl_key <filename>`

This option specifies the location on disk of the file containing IDDS's own certificate. iPlanet Directory Server uses a different SSL certificate and key database technology, so this option has no direct equivalent. It will be necessary to generate a new key pair and recertify any servers when transitioning to iPlanet Directory Server.

`ssl_version <integer>`

This option specifies what versions of SSL are supported. Consult the *iPlanet Directory Server Administration Guide* for additional information on SSL configuration.

`stacksize_conn <integer>`

This option specifies the size in bytes of the stack for each thread maintaining the state of a connection. There is no equivalent for iPlanet Directory Server.

`stacksize_op <integer>`

This option specifies the size in bytes of the stack for each thread maintaining the state of an operation. There is no equivalent for iPlanet Directory Server.

`statslog full | op | conn | mods | none | off`

This option specifies the kinds of operations that the directory server will log. This feature is not available in iPlanet Directory Server.

`timelimit <integer>`

This option specifies the maximum number of seconds (in real time) IDDS will spend answering a search request. In iPlanet Directory Server, this is represented by the `nsslapd-timelimit` attribute of `cn=config`.

`timeout <integer>`

This option specifies the maximum number of seconds that a client connection can remain idle before IDDS closes it. In iPlanet Directory Server, this is represented by the `nsslapd-idletimeout` attribute of `cn=config`.

`version_override <string>`

This option overrides the implementation name in the `cn=monitor` entry. This feature is not available in iPlanet Directory Server.

`virtual_attrs tcl <scriptname> <objectclass>`

This option specifies the location of a virtual attribute script file. This features is not available in iPlanet Directory Server. Contact iPlanet Professional Services at http://www.iplanet.com/services/pro_serv/index.html for assistance in converting virtual attribute script functionality for use with iPlanet Directory Server.

General Backend Options

Options in this section apply to the IDDS backend in which they are defined.

`confidential { yes | no }`

This option specifies whether data stored in a backend is confidential. This feature is not available in iPlanet Directory Server.

`database <databasetype>`

This option marks the beginning of a new database definition. iPlanet Directory Server stores its configuration as entries below `cn=config,cn=ldbm database`.

`end_database`

This option marks the end of a database definition. iPlanet Directory Server stores its configuration as entries.

`lastmod { on | off }`

This option controls whether IDDS will automatically maintain the `modifiersName`, `modifyTimestamp`, `creatorsName`, and `createTimestamp` attributes for entries. In iPlanet Directory Server, this is represented by the `nsslapd-lastmod` attribute of `cn=config`.

`mgrdn <dn>`

This option specifies the DN of an entry that is not subject to access control or administrative limit restrictions for operations on this database. This per-database manager is not available in iPlanet Directory Server.

`mgrpw <string>`

This option specifies a password for the DN given above that will always work, regardless of whether an entry with the given DN exists or has a password. This option has no equivalent in iPlanet Directory Server.

`readonly { on | off }`

This option puts the database into “read-only” mode. In iPlanet Directory Server, this is represented by the `nsslapd-readonly` attribute of `cn=config`.

`referral <URL>`

If IDDS receives a DIT modification operation for this database from a user other than the manager or one listed as an `updatedn` (see below), IDDS will return a referral to another server indicated by this URL. This option has no equivalent in iPlanet Directory Server.

`sizelimit <integer>`

This option specifies the maximum number of entries to return from this database. There is no direct equivalent in iPlanet Directory Server.

`suffix <dn suffix>`

This option specifies the DN suffix of queries that will be passed to this backend database. iPlanet Directory Server stores its configuration in entries differently; when the database is created, the suffix is specified in the mapping tree entry below `cn=config`.

`updatedn <dn>`

This option specifies the DN of another LDAP server allowed to make changes to the replica. Because the replication mechanism in iPlanet Directory Server is different, there is no direct equivalent.

LDBM Backend-Specific Options

Options in this category apply to an LDBM backend database.

`cachesize <integer>`

This option specifies the size in *entries* of the in-memory cache maintained by the LDBM backend database instance. In iPlanet Directory Server, this is controlled by the `nsslapd-cachesize` attribute of the database configuration entry.

`cachewrites { yes | no }`

This option specifies whether modifications to the database should be cached in memory for up to one minute. This option has no equivalent in iPlanet Directory Server.

`dbcachesize <integer>`

This option specifies the size in bytes of the in-memory cache used by the Sleepycat b-tree database for each file. In iPlanet Directory Server, this is controlled by the `nsslapd-dbcachesize` attribute of the database configuration entry.

`dbcachesize_load <integer>`

This option specifies an alternate value of `dbcachesize` that `ldif2ldbm` should use. This option has no equivalent in iPlanet Directory Server.

`directory <directory>`

This option specifies the directory where the LDBM files containing the database and associated indices live. In iPlanet Directory Server, this is controlled by the `nsslapd-directory` attribute of the database configuration entry.

`ignore_onelevel_refs { yes | no }`

When set to `yes`, this option causes referrals to be ignored when performing one-level searches; referral entries would be treated as regular local entries. This feature is not available in iPlanet Directory Server.

`ignore_refs { yes | no }`

If set to `yes`, then referral entries cannot be used in the LDBM database. This option has no equivalent in iPlanet Directory Server.

`index <attrlist> [pres,eq,approx,sub,none] [preload] [unique] [referential]`

This option specifies the indices to maintain for the given attribute. In iPlanet Directory Server, there is a separate configuration entry for each attribute being indexed.

There is no equivalent in iPlanet Directory Server for the `preload` modifier.

In iPlanet Directory Server, the uid uniqueness and referential integrity plug-ins can be used to provide the IDDS functionality described by the `unique` and `referential` modifiers.

`indexcachesize <integer>`

This option specifies the maximum size in datums of the internal cache array for each attribute index. There is no equivalent in iPlanet Directory Server.

`indexonly { yes | no }`

This option specifies whether filter components in subtree and one level searches that would not make use of attribute index files should be ignored. There is no equivalent in iPlanet Directory Server.

`mode <integer>`

This option specifies the file protection mode that newly created database index files should have. There is no equivalent in iPlanet Directory Server as files should always be of mode 0600.

`pagesize <integer>`

This option specifies the size of each page in the underlying database files. There is no equivalent in iPlanet Directory Server.

`preload_entries { yes | no }`

This option specifies whether IDDS should load all the entries into its cache when it starts. This feature is not available in iPlanet Directory Server.

`pwdhash {none|crypt|ssha}`

In iPlanet Directory Server, this feature is provided by the password hashing plug-ins.

`replogfile <filename>`

This option tells IDDS to write a log file of updates made by clients. As iPlanet Directory Server implements its own replication without needing a log file, there is no equivalent for this option.

`require_index {yes | no}`

This option specifies whether search filters must make use of at least one attribute index. In iPlanet Directory Server, this is controlled by the `nsslapd-require-index` attribute of the database configuration entry.

Chaining Options

This section describes options that can be set in a backend of the `chaining` type.

`chainto <URL> ...`

This option specifies the servers to which chained operations should be forwarded. Multiple URLs can be specified, and IDDS will try each in turn until a server can be contacted. This option is represented in iPlanet Directory Server as the `nsFarmServerURL` attribute.

`encryptchain { yes | no }`

This option specifies whether IDDS should use SSL/TLS to encrypt the chained operations. There is no direct equivalent in iPlanet Directory Server.

`ldapversion <integer>`

This specifies the version to present when binding. This option is not available in iPlanet Directory Server.

`mapto <dn>`

The `mapto` option is not available in iPlanet Directory Server.

`maxhops <integer>`

This specifies the number of times an operation can be chained. In iPlanet Directory Server, this is represented by the `nsHopLimit` configuration attribute.

`maxwait <integer>`

This specifies the maximum number of seconds IDDS will wait for a reply from a chained operation before abandoning it. There is no direct equivalent in iPlanet Directory Server.

Chapter 5

Access Control

This chapter documents the differences in the access control models and syntax between IDDS and iPlanet Directory Server and gives some simple examples. This will help you implement equivalent access control on your data in iPlanet Directory Server as you have in IDDS. This chapter contains the following sections:

- Migrating Access Control
- A Quick Comparison of Syntax and Semantics
- Evaluating Access Control
- Groups
- Examples
- Specifying the Entries and Attributes Subject to Access Control
- Specifying the Users to Whom Access is Granted
- Specifying Access Permissions to Grant

Migrating Access Control

The migration process for access control is as follows:

1. Rewrite IDDS access directives as iPlanet Directory Server Access Control Instructions (ACIs).
2. Optionally, convert groups into preferred `groupOfUniqueNames` format.
3. Import the ACI into the iPlanet Directory Server directory tree.
4. Verify the behavior of the ACIs.

IDDS access directives must be rewritten as a set of iPlanet Directory Server ACIs. You can create ACIs in LDIF form or you may create them using the iPlanet Directory Server Console (or a combination of both methods). If you write in LDIF form, you have the choice of inserting them into the LDIF file that will be used to initialize the database or adding them afterwards with `ldapmodify` or the console once the database has been initialized. The console has a specialized ACI editor but it can also handle ACI in LDIF form.

The recommended approach is to insert ACIs with a text editor into the LDIF file used to load the database. Then use the console to quickly experiment with and tweak the ACIs.

A Quick Comparison of Syntax and Semantics

IDDS access directives are written in the `sls.conf` configuration file. They can be specific to a backend database or global to all backend databases.

iPlanet Directory Server ACIs, on the other hand, are stored in the directory as attributes of entries. If an entry containing an ACI is a leaf entry, then the ACI applies to that entry only. Otherwise the ACI applies to the entry itself and all entries below it.

If there is one database suffix superior to all others, then that suffix entry is a common ancestor entry into which the global ACIs can be placed. Otherwise the global ACIs can be duplicated in the suffix entry of each database backend. A database's local ACIs can be placed in the database suffix entry.

The syntax of IDDS access directives and iPlanet Directory Server ACIs are quite different. The syntax of an IDDS access directive is:

```
access to <what> [ by <who> <access> ]+
```

The syntax of an iPlanet Directory Server ACI value is:

```
(<target>)(version 3.0; acl "name"; [ <permission> <bind_rule>; ]+)
```

Comparing these syntaxes, `<what>` is analogous to `<target>`, `<who>` is analogous to `<bind_rule>`, and `<access>` is analogous to `<permission>`. Each of these are discussed in more detail later.

ACIs may be given a descriptive name for ease of management and comprehension, for example, "Default anonymous access."

IDDS allows regular expressions in some places. However, only wildcards may be used in iPlanet Directory Server. The most common IDDS regular expression construct ".*" may be represented in iPlanet Directory Server using the "*" wildcard.

IDDS has a hierarchy where each level implies all lower levels of access. For example, granting read access also grants search and compare. In iPlanet Directory Server, rights are granted independently of one another (but note that "all" is shorthand for all rights other than proxy).

In IDDS write access is used for attributes and also for entry addition and deletion. In iPlanet Directory Server write is only used for attributes; there are separate add and delete permissions for entry addition and deletion.

Evaluating Access Control

In IDDS, access directives local to the current database are examined first, followed by global access directives. The order of evaluation of access directives makes their placement in the configuration file important.

In iPlanet Directory Server, the server compiles a list of the ACIs present on the entry and on the parent entries up to the top level entry. ACIs are evaluated across all databases for a server. The evaluation of the list of ACIs is done based on the semantics of the ACIs, not on their placement in the directory tree. The most restrictive ACI in the list takes precedence.

For more information, see the *iPlanet Directory Server Administrator's Guide.*

Groups

A group may be defined as an entry of objectclass `groupOfNames` or `groupOfUniqueNames`, the difference being that an optional unique identifier may be assigned to DN's contained in the latter. These groups are called static groups in iPlanet Directory Server terminology. For example, consider the following entry of objectclass `groupOfNames`:

```
dn: cn=mgrs,dc=innosoft,dc=com
objectclass: groupOfNames
cn: mgrs
member: cn=John Smith,dc=innosoft,dc=com
member: cn=Jane Doe,dc=innosoft,dc=com
```

Both IDDS and iPlanet Directory Server permit groups of either objectclass to be referenced in access control statements. However, it is easier to manage a static group in the iPlanet Directory Server Console if it is a `groupOfUniqueNames` because there is a specialized editor for such entries. You may wish to convert all your groups into this form. The same example group expressed as a `groupOfUniqueNames` would look like this:

```
dn: cn=mgrs,dc=innosoft,dc=com
objectclass: groupOfUniqueNames
cn: mgrs
uniquemember: cn=John Smith,dc=innosoft,dc=com
uniquemember: cn=Jane Doe,dc=innosoft,dc=com
```

Examples

The following examples have some lines folded for readability. This line folding is supported in both the IDDS configuration file and the LDIF format used to write ACIs in iPlanet Directory Server.

General Read Access

IDDS

```
access to * by * read
```

iPlanet Directory Server

```
aci: (targetattr=*)(version 3.0; acl "anyone read";
   allow (read,search,compare) userdn="ldap:///anyone";)
```

This ACI must be placed in the suffix entry of the database to which it applies.

Access to Subtrees

IDDS

```
access to dnsub="ou=sales,dc=innosoft,dc=com" by * search
   access to dnsub="dc=innosoft,dc=com" by * read
```

iPlanet Directory Server

```
aci: (targetattr=*)(target="ldap:///ou=sales,dc=innosoft,dc=com")
   (version 3.0; acl "anyone search";
   allow (search,compare) userdn="ldap:///anyone";)
aci: (targetattr=*)(target="ldap:///dc=innosoft,dc=com")
   (version 3.0; acl "anyone read";
   allow (read,search,compare) userdn="ldap:///anyone";)
```

Access to a Specific Attribute

IDDS

```
access to dnsub="dc=innosoft,dc=com" attr=phone
   by self write
   by dnsub="dc=innosoft,dc=com" search
   by domain=.*\.innosoft\.com read
   by * compare
access to dnsub="dc=innosoft,dc=com"
   by self write
   by dnsub="dc=innosoft,dc=com" compare
   by * none
```

iPlanet Directory Server

```
aci:(targetattr="phone")(target="ldap:///dc=innosoft,dc=com")
   (version 3.0; acl "phone";
   allow (search,compare) userdn = "ldap:///*,dc=innosoft,dc=com";
   allow (read,search,compare) dns = "*.innosoft.com";
   allow (compare) userdn = "ldap:///anyone";)
aci: (targetattr=*)(target="ldap:///dc=innosoft,dc=com")
   (version 3.0; acl "default";
   allow (all) userdn = "ldap:///self";
   allow (compare) userdn = "ldap:///*,dc=innosoft,dc=com";)
```

Note that there is no need to translate the “by * none” portion of the IDDS directive because by default users are denied access. Also note that there is no need to translate “by self write” in the phone ACI since that is already in the default ACI applying to all attributes—unlike IDDS, both ACIs are considered when looking at the phone attribute.

Group Management

IDDS

```
access to attr=member,entry by dnattr=member selfwrite
```

iPlanet Directory Server

```
aci: (targetattr=member)(version 3.0; acl "self-remove group";
   allow (selfwrite) userattr = "member#USERDN";)
```

The above ACI allows any group member to remove themselves from a group.

Access By a Group

IDDS

```
access to * by dnbase="cn=admin,dc=innosoft,dc=com" write
   by group="cn=mgrs,dc=innosoft,dc=com" write
   by * read
```

iPlanet Directory Server

```
aci: (targetattr=*)(version 3.0; acl "mgrs write";
   allow (all) userdn = "ldap:///cn=admin,dc=innosoft,dc=com";
   allow (all) groupdn = "ldap:///cn=mgrs,dc=innosoft,dc=com";
   allow (read,search,compare) userdn = "ldap:///anyone";)
```

Specifying the Entries and Attributes Subject to Access Control

In IDDS terminology this is known as <what> and in iPlanet Directory Server terminology <target>.

Everything

IDDS

```
   to *
```

iPlanet Directory Server

```
   (target="<suffix>")(targetattr=*)
```

There is no direct equivalent in iPlanet Directory Server to target all entries. Instead, for each database suffix you must have an ACI targeting the suffix. For example, for the database suffix “dc=innosoft,dc=com”, use an ACI containing:

```
   (target="ldap:///dc=innosoft,dc=com")
```

If the ACI is in the database suffix entry, then you may omit the target specification entirely, but not the `targetattr` keyword. In iPlanet Directory Server, when you target an entry, the target does not include all of the attributes under that entry.

A Specific Entry

IDDS

```
to dnbase=<dn>
```

iPlanet Directory Server

```
(targetfilter=(<rdn>))
```

Targeting a single directory entry is not as straightforward in iPlanet Directory Server. The *iPlanet Directory Server Administrator's Guide* discusses several possibilities, one of which is to use `targetfilter=(<rdn>)`. For example:

```
dnbase="dc=innosoft,dc=com"
```

becomes:

```
(targetfilter=(dc=innosoft))
```

The above example works provided none of the entries below the target entry contain the attribute `dc=innosoft`.

All Entries in a Subtree

IDDS

```
to dnsub=<dn>
```

iPlanet Directory Server

```
(target="ldap:///<dn>")
```

Entries Whose DNs Have a Common Pattern

IDDS

```
to dn=<regular expression>
```

iPlanet Directory Server

```
(target="ldap:///<wildcard-dn>")
```

Only wildcards may be used in iPlanet Directory Server targets. For example:

```
dn="uid=.*,ou=.*,dc=siroe,dc=com"
```

becomes:

```
(target="ldap:///uid=*,ou=*,dc=siroe,dc=com")
```

Entries Satisfying an LDAP Filter

IDDS

```
filter=<ldap filter>
```

iPlanet Directory Server

```
(targetfilter = "<ldap filter>")
```

Specific Attributes

IDDS

```
attrs=<attribute list>
attr=<attribute list>
```

iPlanet Directory Server

```
(targetattr = "attribute1 || attribute2 ... || attributen")
```

For example:

```
attr=cn,sn,uid
```

becomes

```
(targetattr = "cn || sn || uid")
```

The IDDS pseudo-attributes entry and children are not needed in iPlanet Directory Server. You do not need explicit access to the entry in addition to its attributes. To delete an entry you must be granted the “Delete” right. To add an entry you must be granted the “Add” right.

Specifying the Users to Whom Access is Granted

In IDDS terminology this is known as `<who>` and in iPlanet Directory Server terminology `<bind_rule>`.

Everyone

IDDS

```
*
```

iPlanet Directory Server

```
userdn = "ldap:///anyone"
```

Self

IDDS

```
self
```

iPlanet Directory Server

```
userdn = "ldap:///self";
userdn = "ldap:///all"
```

In iPlanet Directory Server, the `anyone` keyword grants anonymous access; the `all` keyword grants access to all authenticated users.

Users Whose DNs Have a Common Pattern

IDDS

```
dn=<regular expression>
```

iPlanet Directory Server

```
userdn = "ldap:///<wildcard-dn>";
```

Only wildcards may be used in iPlanet Directory Server. For example:

```
dn="uid=.*,ou=.*,dc=siroe,dc=com"
```

becomes:

```
userdn = "ldap:///uid=*,ou=*,dc=siroe,dc=com";
```

A Specific User

IDDS

```
dnbase=<dn>
```

iPlanet Directory Server

```
userdn = "ldap:///<dn>";
```

Users Whose DN is Under a Specific Subtree

IDDS

```
dnsub=<dn>
```

iPlanet Directory Server

```
userdn = "ldap:///*,<dn>";
```

Users Connecting From a Specific Network Location

IDDS

```
addr=<regular expression>
```

iPlanet Directory Server

```
ip = "<wildcard-ip-address>";
```

Only wildcards may be used in iPlanet Directory Server. For example:

```
addr=129\.153\.129\..*
```

becomes

```
ip = "129.153.129.*";
```

Users in a Specific DNS Domain

IDDS

```
domain=<regular expression>
```

iPlanet Directory Server

```
dns = "<wildcard-domain>";
```

For example:

```
domain=.*\.innosoft\.com
```

becomes:

```
dns = "*.innosoft.com";
```

Users Whose DN Matches an Attribute Value

IDDS

```
dnattr=<dn-valued attribute name>
```

iPlanet Directory Server

```
userattr = "<dn-valued attribute name>#USERDN";
```

Users Belonging to a Specific Group

IDDS

```
group=<dn>
```

iPlanet Directory Server

```
groupdn = "ldap:///<dn>";
```

There is no need to change the directory group entry referenced by `<dn>`.

Specifying Access Permissions to Grant

In IDDS terminology this is known as `<access>` and in iPlanet Directory Server terminology `<permission>`, defined as `allow | deny (<rights>)`.

IDDS has the following levels of access:

```
none | compare | search | read | selfwrite | write
```

iPlanet Directory Server has the following rights:

```
read | write | add | delete | search | compare | selfwrite |
proxy | all
```

IDDS has a hierarchy where each level implies all lower levels of access. For example, granting read access also grants search and compare. In iPlanet Directory Server, rights are granted independently of one another (but note that “all” is shorthand for all rights except proxy).

Compare, search, read, and selfwrite have the same meaning in IDDS and iPlanet Directory Server. So, read access in IDDS can be replaced with (read, search, compare) rights in iPlanet Directory Server. Write access in IDDS can be replaced with (all) rights in iPlanet Directory Server.

In IDDS write access is used for attributes and also for entry addition and deletion. In iPlanet Directory Server write is only used for attributes; there are separate add and delete permissions for entry addition and deletion.

To Add, Delete, or Modify an Attribute

IDDS: You must have write access to the attribute and to the entry itself through the “entry” attribute.

iPlanet Directory Server: You must have write permission to the attribute. You do not need separate write permission to the entry itself.

To Add an Entry

IDDS: You must have write access to the “children” attribute of the parent.

iPlanet Directory Server: You must have add permission on the entry (inherited of course from an ACI higher in the tree).

To Delete an Entry

IDDS: You must have write access to the “entry” attribute.

iPlanet Directory Server: You must have delete permission on the entry.

iPlanet Directory Server Information

This chapter covers information provided by the iPlanet Directory Server, and includes the following sections:

- Monitoring Using SNMP
- Monitoring Using LDAP
- Root DSE
- Change Log
- Subschema
- Operational Attributes in Entries

NOTE The iPlanet Directory Server automatically maintains certain attributes and entries that are available for clients to search. This information is read-only.

Monitoring Using SNMP

iPlanet Directory Server contains an SNMP subagent for UNIX platforms, and integrates with Windows NT and Windows 2000 SNMP framework. iPlanet Directory Server provides a MIB that is similar to that of RFC 2248 *Network Services Monitoring MIB*; however, it does not entirely support RFC 2248, because the OIDs of the management variables are different.

Monitoring Using LDAP

iPlanet Directory Server supports a monitoring interface you can use to find out many useful bits of information about what the server is currently doing, how many connections it has, how many threads are working, etc. You can access the monitor feature by doing a search of `"cn=monitor"` with the filter `"(objectclass=*)"`. Additional entries are located below this entry.

Root DSE

The root DSE is the entry at the base of the tree that has no name. The attributes of this entry are defined in RFC 2251 *Lightweight Directory Access Protocol (v3)* and describe the configuration and capabilities of the server itself.

The following attributes of the root DSE are of interest to client writers:

`namingContexts`

Each value of this attribute specifies the suffix of a database.

`changelog`

This attribute is present if the change log is defined.

`subschemaSubentry`

The value of this attribute is the location of the subschema entry.

`supportedExtension`

This attribute lists the protocol extensions known to the server.

`supportedControl`

This attribute lists the controls implemented by the server.

`supportedSASLMechanisms`

This attribute lists the SASL mechanisms implemented by the server, including “DIGEST-MD5” and “EXTERNAL” for TLS.

Change Log

The change log allows specialized LDAPv3 clients to retrieve a listing of modifications which have been made to the portions of the directory tree held by the server. The change log is present if the `Retro Changelog` plug-in is enabled and the server restarted.

Entries are only added to the change log when changes are received over protocol. The change log is not supported on a replication consumer or in a multi-master deployment.

Subschema

As described in RFC 2251 *Lightweight Directory Access Protocol (v3)* and RFC 2252 *Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions*, the server constructs a subschema subentry automatically. This allows clients to determine the schema supported by the server.

In order for user-defined schema to be listed in the subschema subentry, the administrator must assign OBJECT IDENTIFIERs for all attribute types and object classes. This entry is read-only.

The location of the subschema subentry is given as a value of the `subschemaSubentry` attribute in the root DSE. The values of this attribute are distinguished names, such as "`cn=schema,dc=ldap,dc=innosoft,dc=com`". The subschema subentry can then be retrieved by performing a base object search of the entry by that name with filter "`(objectclass=*)`".

The subschema subentry provides the following attributes, whose syntaxes are defined in RFC 2252 *Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions*: `attributeTypes`, `objectClasses` and `matchingRules`.

Operational Attributes in Entries

In iPlanet Directory Server, the attributes `creatorsName`, `createTimestamp`, `modifiersName` and `modifyTimestamp` are maintained with each entry and held on disk.

Creator and Modifier's Names and Timestamps

When an entry is created using an `Add` operation or through `ldif2ldbm`, iPlanet Directory Server automatically includes the attributes `creatorsName` and `createTimestamp` in the entry. When an entry is modified, iPlanet Directory Server automatically includes the attributes `modifiersName` and `modifyTimestamp` in the entry.

dseType

This attribute is not provided by iPlanet Directory Server.

subschemaSubentry

The `subschemaSubentry` attribute is defined in X.501(1993). The value of the attribute is the Distinguished Name of the subschema entry, as described in the previous section.

modifyRights

The `modifyRights` attribute allows a client to determine whether it is permitted to remove the entry, or modify the attributes of an entry. This feature is not supported in iPlanet Directory Server.

fromEntry

The `fromEntry` attribute specifies whether the returned search result is taken from a master copy of the directory entry. This feature is not supported in iPlanet Directory Server.