Sun Java System Directory Server Enterprise Edition 6.2 Man Page Reference

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 820–2495
August 29, 2007

Contents

Administration Commands

Preface

A man page is intended to answer concisely the question “What does it do?” The man pages in general comprise a reference manual. Man pages are not intended to be a tutorial.

Overview

The following contains a brief description of each man page section and the information the section references.

- Section 1 describes, in alphabetical order, commands available for Directory Server Enterprise Edition.
- Section 1M describes, in alphabetical order, commands that are used chiefly for Directory Server Enterprise Edition maintenance and administration purposes.
- Section 4 outlines the formats of files delivered with Directory Server Enterprise Edition.
- Section 5dsconf describes Directory Server configuration properties. You modify these properties using the `dsconf` command.

  Section 5dsconf also describes legacy Directory Server configuration attributes. You modify these attributes using the `ldapmodify` command on the entries under `cn=config`.
- Section 5dpconf describes Directory Proxy Server configuration properties. You modify these properties using the `dpconf` command.
- Section 5dssd describes collections of LDAP schema objects that Directory Server provides. See `Intro`(5dssd) for an introduction to the LDAP schema reference documentation.
- Section 5dsat describes attribute types defined in the LDAP schema that Directory Server provides.
- Section 5dsoc describes object classes defined in the LDAP schema that Directory Server provides.

Below is a generic format for man pages. The man pages of each manual section generally follow this order, but include only needed headings. See `man(1)` for more information about man pages in general.

NAME

This section gives the names of the commands or functions documented, followed by a brief description of what they do.

SYNOPSIS

This section shows the syntax of commands or functions. When a command or file does not exist in the standard path, its full path name is shown. Options and arguments are alphabetized, with single letter arguments first, and options with arguments next, unless a different argument order is required.

The following special characters are used in this section:

[ ] Brackets. The option or argument enclosed in these brackets is optional. If the brackets are omitted, the argument must be specified.

`. . .` Ellipses. Several values can be provided for the previous argument, or the previous argument can be specified multiple times, for example, `"filename ...".`

| Separator. Only one of the arguments separated by this character can be specified at a time.

{ } Braces. The options and/or arguments enclosed within braces are interdependent, such that everything enclosed must be treated as a unit.

DESCRIPTION

This section defines the functionality and behavior of the service. Thus it describes concisely what the command does. It does not discuss OPTIONS or cite EXAMPLES. Interactive commands, subcommands, requests, macros, and functions are described under USAGE.

OPTIONS

This secton lists the command options with a concise summary of what each option does. The options are listed literally and in the order they appear in the SYNOPSIS section. Possible arguments to options are discussed under the option, and where appropriate, default values are supplied.

OPERANDS

This section lists the command operands and describes how they affect the actions of the command.

OUTPUT

This section describes the output – standard output, standard error, or output files – generated by the command.

RETURN VALUES

If the man page documents functions that return values, this section lists these values and describes the conditions under which they are returned. If a function can return only constant values, such as 0 or −1, these values are listed in tagged paragraphs. Otherwise, a single paragraph describes the return values of each function. Functions declared void do not return values, so they are not discussed in RETURN VALUES.

ERRORS

On failure, most functions place an error code in the global variable `errno` indicating why they failed. This section lists alphabetically all error codes a function can generate and describes the conditions that cause each error. When more than one condition can cause the same error, each condition is described in a separate paragraph under the error code.

EXAMPLES

This section provides examples of usage or of how to use a command or function. Wherever possible a complete example including command-line entry and machine response is shown. Whenever an example is given, the prompt is shown as `$`, or if the user must be superuser, `#`. Examples are followed by explanations, variable substitution rules, or returned values. Most examples illustrate concepts from the SYNOPSIS, DESCRIPTION, OPTIONS, and USAGE sections.

ENVIRONMENT VARIABLES

This section lists any environment variables that the command or function affects, followed by a brief description of the effect.

EXIT STATUS

This section lists the values the command returns to the calling program or shell and the conditions that cause these values to be returned. Usually, zero is returned for successful completion, and values other than zero for various error conditions.

FILES

This section lists all file names referred to by the man page, files of interest, and files created or required by commands. Each is followed by a descriptive summary or explanation.

ATTRIBUTES

This section lists characteristics of commands, utilities, and device drivers by defining the attribute type and its corresponding value. See `attributes`(5) for more information.

SEE ALSO — This section lists references to other man pages, in-house documentation, and outside publications.

DIAGNOSTICS — This section lists diagnostic messages with a brief explanation of the condition causing the error.

WARNINGS — This section lists warnings about special conditions which could seriously affect your working conditions. This is not a list of diagnostics.

NOTES — This section lists additional information that does not belong anywhere else on the page. It takes the form of an aside to the user, covering points of special interest. Critical information is never covered here.

BUGS — This section describes known bugs and, wherever possible, suggests workarounds.

Directory Server Enterprise Edition Documentation Set

This Directory Server Enterprise Edition documentation set explains how to use Sun Java System Directory Server Enterprise Edition to evaluate, design, deploy, and administer directory services. In addition, it shows how to develop client applications for Directory Server Enterprise Edition. The Directory Server Enterprise Edition documentation set is available at `http://docs.sun.com/coll/1224.3`.

For an introduction to Directory Server Enterprise Edition, review the following documents in the order in which they are listed.

TABLE P–1 Directory Server Enterprise Edition Documentation

Document Title	Contents
Sun Java System Directory Server Enterprise Edition 6.2 Release Notes	Contains the latest information about Directory Server Enterprise Edition, including known problems.
Sun Java System Directory Server Enterprise Edition 6.2 Evaluation Guide	Introduces the key features of this release. Demonstrates how these features work and what they offer in the context of a fictional deployment that you can implement on a single system.
Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide	Explains how to plan and design highly available, highly scalable directory services based on Directory Server Enterprise Edition. Presents the basic concepts and principles of deployment planning and design. Discusses the solution life cycle, and provides high-level examples and strategies to use when planning solutions based on Directory Server Enterprise Edition.

TABLE P–1 Directory Server Enterprise Edition Documentation *(Continued)*

Document Title	Contents
Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide	Explains how to install the Directory Server Enterprise Edition software. Shows how to select which components to install, configure those components after installation, and verify that the configured components function properly. For instructions on installing Directory Editor, go to `http://docs.sun.com/coll/DirEdit_05q1`. Make sure you read the information in *Sun Java System Directory Server Enterprise Edition 6.2 Release Notes* concerning Directory Editor before you install Directory Editor.
Sun Java System Directory Server Enterprise Edition 6.2 Migration Guide	Provides instructions for upgrading components from earlier versions of Directory Server, Directory Proxy Server, and Identity Synchronization for Windows.
Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide	Provides command-line instructions for administering Directory Server Enterprise Edition. For hints and instructions on using the Directory Service Control Center, DSCC, to administer Directory Server Enterprise Edition, see the online help provided in DSCC. For instructions on administering Directory Editor, go to `http://docs.sun.com/coll/DirEdit_05q1`. For instructions on installing and configuring Identity Synchronization for Windows, see Part II, “Installing Identity Synchronization for Windows,” in *Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide*.
Sun Java System Directory Server Enterprise Edition 6.2 Developer's Guide	Shows how to develop directory client applications with the tools and APIs that are provided as part of Directory Server Enterprise Edition.
Sun Java System Directory Server Enterprise Edition 6.2 Reference	Introduces the technical and conceptual foundations of Directory Server Enterprise Edition. Describes its components, architecture, processes, and features. Also provides a reference to the developer APIs.
Sun Java System Directory Server Enterprise Edition 6.2 Man Page Reference	Describes the command-line tools, schema objects, and other public interfaces that are available through Directory Server Enterprise Edition. Individual sections of this document can be installed as online manual pages.
Sun Java System Directory Server Enterprise Edition 6.2 Troubleshooting Guide	Provides information for defining the scope of the problem, gathering data, and troubleshooting the problem areas using various tools.
Sun Java System Identity Synchronization for Windows 6.0 Deployment Planning Guide	Provides general guidelines and best practices for planning and deploying Identity Synchronization for Windows

Related Reading

The SLAMD Distributed Load Generation Engine (SLAMD) is a Java^{TM} application that is designed to stress test and analyze the performance of network-based applications. It was originally developed by Sun Microsystems, Inc. to benchmark and analyze the performance of LDAP directory servers. SLAMD is available as an open source application under the Sun Public License, an OSI-approved open source license. To obtain information about SLAMD, go to `http://www.slamd.com/`. SLAMD is also available as a java.net project. See `https://slamd.dev.java.net/`.

Java Naming and Directory Interface (JNDI) technology supports accessing the Directory Server using LDAP and DSML v2 from Java applications. For information about JNDI, see `http://java.sun.com/products/jndi/`. The *JNDI Tutorial* contains detailed descriptions and examples of how to use JNDI. this tutorial is at `http://java.sun.com/products/jndi/tutorial/`.

Directory Server Enterprise Edition can be purchased as a standalone product or as a component of Sun Java Enterprise System. Java Enterprise System is a software infrastructure that supports enterprise applications distributed across a network or Internet environment. If Directory Server Enterprise Edition was purchased as a component of Java Enterprise System, you should be familiar with the system documentation at `http://docs.sun.com/coll/1286.3`.

Redistributable Files

Directory Server Enterprise Edition does not provide any files that you can redistribute.

Default Paths and Command Locations

This section explains the default paths used in the documentation, and gives the locations of commands on different operating systems and deployment types.

Default Paths

The following table describes the default paths that are used in this book.

TABLE P–2 Default Paths

Placeholder	Description	Default Value
install-path	Represents the base installation directory for Directory Server Enterprise Edition software. The software is installed in directories below this base *install-path*. For example, Directory Server software is installed in *install-path*`/ds6/`.	When you install from a zip distribution using dsee_deploy(1M), the default *install-path* is the current directory. You can set the *install-path* using the `-i` option of the `dsee_deploy` command. When you install from a native package distribution, such as you would using the Java Enterprise System installer, the default *install-path* is one of the following locations: ■ Solaris systems - `/opt/SUNWdsee/`. ■ Red Hat systems - `/opt/sun/`. ■ Windows systems - `C:\Program Files\Sun\JavaES5\DSEE`.
instance-path	Represents the full path to an instance of Directory Server or Directory Proxy Server. The documentation uses `/local/ds/` for Directory Server and `/local/dps/` for Directory Proxy Server.	No default path exists. Instance paths must nevertheless always be found on a *local* file system. The following directories are recommended: `/var` on Solaris systems `/global` if you are using Sun Cluster
serverroot	Represents the parent directory of the Identity Synchronization for Windows installation location	Depends on your installation. Note the concept of a *serverroot* no longer exists for Directory Server.
`isw-`*hostname*	Represents the Identity Synchronization for Windows instance directory	Depends on your installation
`/`*path*`/`*to*`/cert8.db`	Represents the default path and file name of the client's certificate database for Identity Synchronization for Windows	*current-working-dir*`/cert8.db`
serverroot`/isw-`*hostname*`/ logs/`	Represents the default path to the Identity Synchronization for Windows local logs for the System Manager, each connector, and the Central Logger	Depends on your installation
serverroot`/isw-`*hostname*`/ logs/central/`	Represents the default path to the Identity Synchronization for Windows central logs	Depends on your installation

Command Locations

To know more about each of the commands, see the relevant man pages. For full descriptions of the files installed, see also the following documentation.

- Chapter 14, “Directory Server File Reference,” in *Sun Java System Directory Server Enterprise Edition 6.2 Reference* covers files installed with Directory Server, and files created for server instances.
- Chapter 25, “Directory Proxy Server File Reference,” in *Sun Java System Directory Server Enterprise Edition 6.2 Reference* covers files installed with Directory Proxy Server, and files created for server instances.
- Appendix A, “Directory Server Resource Kit File Reference,” in *Sun Java System Directory Server Enterprise Edition 6.2 Reference* covers files installed with Directory Server Resource Kit.

REFERENCE

User Commands

Name authrate – measure rate of authentication to an LDAP directory

Synopsis

```
install-path/dsrk6/bin/authrate
 [options]
```

Description The `authrate` command measures the rate at which a given bind DN can authenticate to an LDAP directory. As with all measures of performance, results depend on many factors, including what options you pass to the `authrate` command, and also how the directory service itself is tuned.

The command uses LDAP v3, and cannot be used to authenticate to an LDAP v2 directory not supporting LDAP v3.

Options The `authrate` command supports the following options:

`-C` *messages* Display the specified number of results messages before exiting. Results messages appear by default as output on standard out, similar to the following:

```
Avg r=2584.00/thr (516.80/sec), total=  7752
```

This shows output for three threads authenticating for five seconds. The average bind rate per thread is 516.80 per thread per second for the interval measured. The total shown for all threads is 7752.

Default is to continue iterating until the command is interrupted.

`-D` *bindDN* Use the specified bind DN to authenticate to the directory.

If the bind DN is not specified, the `authrate` command attempts anonymous authentication.

`-h` *hostname* Connect to the directory on the specified host.

Enclose IPv6 addresses in brackets (`[]`) as described in RFC 2732.

Default is to connect to the local host on the loopback address, `127.0.0.1`.

`-i` *filename* Use the file specified to read bind DNs and passwords at random.

Refer to Random Bind DN Syntax and Random Bind DN Substitution for details.

`-j` *seconds* Display results each specified number of seconds.

Default is to display results every 5 seconds.

`-k` Keep connections open, measuring only the time required to perform the bind operation.

Default is to measure both the bind and unbind time as part of the authentication sequence.

`-m` *maxAuth* Perform no more than the specified number of binds per thread.

Default is for each thread to continue iterating until the command is interrupted.

`-p` *port* Connect to the directory on the specified port.

Default is to connect to the default simple authentication port for LDAP, `389`.

`-q` Run in quiet mode, not displaying results.

Default is to display results every 5 seconds, which you can adjust using the `-j` option.

`-r` *maxRand* Use the specified maximum to determine the range for random numbers replacing `%d` formatting specifications when authenticating with random bind DNs and passwords.

When you use this option twice, the first occurrence generates random numbers in the range [0,*maxRand1*–1] for the first `%d`, the second [1,*maxRand2*] for the second `%d`.

`-S` *randSeed* Use the specified seed, an `unsigned int`, for random number generation.

Default seed is `0`.

`-t` *threads* Use the specified number of the threads to connect to the server.

Default is to use one thread.

`-u` Do not unbind as part of the authentication sequence.

Default is to unbind as part of the authentication sequence.

`-v` Display verbose output.

`-W` *filename* Read the bind password from the specified file.

`-w` *password* Use the specified bind password to authenticate to the directory.

`-w –` Prompt for the bind password so it does not appear on the command line or in a file.

Extended Description

The `authrate` command repeatedly initializes a connection and binds to a directory server, without performing any other operation. Threads may be configured to keep open connections and perform LDAP binds repeatedly. The command-line options let you specify the bind credentials.

The command uses LDAP v3, and cannot be used to authenticate to an LDAP v2 directory not supporting LDAP v3. Furthermore, the `authrate` command uses simple authentication, not secure binding.

By default, the `authrate` command attempts to bind indefinitely, displaying results periodically, and displaying any errors encountered as well without interrupting operation.

To simulate real use conditions and reduce any artifacts due to the repetitive nature of the tests, the `authrate` command provides a mechanism for generating a random bind DN for authentication.

Random Bind DN Syntax

Include randomly generated numbers by specifying `%d` and `%s` placeholders in the bind DN and the bind password. These placeholders are then replaced according to the following rules:

`%d` Replace this placeholder with random integer values depending on the *maxRand* parameter to the `-r` option.

The `-r` option may be used at most two times to generate random bind DNs. When used in the bind DN, replacement values for the `%d` placeholder range over [0,*maxRand1*-1] for the first use of the `-r` option, and over [1,*maxRand2*] for the second.

The `%d` may be used up to eight times to generate a random password. When used in the bind password, replacement values for the `%d` placeholder range over [0,*maxRand1*-1] for each use of the `-r` option.

When the the number of `%d` placeholders exceeds the number of `-r` options, only one value for each use of the `-r` option is generated. Each `%d` placeholder is replaced with a generated value.

`%s` Replace this placeholder with random strings from the file specified using the `-i` option.

Replacement values for this placeholder are randomly selected lines of the file specified.

Random Bind DN Substitution

The `authrate` command requires that you apply the following rules for substitutions, displaying an error message when the used incorrectly:

- Use only one type of placeholder, either `%d` or `%s`, per invocation of the `authrate` command.
- Use `%%d` and `%%s` to specify literal strings `%d` and `%s`, respectively.

In order to use this random authentication mechanism, you must populate your directory accordingly. For example, you can measure the authentication rate using the following command:

```
$ authrate -D "uid=test%d,ou=test,dc=example,dc=com" -w "auth%d%d" -r 100
```

In order for the authrate command to bind effectively, your directory must contain entries corresponding to the following LDIF excerpt:

```
dn: uid=test0,ou=test,dc=example,dc=com
userPassword: auth00

dn: uid=test1,ou=test,dc=example,dc=com
userPassword: auth11

dn: uid=test2,ou=test,dc=example,dc=com
userPassword: auth22
...

dn: uid=test10,ou=test,dc=example,dc=com
userPassword: auth1010
...

dn: uid=test99,ou=test,dc=example,dc=com
userPassword: auth9999
```

Examples

Examples in this section use the following conventions:

- The authrate command is found in a directory present in the PATH used for the examples.
- The directory server is located on a system named host.
- The directory has been configured to support anonymous access for search and read. Therefore, you do not have to specify bind information.
- The directory server listens on port 389, the default for non-SSL connections.

EXAMPLE 1 authrate: Sample Output

The following command performs anonymous binds until it has displayed five results messages. Notice that each line concerns only the elapsed interval.

```
$ authrate -C 5
Avg r=1952.00/thr (390.40/sec), total=  1952
Avg r=1937.00/thr (387.40/sec), total=  1937
Avg r=1938.00/thr (387.60/sec), total=  1938
Avg r=1921.00/thr (384.20/sec), total=  1921
Avg r=1921.00/thr (384.20/sec), total=  1921
All threads exited
```

Notice also that a result message provides the following items of information:

- The average rate of authentication per thread of execution
- The average rate of authentication per second
- The total number of authentication operations performed during the interval the results message concerns

EXAMPLE 2 `authrate`: Two Threads

The following command performs anonymous binds until it has displayed five results messages, using three threads to bind. Notice that each line concerns only the elapsed interval.

```
$ authrate -C 5 -t 3
Avg r= 300.00/thr (180.00/sec), total=   900
Avg r= 300.00/thr (180.00/sec), total=   900
Avg r= 299.67/thr (179.80/sec), total=   899
Avg r= 298.00/thr (178.80/sec), total=   894
Avg r= 299.33/thr (179.60/sec), total=   898
All threads exited
```

Here the average per thread, approximate 300 binds, is shown for each interval of three seconds. The averages given in parentheses, approximately 180 per second, represent the average bind rate over the interval. The totals shown represent the total number of binds over the interval.

EXAMPLE 3 `authrate`: Full Authentication Rate

The following command applies the mechanism described in Random Bind DN Substitution, performing full authentication (open, bind, unbind, close) with randomly generated bind DNs and passwords.

```
$ authrate -D "uid=test%d,ou=test,dc=example,dc=com" -w "auth%d%d" -r 100 -C 5
Avg r=1301.00/thr (260.20/sec), total=  1301
Avg r=1307.00/thr (261.40/sec), total=  1307
Avg r=1281.00/thr (256.20/sec), total=  1281
Avg r=1316.00/thr (263.20/sec), total=  1316
Avg r=1313.00/thr (262.60/sec), total=  1313
All threads exited
```

EXAMPLE 4 `authrate`: Bind Rate Alone

The following command applies the mechanism described in Random Bind DN Substitution, keeping the connection open and binding repeatedly with randomly generated bind DNs and passwords.

```
$ authrate -D "uid=test%d,ou=test,dc=example,dc=com" -w "auth%d%d" -r 100 -k -C 5
Avg r=2584.00/thr (516.80/sec), total=  2584
Avg r=2603.00/thr (520.60/sec), total=  2603
Avg r=2592.00/thr (518.40/sec), total=  2592
Avg r=2613.00/thr (522.60/sec), total=  2613
Avg r=2560.00/thr (512.00/sec), total=  2560
All threads exited
```

Exit Status

The `authrate` command returns the following exit status codes.

`0`	Successful completion.
non-zero	An error occurred.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	Zip distribution only
Stability Level	Evolving

See Also

makeldif(1), modrate(1), searchrate(1)

Name dsmlmodify – add, modify, rename, move, or delete directory entries

Synopsis *install-path*`/dsrk6/bin/dsmlmodify`
`-h` *hostURL* `[`*options*`] -f` *filename*

Description The `dsmlmodify` command requests the addition, modification, rename, move, or deletion of entries stored in a directory accessible through Directory Services Markup Language (DSML) v2.

You must specify additions and modifications in the proper order, because the directory performs the updates in the order you request them. For example, to add entries to a subtree that does not yet exist, you must first update the base entry at the root of the subtree before adding entries under the base entry.

Options The `dsmlmodify` command supports the following options:

`-D` *user-identifier*

Use the specified user identifier to authenticate.

The user identifier is the HTTP-layer identifier. The HTTP-layer identifier is typically mapped to an account in the directory. For example, if the `uid` value is used for HTTP-layer authentication, which maps in the directory to bind DN `dn:uid=`*user-identifier*`,ou=people,dc=example,dc=com`, then the `dsmlmodify -D bjensen` command would end up using permissions for directory operations based on the permissions for the account with entry DN `uid=bjensen,ou=people,dc=example,dc=com`. The *user-identifier* thus depends closely on the identity mapping between the HTTP layer and the LDAP layer.

If the user identifier and its password are omitted, the `dsmlmodify` command binds anonymously. The user identifier determines what entries and attributes the user can modify, according to the permissions for the user.

`-f` *filename*

Read the modifications from a file using DSML syntax.

The following content for example allows modification of Barbara Jensen's password:

```
<modifyRequest dn="uid=bjensen,ou=people,dc=example,dc=com">
<modification name="userpassword" operation="replace">
<value>newpassword</value>
</modification>
</modifyRequest>
```

`-h` *hostURL*

Use the specified URL to access the directory.

The host URL takes the form `http://`*host*`:`*port* where *host* represents the host on which the directory runs, and *port* is the port on which the directory listens for DSML requests.

`-j` *filename* Read the bind password for simple HTTP authentication from the specified file.

`-w –` Prompt for the bind password for simple HTTP authentication.

`-w` *password* Use the specified bind password for simple HTTP authentication.

Examples Examples in this section use the following conventions:

- The `dsmlmodify` command is found in a directory present in the `PATH` used for the examples.
- The directory server is located on a system named `host`.
- The directory server listens for DSML requests over HTTP on port 8080.

EXAMPLE 1 `dsmlmodify`: Adding an Entry

The following commands demonstrate adding an entry:

```
$ cat add.dsml
<addRequest dn="uid=ajohnson,ou=people,dc=example,dc=com">
   <attr name="objectclass"><value>top</value></attr>
   <attr name="objectclass"><value>person</value></attr>
   <attr name="objectclass"><value>organizationalPerson</value></attr>
   <attr name="objectclass"><value>inetOrgPerson</value></attr>
   <attr name="uid"><value>ajohnson</value></attr>
   <attr name="sn"><value>Johnson</value></attr>
   <attr name="cn"><value>Alice</value></attr>
   <attr name="mail"><value>alice.johnson@example.com</value></attr>
   <attr name="userPassword"><value>weakness</value></attr>
</addRequest>
$ dsmlmodify -h http://host:8080 -D hmiller -w - -f add.dsml
Enter bind password:
...
```

If you read `Example.ldif`, you see that `hmiller`'s password is `hillock`.

EXAMPLE 2 `dsmlmodify`: Modifying an Entry

The following commands demonstrate modifying an entry:

```
$ cat mod.dsml
<modifyRequest dn="uid=bjensen,ou=people,dc=example,dc=com">
<modification name="userpassword" operation="replace">
<value>newpassword</value>
```

EXAMPLE 2 `dsmlmodify`: Modifying an Entry *(Continued)*

```
</modification>
</modifyRequest>
$ dsmlmodify -h http://host:8080 -D bjensen -w - -f mod.dsml
Enter bind password:
...
```

If you read `Example.ldif`, you see that the `bjensen`'s password is `hifalutin`.

EXAMPLE 3 `dsmlmodify`: Deleting an Entry

The following commands demonstrate deleting an entry:

```
$ cat del.dsml
<delRequest dn="uid=ajohnson,ou=people,dc=example,dc=com" />
$ dsmlmodify -h http://host:8080 -D hmiller -w - -f del.dsml
Enter bind password:
...
```

If you read `Example.ldif`, you see that `hmiller`'s password is `hillock`.

EXAMPLE 4 `dsmlmodify`: Renaming an Entry

The following commands demonstrate renaming an entry:

```
$ cat rdn.dsml
<modDNRequest
  dn="uid=ajohnson,ou=people,dc=example,dc=com"
  newrdn="uid=aweiss"
  deleteoldrdn="true"
  newSuperior="ou=people,dc=example,dc=com"/>
$ dsmlmodify -h http://host:8080 -D hmiller -w - -f rdn.dsml
Enter bind password:
...
```

If you read `Example.ldif`, you see that `hmiller`'s password is `hillock`.

Exit Status

Exit status values are returned as part of the response, including both the code and the description as described in the DSML v2 standard. Common exit status codes follow:

`0` Successful completion; `success`.

`1` Server encountered errors while processing the request; `operationsError`.

`2` Server encountered errors while processing the request; `protocolError`.

10 Base DN belongs to an entry handled by neither server, and the referral URL identifies another server that handles the entry; `referral`.

16 Attribute to be modified does not exist; `noSuchAttribute`.

19 Attribute modification requested is not a proper modification. For example, a requested change to `userpassword` would result in a user password shorter than the minimum length allowed; `constraintViolation`.

20 Attribute to add already exists with specified value; `attributeOrValueExists`.

21 In response to a request to modify directory schema, the requested modification includes no object class or attribute type specification; `invalidAttributeSyntax`.

32 Base DN belongs to an entry handled by neither server, and no referral URL is available for the entry; `noSuchObject`.

50 Bind DN user does not have permission to read the entry from the directory; `insufficientAccessRights`.

53 Directory is read-only; `unwillingToPerform`.

65 Requested modification would cause the entry not to comply with the schema; `objectClassViolation`.

67 Requested modification would cause the entry to be missing attributes that are components of the entry DN; `notAllowedOnRDN`.

68 An entry already exists with the same DN as the entry to add; `entryAlreadyExists`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	Zip distribution only
Stability Level	Evolving

See Also

dsmlsearch(1), `ldap_error`(3LDAP)

Name dsmlsearch – find directory entries

Synopsis

```
install-path/dsrk6/bin/dsmlsearch
-h hostURL -b baseDN [options] [attribute]...
```

Description The `dsmlsearch` command searches for entries stored in a directory accessible through Directory Services Markup Language (DSML) v2, and displays the results in DSML format, including the specified attributes or all attributes returned if none are specified.

Filter files contain filters in DSML format. The `dsmlsearch` command does not support LDAP-style filters.

Options The `dsmlsearch` command supports the following options:

`-a` *deref*

Dereference aliases as specified during a search. Possible values for the *deref* argument include:

`derefAlways`	Dereference aliases both when finding the base DN, and when searching below it.
`derefFindingBaseObj`	Dereference aliases when finding the base DN.
`neverDerefAliases`	Never dereference aliases (default).

This option has no effect when used with directories that do not support alias dereferencing.

`-b` *baseDN*

Use the entry with the specified distinguished name (DN) as the base entry for the search scope.

`-D` *user-identifier*

Use the specified user identifier to authenticate.

The user identifier is the HTTP-layer identifier. The HTTP-layer identifier is typically mapped to an account in the directory. For example, if the `uid` value is used for HTTP-layer authentication, which maps in the directory to bind DN `dn:uid=`*user-identifier*`,ou=people,dc=example,dc=com`, then the `dsmlsearch -D bjensen` command would end up using permissions for directory operations based on the permissions for the account with entry DN `uid=bjensen,ou=people,dc=example,dc=com`. The *user-identifier* thus depends closely on the identity mapping between the HTTP layer and the LDAP layer.

If the user identifier and its password are omitted, the `dsmlsearch` command binds anonymously. The user identifier determines what entries and attributes the user can read, according to the permissions for the user.

`-f` *filename*

Read the search filter or filters from the specified file.

`-h` *hostURL*

Use the specified URL to access the directory.

The host URL takes the form `http://`*host*`:`*port* where *host* represents the host on which the directory runs, and *port* is the port on which the directory listens for DSML requests.

`-j` *filename*

Read the bind password for simple HTTP authentication from the specified file.

`-l` *timelimit*

Interrupt the search if the time limit specified in seconds is exceeded.

`-s` *scope*

Use the specified search scope.

The following values are supported for *scope*:

`baseObject`

Examine only the entry specified by the argument to the `-b` option.

`singleLevel`

Examine only to the entry specified by the argument to the `-b` option and its immediate children.

`wholeSubtree`

(Default) Examine the subtree whose root is the entry specified by the argument to the `-b` option.

`-w –`

Prompt for the bind password for simple HTTP authentication.

`-w` *password*

Use the specified bind password for simple HTTP authentication.

`-z` *maxEntries*

Return no more than the specified number of entries.

Examples

Examples in this section use the following conventions:

- The `dsmlsearch` command is found in a directory present in the `PATH` used for the examples.
- The directory server is located on a system named `host`.
- The directory has been configured to support anonymous access for search and read. Therefore, you do not have to specify bind information.
- The directory server listens for DSML requests over HTTP on port 8080.

EXAMPLE 1 `dsmlsearch`: Returning All Entries

The following command returns all entries in the suffix under the base DN. Use this only when you need to retrieve all entries and attributes:

```
$ cat filter
<filter>
<present name="objectclass"/>
</filter>
$ dsmlsearch -h http://host:8080 -b dc=example,dc=com -f filter
```

EXAMPLE 2 `dsmlsearch`: Narrowing a Search

The following command employs a more specific filter to narrow the search:

```
$ cat filter
<filter>
<equalityMatch name="uid">
<value>bjensen</value>
</equalityMatch>
</filter>
$ dsmlsearch -h http://host:8080 -b dc=example,dc=com -f filter
```

EXAMPLE 3 `dsmlsearch`: Searching the Root DSE

The following command searches the root DSE entry, which contains the list of suffixes supported by the directory and potentially other information. Notice you specify the scope as only the base entry:

```
$ cat filter
<filter>
<present name="objectclass"/>
</filter>
$ dsmlsearch -h http://host:8080 -b "" -s baseObject -f filter
```

EXAMPLE 4 `dsmlsearch`: Searching the Schema Entry

The following command searches the schema entry, which contains the directory schema. Notice you specify the scope as only the base entry:

```
$ cat filter
<filter>
<present name="objectclass"/>
</filter>
$ dsmlsearch -h http://host:8080 -b cn=schema -s baseObject -f filter
```

EXAMPLE 5 `dsmlsearch`: Filter Examples

The following list shows LDAP search filters with corresponding DSML search filters.

LDAP filter: `(cn=Barbara Francis)`

DSML filter:

```
<filter>
<equalityMatch name="cn">
<value>Barbara Francis</value>
</equalityMatch>
</filter>
```

EXAMPLE 5 `dsmlsearch`: Filter Examples *(Continued)*

LDAP filter: `(cn=*Barb*)`

DSML filter:

```
<filter>
<substrings name="cn">
<any>Barb</any>
</substrings>
</filter>
```

LDAP filter: `(cn~=Barbare)`

DSML filter:

```
<filter>
<approxMatch name="cn">
<value>Barbare</value>
</approxMatch>
</filter>
```

LDAP filter: `(!(cn=*Barbara*))`

DSML filter:

```
<filter>
 <not>
  <substrings name="cn">
   <any>Barbara</any>
  </substrings>
 </not>
</filter>
```

LDAP filter: `(&(cn=*Barbara*)(cn=*Francis*))`

DSML filter:

```
<filter>
 <and>
  <substrings name="cn">
   <any>Barbara</any>
  </substrings>
  <substrings name="cn">
   <any>Francis</any>
  </substrings>
 </and>
</filter>
```

LDAP filter: `(|(cn=*Barbara*)(cn=*Jensen*))`

DSML filter:

```
<filter>
 <or>
  <substrings name="cn">
   <any>Barbara</any>
  </substrings>
  <substrings name="cn">
   <any>Jensen</any>
  </substrings>
```

EXAMPLE 5 `dsmlsearch`: Filter Examples *(Continued)*

```
                                                   </or>
                                                  </filter>
```

Exit Status

Exit status values are returned as part of the response, including both the code and the description as described in the DSML v2 standard. Common exit status codes follow:

`0` Successful completion; `success`.

`1` Server encountered errors while processing the request; `operationsError`.

`2` Server encountered errors while processing the request; `protocolError`.

`3` Search exceeded the time limit for operations on the server; `timeLimitExceeded`.

`4` Search returned more results than the maximum number allowed by the server; `sizeLimitExceeded`.

`10` Base DN belongs to an entry handled by neither server, and the referral URL identifies another server that handles the entry; `referral`.

`11` Search returned more results than the maximum number a client application is allowed by the server to retrieve; `adminLimitExceeded`.

`32` Base DN belongs to an entry handled by neither server, and no referral URL is available for the entry; `noSuchObject`.

`50` Bind DN user does not have permission to read the entry from the directory; `insufficientAccessRights`.

`53` Directory is read-only; `unwillingToPerform`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	Zip distribution only
Stability Level	Evolving

See Also

dsmlmodify(1), `ldap_error`(3LDAP)

Name entrycmp – compare the same entry on two or more different servers

Synopsis

```
install-path/ds6/bin/entrycmp
 [-D bindDN] [-w password] [-n] [-p port] [-j file] [-T timeout]
 [-J file] [-W keypassword] [-K keydbpath] [-N certname]
 [-P certdbpath] [-e SSL port]  ServerSpec  entryDN
```

Description The `entrycmp` command compares the same entry on two or more different servers. An entry is retrieved from the master and the `nsuniqueid` of the entry is used to retrieve the same entry from a specified consumer. All the attributes and values of the two entries are compared. If they are identical, the entries are considered to be the same.

Options The following options are supported:

`-D` The distinguished name with which to bind to the server. This parameter is optional if the server is configured to support anonymous access. If a DN is specified in the *ServerSpec*, this overrides the `-D` option.

`-j` If specifying the default password at the command-line poses a security risk, the password can be stored in a file. The `-j` option specifies this file.

`-n` Specifies that `entrycmp` should not run in interactive mode. Running in interactive mode allows you to re-enter the bindDN, password, host and port, if a bind error occurs.

`-p` The TCP port used by Directory Server. The default port is 389. If a port is specified in the *ServerSpec*, this overrides the `-p` option.

`-T` Specifies the number of seconds after which `entrycmp` will time out if the server connection goes down.

`-w` The password associated with the distinguished name specified by the `-D` option. If a password is specified in the *ServerSpec*, this overrides the `-w` option.

entryDN The DN of the entry that you wish to compare.

ServerSpec The server specification. The server specification is of one of the following forms.

```
-s|-S HostSpec [-c|-C HostSpec ...]

-c|-C HostSpec [-s|-S HostSpec ...]
```

Here, `-s` refers to the supplier replica. `-c` refers to the consumer replica. Lower case specifies non-SSL options. Upper case specifies SSL options.

Host Spec The host specification, which takes the form `[bindDN:[password]]@]`*host*`[:port]`. The following is an example:

```
cn=admin,cn=Administrators,cn=config:mypword@myserver:1389
```

If you are using SSL, use `-S` and `-C` in the server specification. In this case, *HostSpec* specifies the certificate name and key password, rather than the bindDN and password. Specifying both more than one `-s`, and also more than one `-c` generates an error. If no `-c` option is specified, the `-s` *HostSpec* may refer to any server, either a consumer or a supplier.

Ssl Options

You can use the following options to specify that `entrycmp` uses LDAPS when communicating with the Directory Server. You can also use these options if you want to use certificate-based authentication. These options are valid only when LDAPS has been turned on and configured.

`-e` Default SSL port, 636.

`-J` This option has the same function as the `-j` option, for the key password.

`-K` Specifies the name of the certificate key used for certificate-based client authentication. For example, `-K` *Server-Key*.

`-N` Specifies the certificate name to use for certificate-based client authentication. For example, `- N` *Server-Cert*. If this option is specified, the `-W` option is required.

`-P` Specifies the location of the certificate database.

`-W` Specifies the password for the certificate database identified by the `-P` option. For example, `-W` *serverpassword*.

Examples

EXAMPLE 1 Specifying an entry DN

```
$ entrycmp -D cn=admin,cn=Administrators,cn=config -w mypword \
 -s myserver:1389 "uid=csmith,ou=people,dc=example,dc=com"
```

Exit Status

The following exit values are returned:

`0` Successful completion, that is a match was found.

`1` An error occurred, and no match was found.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Stable

See Also

`insync`(1), `repldisc`(1)

Notes

The node on which you are running the `entrycmp`, `insync`, and `repldisc` tools must be able to reach all the specified hosts. If these hosts are unavailable due to a firewall, VPN, or other network setup reasons, you will encounter difficulties using these tools. For the same reason ensure that all servers are up and running before using these tools.

When identifying hosts, you must use either symbolic names or IP addresses for all hosts since the replication monitoring commands do not address resolution between symbolic names and IP addresses. Using a combination of the two can cause problems. Moreover, on multi-homed hosts, referring to the same Directory Server instance using different names may cause unexpected results.

When SSL is enabled, the directory server on which you are running the tools must have a copy of all the certificates used by the other servers in the topology.

The replication monitoring tools rely on access to `cn=config` to obtain the replication status. This should be taken into account particularly when replication is configured over SSL.

Name

fildif – creates a filtered version of an LDIF input file

Synopsis

install-path`/ds6/bin/fildif`
`-i` *input-file* `[-o` *output-file*`] [-f ] -b` *repl-agmt-dn*
 `-p` *instance-path*

Description

The `fildif` command creates a filtered version of an LDIF input file. `fildif` takes a configuration file as an input parameter. This configuration file must conform to the configuration rules of the filtering service included as part of Directory Server, and must contain the specific set and element entries that define these rules. The configuration rules can be defined by using Directory Service Control Center or at the command-line.

`fildif` does not require the Directory Server instance to be running.

A filtering service configuration is accessed through a replication agreement. The replication agreement entry DN is provided to `fildif` with the `-b` option.

Options

The following options are supported:

`-b` *repl-agmt-dn*
The DN of the replication agreement used as the filtering service configuration entry point. The entry specified must exist in the configuration of the Directory Server instance.

`-f`
Force `fildif` to overwrite the contents of the specified output file, if it exists.

`-i` *input-file*
The input LDIF file whose contents are filtered.

`-o` *output-file*
The output LDIF file in which the filtered results are stored. If no output file is specified, the default output file is `./output.ldif`.

`-p`
The full path to the Directory Server instance whose configuration contains the replication agreement specified as a parameter of the `-b` option.

Examples

EXAMPLE 1 Using All Options

The following example shows the `fildif` command to generate an output file `filt_data.ldif` that overwrites the file if it exists already.

```
$ fildif -i data.ldif -o filt_data.ldif -f \
-b "cn=ds.example.com:389,cn=replica,cn=dc=example\,dc=com,\
cn=mapping tree,cn=config"
-p /local/ds
```

Exit Status

The following exit values are returned:

`0` Successful completion.

`1` An error occurred.

On error, verbose error messages are output to standard output.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Stable

See Also

insync(1), entrycmp(1)

Name insync – indicate the synchronization state between a supplier replica and one or more consumer replicas

Synopsis *install-path*`/ds6/bin/insync`

```
 [-D bindDN] [-w password] [-t] [-n] [-d] [-j file] [-p port]
 [-T timeout] [-J file] [-W keypassword] [-K keydbpath]
 [-N certname] [-P certdbpath] [-e SSL port] [-b ReplicaRoot]
  ServerSpec [interval]
```

Description The `insync` command indicates the synchronization state between a supplier replica and one or more consumer replicas. `insync` compares the RUVs of replicas and displays the time difference or delay (in seconds) between the servers.

Options The following options are supported:

`-b` The suffix (replica root) that has been specified for replication. If `-b` is not specified, the delay for all suffixes is displayed.

`-d` Displays the date of the last change recorded on the master. Using the `-d` option twice (`-d -d`) displays the time difference (in days, minutes and seconds) between the time of the last change and the current time.

`-D` The distinguished name with which to bind to the server. This parameter is optional if the server is configured to support anonymous access. If a DN is specified in the *ServerSpec*, this overrides the `-D` option.

`-j` If specifying the default password at the command-line poses a security risk, the password can be stored in a file. The `-j` option specifies this file.

`-n` Specifies that `insync` should not run in interactive mode. Running in interactive mode allows you to re-enter the bindDN, password, host and port, if a bind error occurs.

`-p` The TCP port used by Directory Server. The default port is 389. If a port is specified in the *ServerSpec*, this overrides the `-p` option.

`-t` Displays the mode of transport (SSL or CLEAR)

`-T` Specifies the number of seconds after which `insync` will time out if the server connection goes down.

`-w` The password associated with the distinguished name specified by the `-D` option. If a password is specified in the *ServerSpec*, this overrides the `-w` option.

ServerSpec The server specification. The server specification is of one of the following forms.

```
-s|-S HostSpec [-c|-C HostSpec ...]

-c|-C HostSpec [-s|-S HostSpec ...]
```

Here, `-s` refers to the supplier replica. `-c` refers to the consumer replica. Lower case specifies non-SSL options. Upper case specifies SSL options.

Host Spec The host specification, which takes the form `[bindDN:[password]]@]`*host*`[:port]`. The following is an example:

```
cn=admin,cn=Administrators,cn=config:mypword@myserver:1389
```

If you are using SSL, use `-S` and `-C` in the server specification. In this case, *HostSpec* specifies the certificate name and key password, rather than the bindDN and password. Specifying both more than one `-s`, and also more than one `-c` generates an error. If no `-c` option is specified, the `-s` *HostSpec* may refer to any server, either a consumer or a supplier.

interval The amount of time (in seconds) after which the synchronization query will start again (in an infinite loop). If no interval is specified, the synchronization query will run only once.

Ssl Options

You can use the following options to specify that `insync` uses LDAPS when communicating with the Directory Server. You can also use these options if you want to use certificate-based authentication. These options are valid only when LDAPS has been turned on and configured.

`-e` Default SSL port, 636.

`-J` This option has the same function as the `-j` option, for the key password.

`-K` Specifies the name of the certificate key used for certificate-based client authentication. For example, `-K` *Server-Key*.

`-N` Specifies the certificate name to use for certificate-based client authentication. For example, `- N` *Server-Cert*. If this option is specified, the `-W` option is required.

`-P` Specifies the location of the certificate database.

`-W` Specifies the password for the certificate database identified by the `-P` option. For example, `-W` *serverpassword*.

Examples

EXAMPLE 1 Single Server, Repeat Each 30 Seconds

Note that the delay changes to 5, indicating that the consumer is 5 seconds behind the supplier.

```
$ insync -D cn=admin,cn=Administrators,cn=config -w mypword \
 -s portugal:1389 30

ReplicaDn         Consumer                 Supplier        Delay

dc=example,dc=com france.example.com:2389 portugal:1389   0

dc=example,dc=com france.example.com:2389 portugal:1389   5
```

EXAMPLE 1 Single Server, Repeat Each 30 Seconds *(Continued)*

```
dc=example,dc=com france.example.com:2389 portugal:1389   0
```

EXAMPLE 2 Getting Date of Last Change

```
$ insync -D cn=admin,cn=Administrators,cn=config -w mypword \
 -s portugal:1389 -b o=rtest -d
```

Exit Status

The following exit values are returned:

`0` Successful completion.

`1` An error occurred.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Stable

See Also

`entrycmp`(1), `repldisc`(1)

Notes

The node on which you are running the `entrycmp`, `insync`, and `repldisc` tools must be able to reach all the specified hosts. If these hosts are unavailable due to a firewall, VPN, or other network setup reasons, you will encounter difficulties using these tools. For the same reason ensure that all servers are up and running before using these tools.

When identifying hosts, you must use either symbolic names or IP addresses for all hosts since the replication monitoring commands do not address resolution between symbolic names and IP addresses. Using a combination of the two can cause problems. Moreover, on multi-homed hosts, referring to the same Directory Server instance using different names may cause unexpected results.

When SSL is enabled, the directory server on which you are running the tools must have a copy of all the certificates used by the other servers in the topology.

If a delay of -1 is returned, `insync` was unable to obtain any replication information. This may indicate that a Total Update has just been run, or that no changes have been sent to the supplier server.

The replication monitoring tools rely on access to `cn=config` to obtain the replication status. This should be taken into account particularly when replication is configured over SSL.

Name

ldapcmp – compare LDAP entries from two directories

Synopsis

```
/opt/SUNWdsee/dsee6/bin/ldapcmp  [-h host1 -p port1 [-h host2 -p port2]]
 [options] -b basedn
```

```
/opt/sun/dsee6/bin/ldapcmp  [-h host1 -p port1 [-h host2 -p port2]]
 [options] -b basedn
```

```
install-path/dsrk6/bin/ldapcmp
 [-h host1 -p port1 [-h host2 -p port2]] [options] -b basedn
```

Description

The `ldapcmp` command compares a Lightweight Directory Access Protocol (LDAP) entry or subtree of entries from one directory with the an entry or subtree of entries from another directory. It detects entries that do not appear in both directories and detects attribute differences in entries that do appear in both directories.

The `ldapcmp` command reports comparison results using the following output syntax:

1only: *DN*	Entry appears only in the first directory specified.
2only: *DN*	Entry appears only in the second directory specified.
DN	Entry appears in both directories, attributes differ. The `ldapcmp` command then explains the differences found:

different: *attrname*	Entries differed by attribute value.
different: *attrname*(*)	Specified attribute found only in one directory.
1: *attrvalue*	Specified value found in first directory.
2: *attrvalue*	Specified value found in second directory.

Options

Although the `-h` (host) and `-p` (port) options are not required, you generally use these options to specify how to access the two directories. If you do not specify any `-h` or `-p` options, the `ldapcmp` command compares the content of the directory listening on the default port of the `localhost` system with itself.

Unless the `LDAP_BASEDN` environment variable is set, you must at minimum provide a *basedn* argument to the `-b` option. The *basedn* argument specifies the distinguished name (DN) of the LDAP entry at the base of the search scope.

The following additional options are supported:

`-0`

Ignore LDAP library version mismatches.

When this option is omitted, the default behavior is to assert that the revision number of the LDAP API be greater than or equal to that used to compile the tool. Also, if the library and the tool have the same vendor name, the tool will assert that the vendor version number of the API be greater than or equal to that used to compile the tool. Revision and version numbers are based on the contents of the `LDAPAPIInfo` structure defined in <`ldap.h`> or header files included by<`ldap.h`>.

`-3`
Check host names in SSL certificates.

`-B`
Allow binary values to be printed, even if the `-o` option is used.

`-D` *binddn*
Use the specified bind DN for accessing both directories, usually enclosed in double quotes (`""`) for the shell.

If the bind DN and its password are omitted, the `ldapcmp` command binds anonymously. The bind DN determines what entries and attributes appear in the comparison results, according to the search permissions for the bind DN.

`-E`
Request that the directories expose (report) bind identities.

`-H`
`-help`
`--help`
`-?`
Display usage information.

`-I` *filename*
Read SSL key password for the client key database specified using the `-P` option from *filename*.

The default is `key3.db`.

`-J` *controloid*[:*criticality*[:*value*|::*base64value*|:<*fileurl*]]
Use the specified control OID.

The *criticality* is false by default.

An LDAP control can be associated with a value. Proxy authorization takes a proxy authorization ID, for example, passed with the control OID, and criticality. If a value is necessary you specify it using *value*, *base64value*, or <*fileurl*.

`-K` *pathname*
Use the SSL key database located in *pathname*, the full path to the key database file.

The default is to search for the key database file, `key3.db`, in the directory specified by the `-P` option.

`-M`
Manage referrals, returning the entry containing the referral instead of the entry obtained by following the referral.

-N *certificate*
Use the specified *certificate* for certificate-based client authentication, for example: `-N "Directory-Cert"`.

Both directories must recognize the specified certificate to perform the comparison.

-O *limit*
Follow at maximum *limit* referral hops. Default is `5`.

-P *filename*
Use the certificate database located in *filename*, the full path to the certificate database file.

The default is to search for the certificate database file, `cert8.db`, in the current directory.

-Q [*token*][:*certificate-name*]
Use PKCS 11.

-R
Do not follow referrals automatically.

-V *n*
Use LDAP protocol version *n*, where *n* is `2` or `3`. Default is `3`.

-W -
Prompt for the password for the client key database specified using the `-P` option.

The `-W` option is required for certificate-based client authentication.

-W *password*
Specify the password for the client key database specified using the `-P` option.

The `-W` option is required for certificate-based client authentication.

-Y *proxydn*
Use the specified proxy DN for accessing both directories, usually enclosed in double quotes ("") for the shell.

-Z
Use SSL to provide certificate-based client authentication.

The `-Z` option requires the `-N` and `-W` options and any other SSL options needed to identify the certificate and the key database.

-d *level*
Set LDAP debug level to the specified value.

The following debug levels are supported:

`1`	Display verbose debugging messages; `LDAP_DEBUG_TRACE`.
`2`	Display messages about the content of network packets; `LDAP_DEBUG_PACKETS`.
`320`	Display messages about LDIF parsing; `LDAP_DEBUG_PARSE`.

`16384`	Display informational messages; `LDAP_DEBUG_ANY`.

Use the sum of the levels to specify more than one debug level. For example, to set the debug level to display both verbose debugging messages, and messages about the content of network packets, specify `-d 3`.

`-h` *host*
Contact the LDAP server on the specified host, which may be a host name or an IP address.

The default is `localhost`.

Specify the host twice to specify hosts for each of the two directories. When you specify the host twice, the first host specified corresponds to the first directory, and the second host corresponds to the second, regardless of the order of other options.

`-i` *charset*
Use the specified character set to override the value of the `LANG` environment variable. This option is useful, as the command converts certain arguments you specify to UTF-8 before sending the request to the server. The following arguments are converted: base DN, bind DN, LDAP filter, and password.

You can prevent the command from converting passwords by using the `-k` option.

Examples of *charset* values include `ISO8859-1`, `ISO8859-15`, `ibm-1275`, and `windows-1251`.

`-j` *filename*
Read the bind password for simple authentication from the specified file.

`-k`
Do not convert the passwords to UTF-8.

`-l` *timelimit*
Interrupt the comparison if the specified time limit is exceeded.

`-m` *pathname*
Use the security module database located in the specified directory.

Use the `-m` option if the security module database is in a different directory from the certificate database itself.

`-n`
Show what would be done, but do not actually do it.

`-o` *attrname=attrvalue*
Use the specified attribute values when performing SASL authentication.

The following *attrname* arguments are supported:

authid	Use the specified authentication identity.
authzid	Use the specified authorization identity.

mech Request the specified SASL mechanism for the bind.

realm Use the specified realm to complete the bind.

secProp Use the specified security level.

The *attrvalue* is a valid value corresponding to the *attrname* you specify.

`-p` *port*
Contact the LDAP server on the specified port.

The default is `389` (`636` if SSL is used).

Specify the port twice to specify ports for each of the two directories. When you specify the port twice, the first port specified corresponds to the first directory, and the second port corresponds to the second, regardless of the order of other options.

`-s` *scope*
Use the specified search scope.

The following values are supported for *scope*:

base Examine only the entry specified by the argument to the `-b` option.

one Examine only to the entry specified by the argument to the `-b` option and its immediate children.

sub (Default) Examine the subtree whose root is the entry specified by the argument to the `-b` option.

`-v`
Run in verbose mode, displaying diagnostics on standard output.

`-W –`
Prompt for the bind password for simple authentication.

`-w` *password*
Use the specified bind password for simple authentication.

`-z` *sizelimit*
Interrupt the comparison if the specified maximum number of entries returned is exceeded.

Examples

All examples in this section use the following conventions:

- All entries to compare are stored under `dc=example,dc=com`.
- The directories have been configured to support anonymous access for search and read. Therefore, you do not have to specify any bind information.
- The directory servers are located on systems named `host1` and `host2`.
- The servers both listen on port number `389`, the default.

EXAMPLE 1 Comparing Two Suffixes

When you specify the root DN of the suffix as the base DN, `ldapcmp` compares all entries of the entire suffix in both directories.

```
$ ldapcmp -h host1 -h host2 -b "dc=example,dc=com"
```

You should have some idea of the size and differences between your directories before comparing them. Comparing two directories is useful for finding small difference between directories. When comparing completely different subtrees, the output can be very large. Narrow your comparison by specifying the base DN of a similar subtree in both directories.

EXAMPLE 2 Comparing Two Entries

The following command compares a single user entry in both directories:

```
$ ldapcmp -h host1 -h host2 -s base \
-b "uid=bjensen,ou=People,dc=example,dc=com"
```

EXAMPLE 3 Setting the Base DN

The following commands set the `LDAP_BASEDN` environment variable, and then compare all entries of the entire base suffix in both directories, running in verbose mode. The syntax of the first command may not work for your shell. Refer to the documentation about your shell for instructions on setting environment variables.

```
$ LDAP_BASEDN="dc=example,dc=com"; export LDAP_BASEDN
$ ldapcmp -v -h host1 -h host2
```

EXAMPLE 4 Comparing Directory Configurations

The following command compares root DSE entries for both directories:

```
$ ldapcmp -h host1 -h host2 -s base -b ""
```

EXAMPLE 5 Comparing Directory Schema

The following command compares schema entries for both directories:

```
$ ldapcmp -h host1 -h host2 -b "cn=schema"
```

Exit Status

The exit status returned reflects the return values of the underlying functions used, which may depend on return values sent by the server. The return values are defined through `<ldap.h>` files both on the client side and on the server side. Common exit status codes follow:

`0` Successful completion; `LDAP_SUCCESS`; `0x00`.

`1` Server encountered errors while processing the request; `LDAP_OPERATIONS_ERROR`; `0x01`.

2 Server encountered errors, such as a BER-decoding error, while processing the request; `LDAP_PROTOCOL_ERROR`; `0x02`.

3 Search exceeded the time limit for operations on the server; `LDAP_TIMELIMIT_EXCEEDED`; `0x03`.

4 Search returned more results than the maximum number allowed by the server; `LDAP_SIZELIMIT_EXCEEDED`; `0x04`.

10 Base DN belongs to an entry handled by neither server, and the referral URL identifies another server that handles the entry; `LDAP_REFERRAL`; `0x0a`.

11 Search returned more results than the maximum number a client application is allowed by the server to retrieve; `LDAP_ADMINLIMIT_EXCEEDED`; `0x0b`.

32 Base DN belongs to an entry handled by neither server, and no referral URL is available for the entry; `LDAP_NO_SUCH_OBJECT`; `0x20`.

50 Bind DN user does not have permission to read the entry from the directory; `LDAP_INSUFFICIENT_ACCESS`; `0x32`.

81 One of the directories did not respond to the request, or the connection was lost; `LDAP_SERVER_DOWN`; `0x51`.

82 An error occurred while receiving results; `LDAP_LOCAL_ERROR`; `0x52`.

83 The request could not be BER-encoded; `LDAP_ENCODING_ERROR`; `0x53`.

84 A result could not be decoded; `LDAP_DECODING_ERROR`; `0x54`.

85 The search exceeded the time limit specified using the `-l` option; `LDAP_TIMEOUT`; `0x55`.

89 An option or argument is not valid; `LDAP_PARAM_ERROR`; `0x59`.

90 Needed memory could not be allocated; `LDAP_NO_MEMORY`; `0x5a`.

91 A specified host name or port is not valid; `LDAP_CONNECT_ERROR`; `0x5b`.

92 At least one server supports only LDAPv2, and the `-V 2` option was not used; `LDAP_NOT_SUPPORTED`; `0x5c`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldapcsdk-tools
Stability Level	Evolving

See Also

`ldapcompare`(1), `ldapdelete`(1), `ldapmodify`(1), `ldappasswd`(1), `ldapsearch`(1)

Name

ldapcompare – compare a value with an LDAP entry attribute value

Synopsis

```
/opt/SUNWdsee/dsee6/bin/ldapcompare  [options] attrtype:attrvalue
 [dn]...
```

```
/opt/SUNWdsee/dsee6/bin/ldapcompare  [options] attrtype::base64value
 [dn]...
```

```
/opt/SUNWdsee/dsee6/bin/ldapcompare  [options] attrtype:<fileurl
 [dn]...
```

```
install-path/dsrk6/bin/ldapcompare
 [options] attrtype:attrvalue [dn]...
```

```
install-path/dsrk6/bin/ldapcompare
 [options] attrtype::base64value [dn]...
```

```
install-path/dsrk6/bin/ldapcompare
 [options] attrtype:<fileurl [dn]...
```

Description

The `ldapcompare` command asserts that a value you specify is the same as an entry attribute value stored by the directory server.

Specify the attribute type, followed by the attribute value, either as a string, a base64–encoded value, or a URL to a file containing the attribute value, such as a photo or certificate. You typically enclose the attribute type/value pair in single quotes (`''`) for the shell.

Also specify one or more entry DNs, separated by space, and typically enclosed in double quotes ("") for the shell. The `ldapcompare` command then compares the specified attribute value to that of attributes on each of the entries indicated by the DNs you provide.

Options

The following options are supported:

`-0`

Ignore LDAP library version mismatches.

When this option is omitted, the default behavior is to assert that the revision number of the LDAP API be greater than or equal to that used to compile the tool. Also, if the library and the tool have the same vendor name, the tool will assert that the vendor version number of the API be greater than or equal to that used to compile the tool. Revision and version numbers are based on the contents of the `LDAPAPIInfo` structure defined in <`ldap.h`> or header files included by <`ldap.h`>.

`-3`

Check host names in SSL certificates.

`-D` *bindDN*

Use the specified bind DN to authenticate to the directory server.

If the bind DN and its password are omitted, the `ldapcompare` command binds anonymously. The bind DN determines what entries and attributes appear in the comparison results, according to the search permissions for the bind DN.

`-E`
Request that the directories expose (report) bind identities.

`-H`
`-help`
`--help`
`-?`
Display usage information.

`-I` *filename*
Read SSL key password for the client key database specified using the `-P` option from *filename*.

The default is `key3.db`.

`-J` *controloid*[:*criticality*[:*value*|::*base64value*|:<*fileurl*]]
Use the specified control OID.

The *criticality*, a boolean, is false by default.

An LDAP control can be associated with a value. Proxy authorization takes a proxy authorization ID, for example, passed with the control OID, and criticality. If a value is necessary you specify it using *value*, *base64value*, or <*fileurl*.

`-K` *pathname*
Use the SSL key database located in *pathname*, the full path to the key database file.

The default is to search for the key database file, `key3.db`, in the directory specified by the `-P` option.

`-M`
Manage referrals, comparing the entry containing the referral instead of the entry obtained by following the referral.

`-N` *certificate*
Use the specified *certificate* for certificate-based client authentication, for example: `-N "Client-Cert"`, where `Client-Cert` is the subject name of the user certificate.

`-O` *limit*
Follow at maximum *limit* referral hops. Default is `5`.

`-P` *pathname*
Use the certificate database located in *pathname*, the full path to the certificate database file.

The default is to search for the certificate database file, `cert8.db`, in the current directory.

`-Q` [*token*][:*certificate-name*]
Use PKCS 11.

`-R`
Do not follow referrals automatically.

`-V` *n*
Use LDAP protocol version *n*, where *n* is `2` or `3`. Default is `3`.

`-W -`
Prompt for the password for the client key database specified using the `-P` option.

The `-W` option is required for certificate-based client authentication.

`-W` *password*
Specify the password for the client key database specified using the `-P` option.

The `-W` option is required for certificate-based client authentication.

`-Y` *proxydn*
Use the rights of the entry having the specified DN for performing LDAP operations. When using this option, you must also specify how to bind before you assume the rights of the proxy. Thus, when using simple authentication, you would also use the `-D` and `-w` options with this option.

Before proxy authentication can work in Directory Server, you must set up the appropriate access control instructions.

`-Z`
Use SSL to provide certificate-based client authentication.

The `-Z` option requires the `-N` and `-W` options and any other SSL options needed to identify the certificate and the key database.

`-ZZ`
Use Start TLS to provide certificate-based client authentication.

The `-ZZ` option requires the `-N` and `-W` options and any other SSL options needed to identify the certificate and the key database.

`-c`
Run in continuous mode, not stopping on errors.

In continuous mode, errors are reported but the `ldapcompare` command continues performing comparisons. When not running in continuous mode, the `ldapcompare` command quits after the first error.

`-d` *level*
Set LDAP debug level to the specified value.

The following debug levels are supported:

`1`	Display verbose debugging messages; `LDAP_DEBUG_TRACE`.
`2`	Display messages about the content of network packets; `LDAP_DEBUG_PACKETS`.
`320`	Display messages about LDIF parsing; `LDAP_DEBUG_PARSE`.

`16384`	Display informational messages; `LDAP_DEBUG_ANY`.

Use the sum of the levels to specify more than one debug level. For example, to set the debug level to display both verbose debugging messages, and messages about the content of network packets, specify `-d 3`.

`-f` *filename*
Read DNs from the specified file.

The file format is one DN per line without quotes around DNs. The `ldapcompare` command reads each line as one literal DN, performing the comparison for each entry whose DN is specified.

`-h` *host*
Contact the LDAP server on the specified host, which may be a host name or an IP address. Enclose IPv6 addresses in brackets (`[]`) as described in RFC 2732.

For example, when mapping the IPv4 address `192.168.0.99` to IPv6, pass the `-h` option with its argument as `-h [::ffff:192.168.0.99]`. Notice the brackets.

When using GSSAPI with Directory Server, specify the *host* as a fully-qualified host name which matches the value of the `nsslapd-localhost` attribute on the `cn=config` entry. The GSSAPI authentication process requires that the host name provided by the client match the one provided by the server.

The default is `localhost`.

`-i` *charset*
Use the specified character set to override the value of the `LANG` environment variable. This option is useful, as the command converts certain arguments you specify to UTF-8 before sending the request to the server. The following arguments are converted: base DN, bind DN, LDAP filter, and password.

You can prevent the command from converting passwords by using the `-k` option.

Examples of *charset* values include `ISO8859-1`, `ISO8859-15`, `ibm-1275`, and `windows-1251`.

`-j` *filename*
Read the bind password for simple authentication from the specified file.

`-k`
Do not convert the passwords to UTF-8.

`-m` *pathname*
Use the security module database located in the specified directory.

Use the `-m` option if the security module database is in a different directory from the certificate database itself.

`-n`
Show what would be done, but do not actually do it.

`-o` *attrname=attrvalue*

Use the specified attribute values when performing SASL authentication.

The following *attrname* arguments are supported:

authid	Use the specified authentication identity.
authzid	Use the specified authorization identity.
mech	Request the specified SASL mechanism for the bind.
realm	Use the specified realm to complete the bind.
secProp	Use the specified security level.

The *attrvalue* is a valid value corresponding to the *attrname* you specify.

`-p` *port*

Contact the LDAP server on the specified port.

The default is `389` (`636` if SSL is used).

`-q`

Run in quiet mode, displaying no information about results of comparisons, but only about LDAP errors.

`-v`

Run in verbose mode, displaying diagnostics on standard output.

`-w` –

Prompt for the bind password for simple authentication.

`-w` *password*

Use the specified bind password for simple authentication.

Examples Examples in this section use the following conventions:

- The directory server is located on a system named `host`.
- The directory server has been configured to support anonymous access for search and read. Therefore, you do not have to specify bind information.
- The directory server listens on port number `389`, the default.

EXAMPLE 1 Comparing String Values

The following command compares a specified string with an attribute value:

```
$ ./ldapcompare -h host ’givenname:Barbara’ \
"uid=bjensen,ou=People,dc=example,dc=com"
comparing type: "givenname" value: "Barbara"
 in entry "uid=bjensen,ou=People,dc=example,dc=com"
compare TRUE
```

EXAMPLE 2 Comparing Base 64 Encoded Values

The following command compares a base64–encoded value with an attribute value:

```
$ ./ldapcompare -h host 'cn::QmFicyBKZW5zZW4=' \
"uid=bjensen,ou=People,dc=example,dc=com"
comparing type: "cn" value: "Babs Jensen"
 in entry "uid=bjensen,ou=People,dc=example,dc=com"
compare TRUE
```

EXAMPLE 3 Comparing Binary Values in Files

The following command compares an image with an attribute value:

```
$ ./ldapcompare -h host 'jpegphoto:<file:///home/bjensen/bjensen.jpg' \
"uid=bjensen,ou=People,dc=example,dc=com"
comparing type: "jpegphoto" value: "NOT ASCII (3674 bytes)"
 in entry "uid=bjensen,ou=People,dc=example,dc=com"
compare TRUE
```

Exit Status

The exit status returned either corresponds to 5 (LDAP_COMPARE_FALSE) or 6 (LDAP_COMPARE_TRUE), or reflects the return values of the underlying functions used, which may depend on return values sent by the server. Common exit status codes follow:

1 Server encountered errors while processing the request; LDAP_OPERATIONS_ERROR; 0x01.

2 Server encountered errors, such as a BER-decoding error, while processing the request; LDAP_PROTOCOL_ERROR; 0x02.

3 Search exceeded the time limit for operations on the server; LDAP_TIMELIMIT_EXCEEDED; 0x03.

5 Operation was successful but the values did not match; LDAP_COMPARE_FALSE; 0x05.

6 Operation was successful and the values match; LDAP_COMPARE_TRUE; 0x06.

10 DN of the entry to compare belongs to an entry handled by neither server, and the referral URL identifies another server that handles the entry; LDAP_REFERRAL; 0x0a.

32 DN of the entry to compare belongs to an entry handled by neither server, and no referral URL is available for the entry; LDAP_NO_SUCH_OBJECT; 0x20.

34 DN of the entry to compare is not a valid DN; LDAP_INVALID_DN_SYNTAX; 0x22.

50 Bind DN user does not have permission to read the entry from the directory; LDAP_INSUFFICIENT_ACCESS; 0x32.

81 One of the directories did not respond to the request, or the connection was lost; LDAP_SERVER_DOWN; 0x51.

82 An error occurred while receiving results; `LDAP_LOCAL_ERROR`; `0x52`.

83 The request could not be BER-encoded; `LDAP_ENCODING_ERROR`; `0x53`.

84 A result could not be decoded; `LDAP_DECODING_ERROR`; `0x54`.

89 An option or argument is not valid; `LDAP_PARAM_ERROR`; `0x59`.

90 Needed memory could not be allocated; `LDAP_NO_MEMORY`; `0x5a`.

91 A specified host name or port is not valid; `LDAP_CONNECT_ERROR`; `0x5b`.

92 At least one server supports only LDAPv2, and the `-V 2` option was not used, or the `-V 2` option was used, but the server no longer supports LDAP v2; `LDAP_NOT_SUPPORTED`; `0x5c`.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldapcsdk-tools
Stability Level	Evolving

See Also `ldapcmp`(1), `ldapdelete`(1), `ldapmodify`(1), `ldapsearch`(1), `ldappasswd`(1)

Name ldapdelete – delete LDAP entries

Synopsis `/opt/SUNWdsee/dsee6/bin/ldapdelete` [*options*] [*dn*]...

`/opt/SUNWdsee/dsee6/bin/ldapdelete` [*options*] < *filename*

install-path`/dsrk6/bin/ldapdelete`
[*options*] [*dn*]...

install-path`/dsrk6/bin/ldapdelete`
[*options*] < *filename*

Description The `ldapdelete` command requests deletion of entries stored by a directory server. You must bind as a user having access to delete the entries specified.

Specify one or more entry DNs, separated by space, and typically enclosed in double quotes ("") for the shell. Alternatively, include DNs in a file, one per line without quotes around DNs. The `ldapdelete` command reads each line as one literal DN.

When deleting a subtree, you must delete child entries before you delete their parent entries.

Options The following options are supported:

`-0`
Ignore LDAP library version mismatches.

When this option is omitted, the default behavior is to assert that the revision number of the LDAP API be greater than or equal to that used to compile the tool. Also, if the library and the tool have the same vendor name, the tool will assert that the vendor version number of the API be greater than or equal to that used to compile the tool. Revision and version numbers are based on the contents of the `LDAPAPIInfo` structure defined in <`ldap.h`> or header files included by <`ldap.h`>.

`-3`
Check host names in SSL certificates.

`-D` *bindDN*
Use the specified bind DN to authenticate to the directory server.

If the bind DN and its password are omitted, the `ldapdelete` command binds anonymously. The bind DN determines whether the delete operation can complete, according to the user permissions.

`-E`
Request that the directories expose (report) bind identities.

`-H`
`-help`
`--help`
`-?`
Display usage information.

`-I` *filename*
Read SSL key password for the client key database specified using the `-P` option from *filename*.

The default is `key3.db`.

`-J` *controloid*[:*criticality*[:*value*|::*base64value*|:<*fileurl*]]
Use the specified control OID.

The *criticality*, a boolean, is false by default.

An LDAP control can be associated with a value. Proxy authorization takes a proxy authorization ID, for example, passed with the control OID, and criticality. If a value is necessary you specify it using *value*, *base64value*, or <*fileurl*.

`-K` *pathname*
Use the SSL key database located in *pathname*, the full path to the key database file.

The default is to search for the key database file, `key3.db`, in the directory specified by the `-P` option.

`-M`
Manage referrals, deleting the entry containing the referral instead of the entry obtained by following the referral.

`-N` *certificate*
Use the specified *certificate* for certificate-based client authentication, for example: `-N "Client-Cert"`, where `Client-Cert` is the subject name of the user certificate.

`-O` *limit*
Follow at maximum *limit* referral hops. Default is `5`.

`-P` *pathname*
Use the certificate database located in *pathname*, the full path to the certificate database file.

The default is to search for the certificate database file, `cert8.db`, in the current directory.

`-Q` [*token*][:*certificate-name*]
Use PKCS 11.

`-R`
Do not follow referrals automatically.

`-V` *n*
Use LDAP protocol version *n*, where *n* is `2` or `3`. Default is `3`.

`-W -`
Prompt for the password for the client key database specified using the `-P` option.

The `-W` option is required for certificate-based client authentication.

`-W` *password*

Specify the password for the client key database specified using the `-P` option.

The `-W` option is required for certificate-based client authentication.

`-Y` *proxydn*

Use the rights of the entry having the specified DN for performing LDAP operations. When using this option, you must also specify how to bind before you assume the rights of the proxy. Thus, when using simple authentication, you would also use the `-D` and `-w` options with this option.

Before proxy authentication can work in Directory Server, you must set up the appropriate access control instructions.

`-Z`

Use SSL to provide certificate-based client authentication.

The `-Z` option requires the `-N` and `-W` options and any other SSL options needed to identify the certificate and the key database.

`-ZZ`

Use Start TLS to provide certificate-based client authentication.

The `-ZZ` option requires the `-N` and `-W` options and any other SSL options needed to identify the certificate and the key database.

`-c`

Run in continuous mode, not stopping on errors.

In continuous mode, errors are reported but the `ldapdelete` command continues. When not running in continuous mode, the `ldapdelete` command quits after the first error.

`-d` *level*

Set LDAP debug level to the specified value.

The following debug levels are supported:

`1`	Display verbose debugging messages; `LDAP_DEBUG_TRACE`.
`2`	Display messages about the content of network packets; `LDAP_DEBUG_PACKETS`.
`320`	Display messages about LDIF parsing; `LDAP_DEBUG_PARSE`.
`16384`	Display informational messages; `LDAP_DEBUG_ANY`.

Use the sum of the levels to specify more than one debug level. For example, to set the debug level to display both verbose debugging messages, and messages about the content of network packets, specify `-d 3`.

`-f` *filename*

Read DNs from the specified file.

The file format is one DN per line without quotes around DNs. The `ldapdelete` command reads each line as one literal DN.

This option has no effect when you also specify DNs on standard input.

`-h` *host*
Contact the LDAP server on the specified host, which may be a host name or an IP address. Enclose IPv6 addresses in brackets (`[]`) as described in RFC 2732.

For example, when mapping the IPv4 address `192.168.0.99` to IPv6, pass the `-h` option with its argument as `-h [::ffff:192.168.0.99]`. Notice the brackets.

When using GSSAPI with Directory Server, specify the *host* as a fully-qualified host name which matches the value of the `nsslapd-localhost` attribute on the `cn=config` entry. The GSSAPI authentication process requires that the host name provided by the client match the one provided by the server.

The default is `localhost`.

`-i` *charset*
Use the specified character set to override the value of the `LANG` environment variable. This option is useful, as the command converts certain arguments you specify to UTF-8 before sending the request to the server. The following arguments are converted: base DN, bind DN, LDAP filter, and password.

You can prevent the command from converting passwords by using the `-k` option.

Examples of *charset* values include `ISO8859-1`, `ISO8859-15`, `ibm-1275`, and `windows-1251`.

`-j` *filename*
Read the bind password for simple authentication from the specified file.

`-k`
Do not convert the passwords to UTF-8.

`-m` *pathname*
Use the security module database located in the specified directory.

Use the `-m` option if the security module database is in a different directory from the certificate database itself.

`-n`
Show what would be done, but do not actually do it.

`-o` *attrname=attrvalue*
Use the specified attribute values when performing SASL authentication.

The following *attrname* arguments are supported:

authid	Use the specified authentication identity.

authzid	Use the specified authorization identity.
mech	Request the specified SASL mechanism for the bind.
realm	Use the specified realm to complete the bind.
secProp	Use the specified security level.

The *attrvalue* is a valid value corresponding to the *attrname* you specify.

`-p` *port*
Contact the LDAP server on the specified port.

The default is `389` (`636` if SSL is used).

`-v`
Run in verbose mode, displaying diagnostics on standard output.

`-w` –
Prompt for the bind password for simple authentication.

`-w` *password*
Use the specified bind password for simple authentication.

Examples

Examples in this section use the following conventions:

- The bind DN given corresponds to a user with permission to delete entries.
- The directory server is located on a system named `host`.
- The directory server listens on port number `389`, the default for non-SSL traffic.
- The directory server listens on port number `636`, the default for SSL traffic. SSL is enabled.

EXAMPLE 1 Deleting an Entry

The following command deletes a single entry from the directory:

```
$ ./ldapdelete -h host -D uid=kvaughan,ou=people,dc=example,dc=com \
-w - uid=scarter,ou=People,dc=example,dc=com
Enter bind password:
$
```

EXAMPLE 2 Deleting an Entry Interactively

The following commands demonstrate deleting an entry whose DN is specified on standard input:

```
$ ./ldapdelete -h host -D uid=kvaughan,ou=People,dc=example,dc=com \
-w - -c -v
Enter bind password:
ldapdelete: started Tues Oct 18 08:31:14 2005
```

EXAMPLE 2 Deleting an Entry Interactively *(Continued)*

```
ldap_init( host, 389 )

uid=scarter, ou=People, dc=example,dc=com
deleting entry uid=scarter, ou=People, dc=example,dc=com
entry removed
^D
$
```

EXAMPLE 3 Deleting Multiple Entries Specified in a File

The following commands demonstrate reading DNs of entries to delete from a file. Notice that the `-c` option is used to continue if an error occurs.

```
$ cat DNfile
uid=scarter, ou=People, dc=example,dc=com
uid=bjensen, ou=People, dc=example,dc=com
$ ./ldapdelete -h host -D uid=kvaughan,ou=People,dc=example,dc=com \
-c -f DNfile -w -
Enter bind password:
$
```

EXAMPLE 4 Using Server Authentication

The following command uses server authentication during the bind, where the server only accepts binds by clients with trusted certificates. Notice only the `-P` option is used without other SSL-related options.

```
$ ./ldapdelete -h host -p 636 -c -f DNfile -P /home/kvaughan/security \
-D uid=kvaughan,ou=People,dc=example,dc=com -w -
Enter bind password:
```

EXAMPLE 5 Using Client Authentication

The following command uses client authentication during the bind, where the server only accepts binds by clients with trusted certificates, and the client must sign the certificate with a password-protected private key. Notice the options used in this example.

```
$ ./ldapdelete -h host -p 636 -c -f DNfile -Z -P /home/kvaughan/security \
-N "kvscert" -K /home/kvaughan/security -W keypassword
```

Exit Status

The exit status returned reflects the return values of the underlying functions used, which may depend on return values sent by the server. Common exit status codes follow:

`0` Successful completion; `LDAP_SUCCESS`; `0x00`.

1 Server encountered errors while processing the request; `LDAP_OPERATIONS_ERROR`; `0x01`.

2 Server encountered errors, such as a BER-decoding error, while processing the request; `LDAP_PROTOCOL_ERROR`; `0x02`.

10 DN of the entry to delete belongs to an entry handled by neither server, and the referral URL identifies another server that handles the entry; `LDAP_REFERRAL`; `0x0a`.

32 DN of the entry to delete belongs to an entry handled by neither server, and no referral URL is available for the entry; `LDAP_NO_SUCH_OBJECT`; `0x20`.

34 DN of the entry to delete is not a valid DN; `LDAP_INVALID_DN_SYNTAX`; `0x22`.

50 Bind DN user does not have permission to read the entry from the directory; `LDAP_INSUFFICIENT_ACCESS`; `0x32`.

53 Directory is read-only; `LDAP_UNWILLING_TO_PERFORM`; `0x35`.

66 Entry specified has child-entries that must be deleted first; `LDAP_NOT_ALLOWED_ON_NONLEAF`; `0x42`.

81 One of the directories did not respond to the request, or the connection was lost; `LDAP_SERVER_DOWN`; `0x51`.

82 An error occurred while receiving results; `LDAP_LOCAL_ERROR`; `0x52`.

83 The request could not be BER-encoded; `LDAP_ENCODING_ERROR`; `0x53`.

84 A result could not be decoded; `LDAP_DECODING_ERROR`; `0x54`.

89 An option or argument is not valid; `LDAP_PARAM_ERROR`; `0x59`.

90 Needed memory could not be allocated; `LDAP_NO_MEMORY`; `0x5a`.

91 A specified host name or port is not valid; `LDAP_CONNECT_ERROR`; `0x5b`.

92 At least one server supports only LDAPv2, and the `-V 2` option was not used, or the `-V 2` option was used, but the server no longer supports LDAP v2; `LDAP_NOT_SUPPORTED`; `0x5c`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldapcsdk-tools
Stability Level	Evolving

See Also

`ldapcmp`(1), `ldapcompare`(1), `ldapmodify`(1), `ldappasswd`(1), `ldapsearch`(1)

Name ldapmodify – add, modify, rename, move, or delete LDAP entries

Synopsis `/opt/SUNWdsee/dsee6/bin/ldapmodify` [*options*]

install-path`/dsrk6/bin/ldapmodify`
[*options*]

Description The `ldapmodify` command requests the addition, modification, rename, move, or deletion of entries stored by a directory server.

You must bind as a user having access to perform the requested operation.

The directory server may check all modifications against its schema, and reject updates that cause entries not to conform to the schema.

You must specify additions and modifications in the proper order, because the directory server performs the updates in the order you request them. For example, to add entries to a subtree that does not yet exist, you must first update the base entry at the root of the subtree before adding entries under the base entry. When a requested operation fails, the `ldapmodify` command stops processing further input unless you use the `-c` option. The `ldapmodify` command does not save rejected entries unless you use the `-e` option.

Options The following options are supported:

`-0`
Ignore LDAP library version mismatches.

When this option is omitted, the default behavior is to assert that the revision number of the LDAP API be greater than or equal to that used to compile the tool. Also, if the library and the tool have the same vendor name, the tool will assert that the vendor version number of the API be greater than or equal to that used to compile the tool. Revision and version numbers are based on the contents of the `LDAPAPIInfo` structure defined in <`ldap.h`> or header files included by <`ldap.h`>.

`-3`
Check host names in SSL certificates.

`-A`
Display non-ASCII values when the `-v` option is used.

`-B` *baseDN*
Bulk import entries into the suffix under the specified DN.

Bulk import using the `ldapmodify` command does not erase entries that already exist.

`-D` *bindDN*
Use the specified bind DN to authenticate to the directory server.

If the bind DN and its password are omitted, the `ldapmodify` command binds anonymously. The bind DN determines what entries and attributes appear in the comparison results, according to the search permissions for the bind DN.

`-E`
Request that the directories expose (report) bind identities.

`-F`
Force application of all modifications, even if some lines are duplicates.

`-H`
`-help`
`--help`
`-?`
Display usage information.

`-I` *filename*
Read SSL key password for the client key database specified using the `-P` option from *filename*.

The default is `key3.db`.

`-J` *controloid*[:*criticality*[:*value*|::*base64value*|:<*fileurl*]]
Use the specified control OID.

The *criticality*, a boolean, is false by default.

An LDAP control can be associated with a value. Proxy authorization takes a proxy authorization ID, for example, passed with the control OID, and criticality. If a value is necessary you specify it using *value*, *base64value*, or <*fileurl*.

`-K` *pathname*
Use the SSL key database located in *pathname*, the full path to the key database file.

The default is to search for the key database file, `key3.db`, in the directory specified by the `-P` option.

`-M`
Manage referrals, modifying the entry containing the referral instead of the entry obtained by following the referral.

`-N` *certificate*
Use the specified *certificate* for certificate-based client authentication, for example: `-N "Client-Cert"`, where `Client-Cert` is the subject name of the user certificate.

`-O` *limit*
Follow at maximum *limit* referral hops. Default is `5`.

`-P` *filename*
Use the certificate database located in *filename*, the full path to the certificate database file.

The default is to search for the certificate database file, `cert8.db`, in the current directory.

`-Q` [*token*][:*certificate-name*]
Use PKCS 11.

`-R`
Do not follow referrals automatically.

`-V` *n*
Use LDAP protocol version *n*, where *n* is `2` or `3`. Default is `3`.

`-W -`
Prompt for the password for the client key database specified using the `-P` option.

The `-W` option is required for certificate-based client authentication.

`-W` *password*
Specify the password for the client key database specified using the `-P` option.

The `-W` option is required for certificate-based client authentication.

`-Y` *proxydn*
Use the rights of the entry having the specified DN for performing LDAP operations. When using this option, you must also specify how to bind before you assume the rights of the proxy. Thus, when using simple authentication, you would also use the `-D` and `-w` options with this option.

Before proxy authentication can work in Directory Server, you must set up the appropriate access control instructions.

`-Z`
Use Start TLS to provide certificate-based client authentication.

The `-ZZ` option requires the `-N` and `-W` options and any other SSL options needed to identify the certificate and the key database.

`-ZZ`
Use a start TLS request .

The `-Z` option requires the `-N` and `-W` options and any other SSL options needed to identify the certificate and the key database.

`-a`
Add LDAP entries, rather than modifying existing entries.

`-b`
Handle binary files.

Note – This option is deprecated. Use standard LDIF notation as described in RFC 2849 instead.

When you use the `-b` option, the `ldapmodify` command scans every attribute value to determine whether it specifies a valid file reference, such as `/home/bjensen/bjensen.jpg`. If so, the `ldapmodify` command uses the content of the specified file as the attribute value.

`-c`
Run in continuous mode, not stopping on errors.

In continuous mode, errors are reported but the `ldapmodify` command continues performing comparisons. When not running in continuous mode, the `ldapmodify` command quits after the first error.

`-d` *level*
Set LDAP debug level to the specified value.

The following debug levels are supported:

`1`	Display verbose debugging messages; `LDAP_DEBUG_TRACE`.
`2`	Display messages about the content of network packets; `LDAP_DEBUG_PACKETS`.
`320`	Display messages about LDIF parsing; `LDAP_DEBUG_PARSE`.
`16384`	Display informational messages; `LDAP_DEBUG_ANY`.

Use the sum of the levels to specify more than one debug level. For example, to set the debug level to display both verbose debugging messages, and messages about the content of network packets, specify `-d 3`.

`-e` *filename*
Save rejected entries in the specified file.

`-f` *filename*
Read modifications from the specified file.

The file format is standard LDIF notation as described in RFC 2849.

`-h` *host*
Contact the LDAP server on the specified host, which may be a host name or an IP address. Enclose IPv6 addresses in brackets (`[]`) as described in RFC 2732.

For example, when mapping the IPv4 address `192.168.0.99` to IPv6, pass the `-h` option with its argument as `-h [::ffff:192.168.0.99]`. Notice the brackets.

When using GSSAPI with Directory Server, specify the *host* as a fully-qualified host name which matches the value of the `nsslapd-localhost` attribute on the `cn=config` entry. The GSSAPI authentication process requires that the host name provided by the client match the one provided by the server.

The default is `localhost`.

`-i` *charset*
Use the specified character set to override the value of the `LANG` environment variable. This option is useful, as the command converts certain arguments you specify to UTF-8 before sending the request to the server. The following arguments are converted: base DN, bind DN, LDAP filter, and password.

You can prevent the command from converting passwords by using the `-k` option.

Examples of *charset* values include `ISO8859-1`, `ISO8859-15`, `ibm-1275`, and `windows-1251`.

`-j` *filename*
Read the bind password for simple authentication from the specified file.

`-k`
Do not convert the passwords to UTF-8.

`-m` *pathname*
Use the security module database located in the specified directory.

Use the `-m` option if the security module database is in a different directory from the certificate database itself.

`-n`
Show what would be done, but do not actually do it.

`-o` *attrname=attrvalue*
Use the specified attribute values when performing SASL authentication.

The following *attrname* arguments are supported:

authid	Use the specified authentication identity.
authzid	Use the specified authorization identity.
mech	Request the specified SASL mechanism for the bind.
realm	Use the specified realm to complete the bind.
secProp	Use the specified security level.

The *attrvalue* is a valid value corresponding to the *attrname* you specify.

`-p` *port*
Contact the LDAP server on the specified port.

The default is `389` (`636` if SSL is used).

`-q`
Run in quiet mode, not displaying information about the operations performed.

`-v`
Run in verbose mode, displaying diagnostics on standard output.

`-w –`
Prompt for the bind password for simple authentication.

`-w` *password*
Use the specified bind password for simple authentication.

Examples Examples in this section use the following conventions:

- The bind DN given corresponds to a user with permission to update entries.
- The directory server is located on a system named `host`.
- The directory server listens on port number `389`, the default for non-SSL traffic.
- The directory server listens on port number `636`, the default for SSL traffic. SSL is enabled.

EXAMPLE 1 Adding an Entry

The following commands demonstrate adding a single entry to the directory:

```
$ cat add.ldif
dn: uid=bcubbins,ou=People,dc=example,dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
uid: bcubbins
givenName: Bartholomew
sn: Cubbins
cn: Bartholomew Cubbins
mail: bcubbins@example.com
userPassword: bcubbins
facsimiletelephonenumber: +1 234 567 8910

$ ldapmodify -a -h host -D uid=bjensen,ou=people,dc=example,dc=com \
-w - -f add.ldif
Enter bind password:
adding new entry uid=bcubbins,ou=People,dc=example,dc=com

$
```

EXAMPLE 2 Modifying an Entry

The following commands demonstrate modifying an entry. Notice a line with a single dash (`-`) separates multiple modifications to a single entry.

```
$ cat modify.ldif
dn: uid=bcubbins,ou=People,dc=example,dc=com
changetype: modify
add: description
description: Added with ldapmodify
-
replace: mail
mail: bart@example.com

$ ./ldapmodify -h host -c -v \
-D uid=bjensen,ou=People,dc=example,dc=com -w - -f modify.ldif
Enter bind password:
```

EXAMPLE 2 Modifying an Entry *(Continued)*

```
modifying entry uid=bcubbins,ou=People,dc=example,dc=com

$
```

EXAMPLE 3 Deleting an Entry Interactively

The following commands delete the entry added and modified in previous examples.

```
$ ./ldapmodify -h host -D uid=bjensen,ou=People,dc=example,dc=com -w -
Enter bind password:
dn: uid=bcubbins,ou=People,dc=example,dc=com
changetype: delete
deleting entry uid=bcubbins,ou=People,dc=example,dc=com
^D

$
```

EXAMPLE 4 Using Server Authentication

The following command uses server authentication during the bind, where the server only accepts binds by clients with trusted certificates. Notice only the `-P` option is used without other SSL-related options.

```
$ ./ldapmodify -h host -p 636 -c -f modify.ldif -P /home/bjensen/security \
-D "uid=bjensen,ou=People,dc=example,dc=com" -w -
Enter bind password:
```

EXAMPLE 5 Using Client Authentication

The following command uses client authentication during the bind, where the server only accepts binds by clients with trusted certificates, and the client must sign the certificate with a password-protected private key. Notice the options used in this example.

```
$ ldapmodify -h host -p 636 -c -Z -P /home/bjensen/security \
 -N "bjscert" -K /home/bjensen/security -W keypassword -f modify.ldif
```

EXAMPLE 6 Moving an Entry

The following command moves an entry from one branch of a suffix to another:

```
$./ldapmodify -h host -D uid=hmiller,ou=people,dc=example,dc=com -w -
Enter bind password:
dn: uid=jwallace,ou=people,dc=example,dc=com
changetype: modrdn
newrdn: uid=jwallace
```

EXAMPLE 6 Moving an Entry *(Continued)*

```
deleteoldrdn: 0
newsuperior: ou=special users,dc=example,dc=com
^D
```

Exit Status

The exit status returned reflects the return values of the underlying functions used, which may depend on return values sent by the server. Common exit status codes follow:

0 Successful completion; `LDAP_SUCCESS`; `0x00`.

1 Server encountered errors while processing the request; `LDAP_OPERATIONS_ERROR`; `0x01`.

2 Server encountered errors, such as a BER-decoding error, while processing the request; `LDAP_PROTOCOL_ERROR`; `0x02`.

10 DN of the entry to modify belongs to an entry handled by neither server, and the referral URL identifies another server that handles the entry; `LDAP_REFERRAL`; `0x0a`.

16 Attribute to be modified does not exist; `LDAP_NO_SUCH_ATTRIBUTE`; `0x10`.

19 Attribute modification requested is not a proper modification. For example, a requested change to `userpassword` would result in a user password shorter than the minimum length allowed; `LDAP_CONSTRAINT_VIOLATION`; `0x13`.

20 Attribute to add already exists with the specified value; `LDAP_TYPE_OR_VALUE_EXISTS`; `0x14`.

21 The value modified does not respect the syntax for the attribute type; `LDAP_INVALID_SYNTAX`; `0x15`.

32 DN of the entry to modify belongs to an entry handled by neither server, and no referral URL is available for the entry; `LDAP_NO_SUCH_OBJECT`; `0x20`.

34 DN of the entry to modify is not a valid DN; `LDAP_INVALID_DN_SYNTAX`; `0x22`.

50 Bind DN user does not have permission to read the entry from the directory; `LDAP_INSUFFICIENT_ACCESS`; `0x32`.

53 Directory is read-only; `LDAP_UNWILLING_TO_PERFORM`; `0x35`.

65 Requested modification would cause the entry not to comply with the directory schema; `LDAP_OBJECT_CLASS_VIOLATION`; `0x41`.

66 Entry specified has child-entries that must be deleted first; `LDAP_NOT_ALLOWED_ON_NONLEAF`; `0x42`.

67 Requested modification would cause the entry to be missing attributes that are components of the entry DN; `LDAP_NOT_ALLOWED_ON_RDN`; `0x43`.

68 An entry already exists with the same DN as the entry to add; LDAP_ALREADY_EXISTS; 0x44.

81 One of the directories did not respond to the request, or the connection was lost; LDAP_SERVER_DOWN; 0x51.

82 An error occurred while receiving results; LDAP_LOCAL_ERROR; 0x52.

83 The request could not be BER-encoded; LDAP_ENCODING_ERROR; 0x53.

84 A result could not be decoded; LDAP_DECODING_ERROR; 0x54.

89 An option or argument is not valid; LDAP_PARAM_ERROR; 0x59.

90 Needed memory could not be allocated; LDAP_NO_MEMORY; 0x5a.

91 A specified host name or port is not valid; LDAP_CONNECT_ERROR; 0x5b.

92 At least one server supports only LDAPv2, and the -V 2 option was not used, or the -V 2 option was used, but the server no longer supports LDAP v2; LDAP_NOT_SUPPORTED; 0x5c.

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldapcsdk-tools
Stability Level	Evolving

See Also

ldapcmp(1), ldapcompare(1), ldapdelete(1), ldappasswd(1), ldapsearch(1)

Name ldappasswd – change the password of an LDAP entry

Synopsis `/opt/SUNWdsee/dsee6/bin/ldappasswd` [*options*] [*auth-id*]

install-path`/dsrk6/bin/ldappasswd`
[*options*] [*auth-id*]

Description The `ldappasswd` command changes the password of an LDAP entry, identified by an *auth-id* such as `uid=bjensen,ou=people,dc=example,dc=com`, stored by a directory server.

The `ldappasswd` command relies on the Password Modify Extended Operation (OID `1.3.6.1.4.1.4203.1.11.1`).

Options The following options are supported:

`-0`
Ignore LDAP library version mismatches.

When this option is omitted, the default behavior is to assert that the revision number of the LDAP API be greater than or equal to that used to compile the tool. Also, if the library and the tool have the same vendor name, the tool will assert that the vendor version number of the API be greater than or equal to that used to compile the tool. Revision and version numbers are based on the contents of the `LDAPAPIInfo` structure defined in <`ldap.h`> or header files included by <`ldap.h`>.

`-3`
Check host names in SSL certificates.

`-A`
Prompt for old password.

`-D` *bindDN*
Use the specified bind DN to authenticate to the directory server.

If the bind DN and its password are omitted, the `ldappasswd` command binds anonymously.

`-E`
Request that the directory expose (report) the bind identity.

`-H`
`-help`
`--help`
`-?`
Display usage information.

`-I` *filename*
Read SSL key password for the client key database specified using the `-P` option from *filename*.

The default is `key3.db`.

`-J` *controloid*[:*criticality*[:*value*|::*base64value*|:<*fileurl*]]
Use the specified control OID.

The *criticality*, a boolean, is false by default.

An LDAP control can be associated with a value. Proxy authorization takes a proxy authorization ID, for example, passed with the control OID, and criticality. If a value is necessary you specify it using *value*, *base64value*, or <*fileurl*.

`-K` *pathname*
Use the SSL key database located in *pathname*, the full path to the key database file.

The default is to search for the key database file, `key3.db`, in the directory specified by the `-P` option.

`-M`
Manage referrals, modifying the entry containing the referral instead of the entry obtained by following the referral.

`-N` *certificate*
Use the specified *certificate* for certificate-based SSL client authentication, for example: `-N "Client-Cert"`, where `Client-Cert` is the subject name of the user certificate.

`-O` *limit*
Follow at maximum *limit* referral hops.

Default is `5`.

`-P` *pathname*
Use the SSL certificate database located in the specified file system directory.

The default is to search for the certificate database file, `cert8.db`, in the current directory.

`-R`
Do not follow referrals automatically.

`-S`
Prompt for the new password.

`-T` *filename*
Read the new password from the specified file.

`-V` *n*
Use LDAP protocol version *n*, where *n* is `2` or `3`. Default is `3`.

`-W -`
Prompt for the password for the client key database specified using the `-P` option.

The `-W` option is required for certificate-based client authentication.

`-W` *password*

Specify the password for the client key database specified using the `-P` option.

The `-W` option is required for certificate-based client authentication.

`-Y` *proxydn*

Use the rights of the entry having the specified DN for performing LDAP operations. When using this option, you must also specify how to bind before you assume the rights of the proxy. Thus, when using simple authentication, you would also use the `-D` and `-w` options with this option.

Before proxy authentication can work in Directory Server, you must set up the appropriate access control instructions.

`-Z`

Use SSL to provide certificate-based client authentication.

The `-Z` option requires the `-N` and `-W` options and any other SSL options needed to identify the certificate and the key database.

`-ZZ`

Use start TLS when possible to connect to the directory.

`-a` *password*

Use the specified old password.

`-h` *host*

Contact the LDAP server on the specified host, which may be a host name or an IP address. Enclose IPv6 addresses in brackets (`[]`) as described in RFC 2732.

For example, when mapping the IPv4 address `192.168.0.99` to IPv6, pass the `-h` option with its argument as `-h [::ffff:192.168.0.99]`. Notice the brackets.

When using GSSAPI with Directory Server, specify the *host* as a fully-qualified host name which matches the value of the `nsslapd-localhost` attribute on the `cn=config` entry. The GSSAPI authentication process requires that the host name provided by the client match the one provided by the server.

The default is `localhost`.

`-i` *charset*

Use the specified character set to override the value of the `LANG` environment variable. This option is useful, as the command converts certain arguments you specify to UTF-8 before sending the request to the server. The following arguments are converted: base DN, bind DN, LDAP filter, and password.

You can prevent the command from converting passwords by using the `-k` option.

Examples of *charset* values include `ISO8859-1`, `ISO8859-15`, `ibm-1275`, and `windows-1251`.

`-j` *filename*
Read the bind password for simple authentication from the specified file.

`-k`
Do not convert the passwords to UTF-8.

`-m` *pathname*
Use the security module database located in the specified file system directory.

Use the `-m` option if the security module database is in a different directory from the certificate database itself.

`-n`
Show what would be done, but do not actually do it.

`-o` *attrname=attrvalue*
Use the specified attribute values when performing SASL authentication.

The following *attrname* arguments are supported:

authid	Use the specified authentication identity.
authzid	Use the specified authorization identity.
mech	Request the specified SASL mechanism for the bind.
realm	Use the specified realm to complete the bind.
secProp	Use the specified security level.

The *attrvalue* is a valid value corresponding to the *attrname* you specify.

`-p` *port*
Contact the LDAP server on the specified port.

The default is `389` (`636` if SSL is used).

`-s` *password*
Use the specified new password.

`-t` *filename*
Read the old password from the specified file.

`-v`
Run in verbose mode, displaying diagnostics on standard output.

`-w` –
Prompt for the bind password for simple authentication.

`-w` *password*
Use the specified bind password for simple authentication.

Examples Examples in this section use the following conventions:

- The directory server is located on a system named `host`.
- The directory server supports the Password Modify Extended Operation (OID `1.3.6.1.4.1.4203.1.11.1`)
- The directory server listens on port number `389`, the default for non-SSL traffic.
- The directory server listens on port number `636`, the default for SSL traffic. SSL is enabled.

EXAMPLE 1 Changing Your User Password

The following command lets Barbara Jensen change her own user password, connecting over simple authentication:

```
$ ./ldappasswd -h host -D uid=bjensen,ou=people,dc=example,dc=com \
-j old.pwd -T new.pwd -t old.pwd uid=bjensen,ou=people,dc=example,dc=com
ldappasswd: password successfully changed
$
```

EXAMPLE 2 Changing The Password For Another User

The following command lets Kirsten Vaughan change Barbara Jensen's password, connecting over simple authentication:

```
$ ./ldappasswd -h host -D uid=kvaughan,ou=people,dc=example,dc=com \
-w - -A -S uid=bjensen,ou=people,dc=example,dc=com
Old Password:
New Password:
Re-enter new Password:
Enter bind password:
ldappasswd: password successfully changed
$
```

EXAMPLE 3 Using Server Authentication

The following command uses server authentication during the bind, where the server only accepts binds by clients with trusted certificates. Notice only the `-P` option is used without other SSL-related options.

```
$ ./ldappasswd -h host -p 636 -P /home/bjensen/security \
-D "uid=bjensen,ou=People,dc=example,dc=com" -w - -A -S -Z \
uid=bjensen,ou=People,dc=example,dc=com
Old Password:
New Password:
Re-enter new Password:
Enter bind password:
ldappasswd: password successfully changed
$
```

EXAMPLE 4 Using Client Authentication

The following command uses client authentication during the bind, where the server only accepts binds by clients with trusted certificates, and the client must sign the certificate with a password-protected private key. Notice the options used in this example.

```
$ ./ldappasswd -h host -p 636 -A -S -P /home/bjensen/security \
-N "bjscert" -W keypassword uid=bjensen,ou=People,dc=example,dc=com
Old Password:
New Password:
Re-enter new Password:
ldappasswd: password successfully changed
$
```

Exit Status

The exit status returned reflects the return values of the underlying functions used, which may depend on return values sent by the server. Common exit status codes follow:

0 Successful completion; LDAP_SUCCESS; 0x00.

1 Server encountered errors while processing the request; LDAP_OPERATIONS_ERROR; 0x01.

2 Server encountered errors, such as a BER-decoding error, while processing the request; LDAP_PROTOCOL_ERROR; 0x02.

10 Entry to modify belongs to an entry handled by neither server, and the referral URL identifies another server that handles the entry; LDAP_REFERRAL; 0x0a.

32 Authentication ID belongs to an entry not handled by the server, and no referral URL is available for the entry; LDAP_NO_SUCH_OBJECT; 0x20.

50 Bind DN user does not have permission to read the entry from the directory; LDAP_INSUFFICIENT_ACCESS; 0x32.

53 Directory does not allow this user to perform this operation; LDAP_UNWILLING_TO_PERFORM; 0x35.

81 One of the directories did not respond to the request, or the connection was lost; LDAP_SERVER_DOWN; 0x51.

83 The request could not be BER-encoded; LDAP_ENCODING_ERROR; 0x53.

84 A result could not be decoded; LDAP_DECODING_ERROR; 0x54.

89 An option or argument is not valid; LDAP_PARAM_ERROR; 0x59.

91 A specified host name or port is not valid; LDAP_CONNECT_ERROR; 0x5b.

92 At least one server supports only LDAPv2, and the -V 2 option was not used, or the -V 2 option was used, but the server no longer supports LDAP v2; LDAP_NOT_SUPPORTED; 0x5c.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldapcsdk-tools
Stability Level	Evolving

See Also

`ldapcmp`(1), `ldapcompare`(1), `ldapdelete`(1), `ldapmodify`(1), `ldapsearch`(1)

Name ldapsearch – find LDAP entries

Synopsis

```
/opt/SUNWdsee/dsee6/bin/ldapsearch -b baseDN [options] filter
 [attribute]...
```

```
/opt/SUNWdsee/dsee6/bin/ldapsearch  -b baseDN [options] -f filename
 [attribute]...
```

```
install-path/dsrk6/bin/ldapsearch
 -b baseDN [options] filter [attribute]...
```

```
install-path/dsrk6/bin/ldapsearch
 -b baseDN [options] -f filename [attribute]...
```

Description The `ldapsearch` command searches for entries stored by a directory server based on the specified LDAP filter.

The `ldapsearch` command displays results found in LDIF format, including the specified attributes, or all attributes returned if none are specified.

Filter files, which are specified using the `-f` *filename* option, contain one filter per line. Specified LDAP filters must comply with RFC 2254.

Options Unless the `LDAP_BASEDN` environment variable is set, you must at minimum provide a *baseDN* argument to the `-b` option. The *baseDN* argument specifies the distinguished name (DN) of the LDAP entry at the base of the search scope.

The following options are supported:

`-0`
Ignore LDAP library version mismatches.

When this option is omitted, the default behavior is to assert that the revision number of the LDAP API be greater than or equal to that used to compile the tool. Also, if the library and the tool have the same vendor name, the tool will assert that the vendor version number of the API be greater than or equal to that used to compile the tool. Revision and version numbers are based on the contents of the `LDAPAPIInfo` structure defined in <`ldap.h`> or header files included by <`ldap.h`>.

`-1`
Omit leading `version: 1` indication in LDIF output, meaning the output is not RFC 2849 compliant.

`-3`
Check host names in SSL certificates.

`-A`
Display non-ASCII values when the `-v` option is used.

`-C` ps:*changetype*[:*changesonly*[:*entrychangecontrols*]]
Perform a persistent search that stops when you type Control-C.

By default, when used with the `-C` option the `ldapsearch` command requests that the directory server return entry change controls with persistent search results. Adjust this behavior with the following arguments:

changetype
Determines which modifications to an entry are detected and displayed in the output. Possible values include:

- `add`
- `any`
- `delete`
- `modify`
- `moddn`

changesonly
Determines when to display search results. Possible values include:

0
f
false — Display initial search results immediately, not waiting for changes. Then display new changes as they occur.

1 — Display changes when they occur (default).

entrychangecontrols
Determines whether to display entry change controls. Possible values include:

0
f
false — Do not display entry change controls.

1 — Display entry change controls (default).

`-D` *bindDN*
Use the specified bind DN to authenticate to the directory server.

If the bind DN and its password are omitted, the `ldapsearch` command binds anonymously. The bind DN determines what entries and attributes appear in the search results, according to the search permissions for the DN.

`-E`
Request that the directories expose (report) bind identities.

`-F` *sep*
Print specified separator character instead of = between attribute types and values.

`-G` *pattern*
Retrieve a virtual list view displaying a portion of the total search results. Use this option with the `-S` and `-x` options to sort entries returned.

The specified pattern may take one of two forms to specify the size of the virtual list view around a *target entry*:

entriesbefore:*entriesafter*:*value*

Return the target entry, which is the first entry in the sorted results whose sort attribute is greater than or equal to the specified value, as well as the specified number of entries before the target entry and the specified number of entries after the target entry.

For example, `-S sn -x -G 5:10:johnson` returns 16 entries in alphabetical order of the surname attribute: 5 less than `johnson`, the entry equal to or following `johnson`, and the 10 subsequent entries.

entriesbefore:*entriesafter*:*index*:*count*

Return the target entry, as well as the specified number of entries before the target entry and the specified number of entries after the target entry. The target entry depends on the *index* and estimated *count* arguments.

The *count* argument may take the following values, with the following results:

count == 0

The target is the entry at the specified *index* position, starting from 1, and relative to the entire list of sorted results.

count == 1

The target is the first entry in the list of sorted results.

count > 1

The target is the first entry in the slice of the list represented by the fraction *index*/*count*.

Use an *index* argument greater than the *count* argument to target the last result in the list.

For example, `-G 5:10:2:4` specifies the *index* closest to the beginning of the second quarter of the entire list. If the search yielded 100 entries, the target index would be 26, and this pattern would return entries 21 through 36.

The number of entries displayed before and after the target entry may be limited by the beginning and end of the virtual list. The `ldapsearch` command displays the control response, giving the count of entries in the virtual list and the index of the target entry. Use these values to refine *index* and *count* arguments.

`-H`
`-help`
`--help`
`-?`
Display usage information.

`-I` *filename*
Read SSL key password for the client key database specified using the `-P` option from *filename*.

The default is `key3.db`.

`-J` *controloid*[:*criticality*[:*value*|::*base64value*|:<*fileurl*]]
Use the specified control OID.

The *criticality*, a boolean, is false by default.

An LDAP control can be associated with a value. Proxy authorization takes a proxy authorization ID, for example, passed with the control OID, and criticality. If a value is necessary you specify it using *value*, *base64value*, or <*fileurl*.

`-K` *pathname*
Use the SSL key database located in *pathname*, the full path to the key database file.

The default is to search for the key database file, `key3.db`, in the directory specified by the `-P` option.

`-M`
Manage referrals, searching the entry containing the referral instead of the entry obtained by following the referral.

`-N` *certificate*
Use the specified *certificate* for certificate-based client authentication, for example: `-N "Client-Cert"`, where `Client-Cert` is the subject name of the user certificate.

`-O` *limit*
Follow at maximum *limit* referral hops. Default is `5`.

`-P` *filename*
Use the certificate database located in *filename*, the full path to the certificate database file.

The default is to search for the certificate database file, `cert8.db`, in the current directory.

`-Q` [*token*][:*certificate-name*]
Use PKCS 11.

`-R`
Do not follow referrals automatically.

`-S` *attrtype*
Sort the results based on the specified attribute.

`-T`
Do not break long lines within individual attribute values.

Default is to break long attribute values according to LDIF rules.

`-U`
When generating temporary file output using the `-t` option, include URLs as attribute types whose value is a file, such as a photo or certificate.

`-V` *n*
Use LDAP protocol version *n*, where *n* is `2` or `3`. Default is `3`.

`-W -`
Prompt for the password for the client key database specified using the `-P` option.

The `-W` option is required for certificate-based client authentication.

`-W` *password*
Specify the password for the client key database specified using the `-P` option.

The `-W` option is required for certificate-based client authentication.

`-X` *attrlist*
When performing a search to get effective rights using the `-c` option, use the list of attributes provided.

`-Y` *proxydn*
Use the rights of the entry having the specified DN for performing LDAP operations. When using this option, you must also specify how to bind before you assume the rights of the proxy. Thus, when using simple authentication, you would also use the `-D` and `-w` options with this option.

Before proxy authentication can work in Directory Server, you must set up the appropriate access control instructions.

`-Z`
Use SSL to provide certificate-based client authentication.

The `-Z` option requires the `-N` and `-W` options and any other SSL options needed to identify the certificate and the key database.

`-ZZ`
Use Start TLS to provide certificate-based client authentication.

The `-ZZ` option requires the `-N` and `-W` options and any other SSL options needed to identify the certificate and the key database.

`-a` *deref*
Dereference aliases as specified during a search. Possible values for the *deref* argument include the following values:

`always`	Dereference aliases both when finding the base DN, and when searching below it.
`find`	Dereference aliases when finding the base DN.
`never`	Never dereference aliases (default).
`search`	Dereference aliases when searching below the base DN, but not when finding the base DN.

This option has no effect when used with directories that do not support alias dereferencing.

`-c` *authzid*
Use the specified authorization ID when to perform a get effective rights search. The following authorization IDs are supported:

`""`	`""` represents an empty string, meaning use the authorization ID already specified for the operation.
`"`*bindDN*`"`	Use the specified bind DN, such as `uid=bjensen,ou=People,dc=example,dc=com`.
`"dn:"`	Use `anonymous` as the authorization ID.

`-d` *level*
Set LDAP debug level to the specified value.

The following debug levels are supported:

`1`	Display verbose debugging messages; `LDAP_DEBUG_TRACE`.
`2`	Display messages about the content of network packets; `LDAP_DEBUG_PACKETS`.
`320`	Display messages about LDIF parsing; `LDAP_DEBUG_PARSE`.
`16384`	Display informational messages; `LDAP_DEBUG_ANY`.

Use the sum of the levels to specify more than one debug level. For example, to set the debug level to display both verbose debugging messages, and messages about the content of network packets, specify `-d 3`.

`-e`
Minimize base64–encoding of resulting attribute values.

`-f` *filename*
Read the search filters from the specified file.

File format is one search filter per line, where search filters conform to RFC 2254.

`-h` *host*
Contact the LDAP server on the specified host, which may be a host name or an IP address. Enclose IPv6 addresses in brackets (`[]`) as described in RFC 2732.

For example, when mapping the IPv4 address `192.168.0.99` to IPv6, pass the `-h` option with its argument as `-h [::ffff:192.168.0.99]`. Notice the brackets.

When using GSSAPI with Directory Server, specify the *host* as a fully-qualified host name which matches the value of the `nsslapd-localhost` attribute on the `cn=config` entry. The GSSAPI authentication process requires that the host name provided by the client match the one provided by the server.

The default is `localhost`.

`-i` *charset*
Use the specified character set to override the value of the `LANG` environment variable. This option is useful, as the command converts certain arguments you specify to UTF-8 before sending the request to the server. The following arguments are converted: base DN, bind DN, LDAP filter, and password.

You can prevent the command from converting passwords by using the `-k` option.

Examples of *charset* values include `ISO8859-1`, `ISO8859-15`, `ibm-1275`, and `windows-1251`.

`-j` *filename*
Read the bind password for simple authentication from the specified file.

`-k`
Do not convert the passwords to UTF-8.

`-l` *timelimit*
Interrupt the search if the specified time limit is exceeded.

`-m` *pathname*
Use the security module database located in the specified directory.

Use the `-m` option if the security module database is in a different directory from the certificate database itself.

`-n`
Show what would be done, but do not actually do it.

`-o` *attrname=attrvalue*
Use the specified attribute values when performing SASL authentication.

The following *attrname* arguments are supported:

authid	Use the specified authentication identity.
authzid	Use the specified authorization identity.
mech	Request the specified SASL mechanism for the bind.

realm	Use the specified realm to complete the bind.
secProp	Use the specified security level.

The *attrvalue* is a valid value corresponding to the *attrname* you specify.

`-p` *port*
Contact the LDAP server on the specified port.

The default is `389` (`636` if SSL is used).

`-s` *scope*
Use the specified search scope.

The following values are supported for *scope*:

base	Examine only the entry specified by the argument to the `-b` option.
one	Examine only to the entry specified by the argument to the `-b` option and its immediate children.
sub	(Default) Examine the subtree whose base is the entry specified by the argument to the `-b` option.

`-t`
Write a temporary file as output for each attribute of each entry in the search results. Such files are written to the system temporary directory, typically `/tmp`. On standard output, write file names in place of attribute values.

When the `-t` option is used, no base64 encoding is performed on any attribute values, regardless of their content.

`-u`
Include user friendly entry names (`ufn:` *userfriendly*) in the results returned.

`-v`
Run in verbose mode, displaying diagnostics on standard output.

`-w` –
Prompt for the bind password for simple authentication.

`-w` *password*
Use the specified bind password for simple authentication.

`-x`
Have the directory server sort results based on entry DNs before returning the results.

`-z` *sizelimit*
Interrupt the search if the specified maximum number of entries returned is exceeded.

International Searches

This section focuses on international searches, and in particular the matching rule filter portion of the `ldapsearch` command.

When you perform search operations, you can request that the directory sort the results based on any language for which the server has a supported *collation order*.

A *matching rule* provides special guidelines for how the directory compares strings during a search operation. In an international search, the matching rule tells the system what collation order and operator to use when performing the search operation. The syntax of the matching rule filter is as follows.

attr:*matchingRule*:=*value*

Here *attr*, *matchingRule*, and *value* mean the following.

- *attr* is an attribute belonging to entries you're searching for, such as `cn` or `mail`.
- *matchingRule* is a string that identifies either the collation order or the collation order and a relational operator, depending on the format you prefer.
- *value* is either the attribute value for which you want to search or a relational operator plus the attribute value for which you want to search. The syntax of the value portion of the filter depends on the matching rule format you use.

The matching rule portion of a search filter can be represented in one of the following ways.

- Use an OID for the matching rule.

  Each locale supported by Directory Server has an associated collation order OID. Locales supported for Directory Server are listed in the reference documentation on “Identifying Supported Locales” in *Sun Java System Directory Server Enterprise Edition 6.2 Reference*. When you use this approach, the matching rule filter has the following form.

  attr:*OID*:=*relational-operator value*

  The relational operator is included in the value portion of the string, separated from the value by a single space. For example, to search for all `departmentNumber` attributes that are at or after `N4709` in the Swedish collation order, use the following filter.

  ```
  departmentNumber:2.16.840.1.113730.3.3.2.46.1:=>= N4709
  ```
- Use a language tag for the matching rule.

  Each locale supported by Directory Server has an associated language tag. When you use this approach, the matching rule filter has the following form.

  attr:*language-tag*:=*relational-operator value*

  The relational operator is included in the value portion of the string, separated from the value by a single space. For example, to search the directory for all description attributes with a value of `estudiante` using the Spanish collation order, use the following filter.

  ```
  cn:es:== estudiante
  ```
- Use an OID and suffix for the matching rule.

As an alternative to using a relational operator-value pair, you can append a suffix that represents a specific operator to the OID in the matching rule portion of the filter. Combine the OID and suffix.

attr:*OID+suffix*:=*value*

For example, to search for `businessCategory` attributes with the value `Softwareprodukte` in the German collation order, use the following filter.

```
businessCategory:2.16.840.1.113730.3.3.2.7.1.3:=Softwareprodukte
```

The *.3* in the previous example is the equality suffix.

- Use a language tag and suffix for the matching rule.

  As an alternative to using a relational operator-value pair, you can append a suffix that represents a specific operator to the language tag in the matching rule portion of the filter. Combine the language tag and suffix.

  attr:*language-tag+suffix*:=*value*

  For example, to search for all surnames that come at or after `La Salle` in the French collation order, use the following filter.

  ```
  sn:fr.4:=La Salle
  ```

Directory Server supports the following types of international searches, designated in your search filter by adding either the search operator, or the search suffix to the OID or language code specifying the appropriate, collation dependent, matching rule.

equality	Search operator: = Suffix operator: `.1`
less than	Search operator: < Suffix operator: `.2`
less than or equal to	Search operator: <= Suffix operator: `.3`
greater than or equal to	Search operator: >= Suffix operator: `.4`
greater than	Search operator: > Suffix operator: `.5`
substring	Search operator: =* Suffix operator: `.6`

Approximate, or phonetic, and presence searches are supported only in English.

Examples

Examples in this section use the following conventions:

- The directory server is located on a system named `host`.
- The directory server has been configured to support anonymous access for search and read. Therefore, you do not have to specify bind information.
- The directory server listens on port number `389`, the default for non-SSL traffic.
- The directory server listens on port number `636`, the default for SSL traffic. SSL is enabled.

EXAMPLE 1 Returning All Entries

The following command returns all entries in the suffix under the base DN. Use this only when you need to retrieve all entries and attributes:

```
$ ldapsearch -h host -b "dc=example,dc=com" "(objectclass=*)"
```

EXAMPLE 2 Narrowing a Search

The following command employs a more specific filter to narrow the search:

```
$ ldapsearch -h host -b "dc=example,dc=com" "(cn=Babs Jensen)"
```

EXAMPLE 3 Searching the Root DSE Entry

The following command searches the root DSE entry, requesting supported naming contexts and supported LDAP versions. Notice you specify the scope as only the base entry:

```
$ ldapsearch -h host -b "" -s base "(objectclass=*)" \
 namingContexts supportedLDAPVersion
version: 1
dn:
namingContexts: dc=example,dc=com
supportedLDAPVersion: 2
supportedLDAPVersion: 3
```

EXAMPLE 4 Searching the Schema Entry

The following command searches the schema entry, which contains the directory schema. Notice that you can request the operational attribute `subSchemaSubEntry` on any entry to determine which entry holds the schema attributes, in this case `cn=schema`. Then you specify the scope as only the base entry:

```
$ ldapsearch -h host -b "" -s base "(objectclass=*)" subSchemaSubEntry
version: 1
dn:
```

EXAMPLE 4 Searching the Schema Entry *(Continued)*

```
subSchemaSubEntry: cn=schema
$ ldapsearch -h host -b "cn=schema" -s base "(objectclass=*)"
version: 1
dn: cn=schema
...
```

EXAMPLE 5 Setting the Base DN

The following commands set the `LDAP_BASEDN` environment variable, and then use it when searching the directory. The syntax of the first command may not work for your shell. Refer to the documentation about your shell for instructions on setting environment variables.

```
$ LDAP_BASEDN="dc=example,dc=com"; export LDAP_BASEDN
$ ldapsearch -h host "(givenname=Barbara)" cn uid
version: 1

dn: uid=bjablons, ou=People, dc=example,dc=com
cn: Barbara Jablonski
uid: bjablons

dn: uid=bhal2, ou=People, dc=example,dc=com
cn: Barbara Hall
uid: bhal2

dn: uid=bjensen, ou=People, dc=example,dc=com
cn: Barbara Jensen
cn: Babs Jensen
uid: bjensen

dn: uid=bmaddox, ou=People, dc=example,dc=com
cn: Barbara Maddox
uid: bmaddox

dn: uid=bfrancis, ou=People, dc=example,dc=com
cn: Barbara Francis
uid: bfrancis
$
```

EXAMPLE 6 Using a Filter File

The following commands demonstrate use of a filter file. The results show the directory server responds to separate searches for each filter.

EXAMPLE 6 Using a Filter File *(Continued)*

```
$ cat filters
sn=Francis
givenname=Barbara
$ ldapsearch -b "dc=example,dc=com" -h host -f filters cn uid
version: 1

dn: uid=rfrancis, ou=People, dc=example,dc=com
cn: Richard Francis
uid: rfrancis

dn: uid=bfrancis, ou=People, dc=example,dc=com
cn: Barbara Francis
uid: bfrancis

dn: uid=bjablons, ou=People, dc=example,dc=com
cn: Barbara Jablonski
uid: bjablons

dn: uid=bhal2, ou=People, dc=example,dc=com
cn: Barbara Hall
uid: bhal2

dn: uid=bjensen, ou=People, dc=example,dc=com
cn: Barbara Jensen
cn: Babs Jensen
uid: bjensen

dn: uid=bmaddox, ou=People, dc=example,dc=com
cn: Barbara Maddox
uid: bmaddox

dn: uid=bfrancis, ou=People, dc=example,dc=com
cn: Barbara Francis
uid: bfrancis
$
```

EXAMPLE 7 Escaping Commas

The following command demonstrates use of the backslash (\) to escape a comma within a base DN.

```
$ ldapsearch -b "o=Example Company\, Inc.,dc=example,dc=com" \
 -h host "(givenname=Barbara)"
```

SSL Authentication Examples

The following examples demonstrate using SSL authentication for searches.

EXAMPLE 8 Using Server Authentication

The following command uses server authentication during the bind, where the server only accepts binds by clients with trusted certificates. Notice only the `-P` option is used without other SSL-related options.

```
$ ldapsearch -h host -p 636 -b dc=example,dc=com \
 -P /home/bjensen/security -D uid=bjensen,ou=people,dc=example,dc=com \
 -w - "(givenname=Barbara)"
Enter bind password:
```

EXAMPLE 9 Using Client Authentication

The following command uses client authentication during the bind, where the server only accepts binds by clients with trusted certificates, and the client must sign the certificate with a password-protected private key. Notice the options used in this example.

```
$ ldapsearch -h host -p 636 -b dc=example,dc=com \
 -P /home/bjensen/security -N "bjscert" -K /home/bjensen/security \
 -W keypassword "(givenname=Barbara)"
```

International Search Examples

The following examples show search filters used to perform international searches on directory data. Each example gives all the possible matching rule filter formats so that you can become familiar with the formats and select the one that works best for you.

EXAMPLE 10 International Less Than Search

When you perform a locale-specific search using the less than operator (<) or suffix (.1), you search for all attribute values that come before the given attribute in a specific collation order.

Any of the following filters can be used to search for all surnames that come before the surname `Marquez` in the Spanish collation order.

```
sn:2.16.840.1.113730.3.3.2.15.1:=< Marquez
sn:es:=< Marquez
sn:2.16.840.1.113730.3.3.2.15.1.1:=Marquez
sn:es.1:=Marquez
```

EXAMPLE 11 International Less Than or Equal To Search

When you perform a locale-specific search using the less than or equal to operator (<=) or suffix (.2), you search for all attribute values that come at or before the given attribute in a specific collation order.

Any of the following filters can be used to search for all room numbers that come at or before room number `CZ422` in the Hungarian collation order.

EXAMPLE 11 International Less Than or Equal To Search *(Continued)*

```
roomNumber:2.16.840.1.113730.3.3.2.23.1:=<= CZ422
roomNumber:hu:=<= CZ422
roomNumber:2.16.840.1.113730.3.3.2.23.1.2:=CZ422
roomNumber:hu.2:=CZ422
```

EXAMPLE 12 International Equality Search

When you perform a locale-specific search using the equal to operator (=) or suffix (.3), you search for all attribute values that match the given attribute in a specific collation order.

Any of the following filters can be used to search for all `businessCategory` attributes with the value `Softwareprodukte` in the German collation order.

```
businessCategory:2.16.840.1.113730.3.3.2.7.1:== Softwareprodukte
businessCategory:de:== Softwareprodukte
businessCategory:2.16.840.1.113730.3.3.2.7.1.3:=Softwareprodukte
businessCategory:de.3:=Softwareprodukte
```

EXAMPLE 13 International Greater Than or Equal To Search

When you perform a locale-specific search using the greater than or equal to operator (>=) or suffix (.4), you search for all attribute values that come at or after the given attribute in a specific collation order.

Any of the following filters can be used to search for all localities that come at or after `Québec` in the French collation order.

```
locality:2.16.840.1.113730.3.3.2.18.1:=>= Québec
locality:fr:=>= Québec
locality:2.16.840.1.113730.3.3.2.18.1.4:=Québec
locality:fr.4:=Québec
```

EXAMPLE 14 International Greater Than Search

When you perform a locale-specific search using the greater than operator (>) or suffix (.5), you search for all attribute values that come at or before the given attribute in a specific collation order.

Any of the following filters can be used to search for all mail hosts that come after host `schranka4` in the Czech collation order.

```
mailHost:2.16.840.1.113730.3.3.2.5.1:=> schranka4
mailHost:cs:=> schranka4
mailHost:2.16.840.1.113730.3.3.2.5.1.5:=schranka4
```

EXAMPLE 14 International Greater Than Search *(Continued)*

```
mailHost:cs.5:=schranka4
```

EXAMPLE 15 International Substring Search

When you perform an international substring search, you search for all values that match the given pattern in the specified collation order.

Any of the following filters can be used to search for all user IDs that end in `ming` in the Chinese collation order.

```
uid:2.16.840.1.113730.3.3.2.49.1:=* *ming
uid:zh:=* *ming
uid:2.16.840.1.113730.3.3.2.49.1.6*_:=*ming_*
uid:zh.6*_:=*ming_*
```

Exit Status

The exit status returned reflects the return values of the underlying functions used, which may depend on return values sent by the server. Common exit status codes follow:

`0` Successful completion; `LDAP_SUCCESS`; `0x00`.

`1` Server encountered errors while processing the request; `LDAP_OPERATIONS_ERROR`; `0x01`.

`2` Server encountered errors, such as a BER-decoding error, while processing the request; `LDAP_PROTOCOL_ERROR`; `0x02`.

`3` Search exceeded the time limit for operations on the server; `LDAP_TIMELIMIT_EXCEEDED`; `0x03`.

`4` Search returned more results than the maximum number allowed by the server; `LDAP_SIZELIMIT_EXCEEDED`; `0x04`.

`10` Base DN belongs to an entry handled by neither server, and the referral URL identifies another server that handles the entry; `LDAP_REFERRAL`; `0x0a`.

`11` Search returned more results than the maximum number a client application is allowed by the server to retrieve; `LDAP_ADMINLIMIT_EXCEEDED`; `0x0b`.

`32` Base DN belongs to an entry handled by neither server, and no referral URL is available for the entry; `LDAP_NO_SUCH_OBJECT`; `0x20`.

`34` Base DN is not a valid DN; `LDAP_INVALID_DN_SYNTAX`; `0x22`.

`50` Bind DN user does not have permission to read the entry from the directory; `LDAP_INSUFFICIENT_ACCESS`; `0x32`.

`53` Directory is read-only; `LDAP_UNWILLING_TO_PERFORM`; `0x35`.

81 The directory server did not respond to the request, or the connection was lost; `LDAP_SERVER_DOWN`; `0x51`.

82 An error occurred while receiving results; `LDAP_LOCAL_ERROR`; `0x52`.

83 The request could not be BER-encoded; `LDAP_ENCODING_ERROR`; `0x53`.

84 A result could not be decoded; `LDAP_DECODING_ERROR`; `0x54`.

85 The search exceeded the time limit specified using the `-l` option; `LDAP_TIMEOUT`; `0x55`.

87 An error occurred while parsing and BER-encoding the specified filter; `LDAP_FILTER_ERROR`; `0x57`.

89 An option or argument is not valid; `LDAP_PARAM_ERROR`; `0x59`.

90 Needed memory could not be allocated; `LDAP_NO_MEMORY`; `0x5a`.

91 A specified host name or port is not valid; `LDAP_CONNECT_ERROR`; `0x5b`.

92 The directory server supports only LDAPv2, and the `-V 2` option was not used, or the `-V 2` option was used, but the server no longer supports LDAP v2; `LDAP_NOT_SUPPORTED`; `0x5c`.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldapcsdk-tools
Stability Level	Evolving

See Also `ldapcmp`(1), `ldapcompare`(1), `ldapdelete`(1), `ldapmodify`(1), `ldappasswd`(1)

Name ldapsubtdel – recursively delete a subtree of LDAP entries

Synopsis

```
install-path/dsrk6/bin/ldapsubtdel
-b baseDN [options]
```

Description The `ldapsubtdel` command attempts recursively to delete a subtree of LDAP entries under the entry having the distinguished name (DN) specified as a parameter to the `-b` option. You must bind as a user having access to delete the entries specified.

Options The `ldapsubtdel` command supports the following options:

`-b` *DN*
: Delete entries under the entry with the specified DN.

  Default is to delete entries under the specified entry, but not to delete the specified entry itself. Use the `-r` option to delete the specified entry as well.

`-D` *bindDN*
: Use the specified bind DN to authenticate to the directory.

  If the bind DN is not specified, the `ldapsubtdel` command attempts anonymous authentication.

`-H`
: Display a usage message.

`-h` *hostname*
: Connect to the directory on the specified host.

  Default is to connect to the local host on the loopback address, `127.0.0.1`.

`-j` *filename*
: Use the bind password in the specified file to authenticate to the directory.

`-M`
: Manage referrals, deleting the entries containing referrals instead of the entries obtained by following referrals.

  Default is to follow referrals and delete the entries to which the entries in the subtree refer.

`-n`
: Display what would be done, but do not carry out any deletions.

  Default is to carry out the deletions.

`-p` *port*
: Connect to the directory on the specified port.

  Default is to connect to the default simple authentication port for LDAP, `389`.

`-r`
: Also delete the entry having the DN specified as the parameter to the `-b` option.

  Default is not to delete the entry specified.

`-V` *LDAPVersion* Use the specified LDAP version, either 2 or 3.

Default is to use version 3.

`-v` Display verbose output, including information about each deletion performed.

`-w` *password* Use the specified bind password to authenticate to the directory.

`-w -` Prompt for the bind password so it does not appear on the command line.

Examples

The example in this section uses the following conventions:

- The `ldapsubtdel` command is found in a directory present in the `PATH` used for the examples.
- The directory server is located on a system named `host`.
- The directory server listens on port 389, the default for non-SSL connections.

EXAMPLE 1 `ldapsubtdel`: Deleting an Entire Subtree

The following command demonstrates deletion of an entire test subtree of LDAP entries:

```
$ ldapsubtdel -h host -D uid=hmiller,ou=people,dc=example,dc=com -w - \
-b ou=test,dc=example,dc=com -r -v
Enter bind password:
Processing subtree ou=test,dc=example,dc=com
Deleting entry uid=test0,ou=test,dc=example,dc=com
...
Deleting entry uid=test99,ou=test,dc=example,dc=com

Deleting entry ou=test,dc=example,dc=com

Successfully deleted subtree ou=test,dc=example,dc=com
```

If you read `Example.ldif`, you see that `hmiller`'s password is `hillock`.

Exit Status

The `ldapsubtdel` command exits with status `0` if it completes successfully. Otherwise it exits with non-zero status.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	Zip distribution only
Stability Level	Evolving

Name

ldif – format input by adding base64 encoding to make it suitable for inclusion in an LDIF file

Synopsis

install-path/dsee6/bin/ldif
 [-b] *attrtype*

Description

The ldif command formats input by adding base64 encoding to make it suitable for inclusion in an LDIF file. This makes it easy to include binary data, such as JPEG images, along with other textual attribute values. In an LDIF file, base64 encoded attribute values are indicated as ::*encoded data*.

In addition to binary data, other values that must be base64 encoded include any value that begins with a semicolon (;) or a space, and any value that contains non-ASCII data, including newlines. The ldif command takes any input and formats it with the correct line continuation and appropriate attribute information.

Options

The following options are supported:

-b Specifies that the ldif command should interpret the entire input as a single binary value.

As an alternative to the -b option, you can use the :<URL specifier notation, which is simpler to use. For example, jpegphoto:<file:///tmp/myphoto.jpg. Although the official notation requires three /// the use of one / is tolerated.

Exit Status

The following exit values are returned:

0 Successful completion.

1 An error occurred.

On error, verbose error messages are output to standard output.

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-shared
Stability Level	Evolving

See Also

ldapmodify(1)

Name ldifxform – reformat LDIF text

Synopsis *install-path*`/dsrk6/bin/ldifxform`
`[-h] [-i` *input.ldif*`] [-o` *output.ldif*`] -c` *command*`...`

Description The `ldifxform` command reformats LDAP Data Interchange Format (LDIF) text, converting between all of the most common character sets, extracting attribute values, modifying attribute names, ordering entries based on attribute values, or giving detailed statistics. In all cases, input LDIF is not changed.

Options The `ldifxform` command supports the following options:

`-c` *command* Apply the specified reformatting operation.

The `ldifxform` command supports the following reformatting operations:

Attribute Modification The `ldifxform` command can replace attribute values and remove attribute. To modify attributes, use the following options:

`-c suppressoptions` Remove all options other than binary from attribute types.

`-c tcut=`*attribute* Remove the specified attribute from output.

Use this option once for each attribute to remove.

`-c tpreserve=`*attribute* Remove all attributes except the specified attribute from output.

Use this option once for each attribute to retain.

`-c treplace=`*old*`:`*new*

Replace the old attributes with the new attribute type in output.

Use this option once for each attribute to rename.

Character Set Conversions

The `ldifxform` command can convert LDIF from one character set to another. To convert character sets, use the following options:

`-c to=`*charSet*

Convert to the specified character set to UTF-8.

`-c from=`*charSet*

Convert from the specified character set to UTF-8.

The following replacements for *charSet* are supported on all platforms:

`88591`

ISO-8859-1 character set

Conversions to this format strip characters not available in the character set.

`ascii`

ASCII character set

Non ASCII characters are replaced with `?` when converting to this format.

`mstxt`

Windows Unicode Text character set

`t61`

T.61 character set used by X.500 and LDAP v2 servers

Conversions to this format strip characters not available in the character set.

Additional character sets may be supported for your platform. Use the `-h` option to display further character sets supported for your platform.

Directory Analysis

The `ldifxform` command can generate statistical information to help you analyze directory content. To generate statistical information, use the following options:

`-c stats`	Generate statistical information and append it to the output.
`-c statsonly`	Generate statistical information instead of other output.

Sorting and Ordering

Many directory servers return search results in the order that entries were loaded into the database. The `ldifxform` command can sort and order the entries before they are imported into the directory. To sort LDIF, use the following options:

`-c order`	Sort entries into hierarchical order.
`-c sort=`*attribute*	Sort entries in increasing order according to their values for the specified attribute. This is equivalent to alphabetical order for string-valued attributes.
`-c sort=^`*attribute*	Sort entries in decreasing order according to their values for the specified attribute. This is equivalent to reverse alphabetical

order for string-valued attributes.

`-c split=`*number*
: Generate the specified number of LDIF files, which can be loaded into the server by multiple clients in parallel. Each output file has a name of the form *output*`_ldifxform_`*c*`_`*n*, where

output
: Reflects the file name passed to the `-o` option

c
: Corresponds to the number of components in the root DN of the LDIF file

n
: The number of the part from 1 to *number*, inclusive.

Text Transformations

The `ldifxform` command can perform a number of text transformations affecting the presentation and encoding of the LDIF text. To perform text transformations, use the following options:

`-c cleanzero`
: Remove trailing zero bytes from attribute values.

  Use this option when processing LDIF from a buggy encoder.

`-c longlines`
: Do not wrap long lines at the 79th column.

The output can be parsed again, but common tools such as `sed` and `grep` on some platforms may not handle lines longer than 1024 characters.

`-c nob64` Undo base64 encoding.

The output cannot be parsed again if any attributes have values that are binary or that begin with special characters.

`-c nocomments` Remove comments from the output.

`-c nodn` Remove DNs from the output.

The output is no longer LDIF.

`-c notypes` Remove attribute types from the output.

The output is no longer LDIF.

`-c sevenbit` Base64 encode any attribute values containing bytes not present in ASCII.

`-h` Display a usage message briefly describing all options.

`-i` *input.ldif* Read input from the file specified.

When this option is omitted, the `ldifxform` command reads from standard input.

`-o` *output.ldif* Write output to the file specified.

When this option is omitted, the `ldifxform` command writes to standard output.

Extended Description

The `ldifxform` command acts as a stream filter, reading input from one file, performing transformations and writing the output to another file. Each transformation is specified by a *command* parameter to the `-c` option. Multiple compatible transformations may be performed simultaneously.

Some transformations produce LDIF output destined to be reloaded into a directory. For example, renaming an attribute can be more easily processed on an LDIF file than online through requests to a directory server.

Other transformations do not produce LDIF; they are intended to provide an analysis of directory contents. For example, you may extract all different values of a specific attribute and list them under the DN in which they occur. The statistical operations provide counts of entries and attributes.

Examples

The examples in this section use the following conventions:

- The `ldifxform` command is found in a directory present in the `PATH` used for the examples.
- The directory server is located on a system named `host`.
- The directory has been configured to support anonymous access for search and read. Therefore, you do not have to specify bind information.
- The directory server listens on port 389, the default for non-SSL connections.

EXAMPLE 1 `ldifxform`: Transforming Search Results to a List

The following command reformats search results into a simple list of employees placed in order by their room number:

```
$ ldapsearch -h host -b dc=example,dc=com "(uid=*jensen)" | ldifxform \
-c "tpreserve=roomNumber" -c "tpreserve=cn" -c "sort=roomNumber" -c nodn -c notypes
version: 1
#:ordered: TRUE

 Barbara Jensen
 Babs Jensen
 0209

 Allison Jensen
 0784

 Kurt Jensen
 1944

 Richard Jensen
 2631
```

EXAMPLE 1 `ldifxform`: Transforming Search Results to a List *(Continued)*

```
 Gern Jensen
 4609

 Ted Jensen
 4717

 Jody Jensen
 4882
```

EXAMPLE 2 `ldifxform`: Generating Statistical Output

The following command generates statistical output from search results:

```
$ ldapsearch -h host -b dc=example,dc=com "(uid=*jensen)" | ldifxform -c statsonly
# Basic statistics
#:linecount: 121
#:entrycount: 7
# Number of nonleaf entries (at least one subordinate)
#:nonleafcount: 1
# Number of leaf entries (no subordinates)
#:leafcount: 7
# Largest number of entries immediately below a single nonleaf entry
#:maximmsubr: 7
# Number of levels in the DIT hierarchy
#:maxdepth: 4
# Largest number of AVAs in an RDN forming an entry’s DN, normally 1.
#:maxrdns: 1
# Attribute types used in the LDIF file
# e is number entries containing this attr, v is total number of values,
# l is total length, m is max length of any one value, s is general syntax
# and x is extra encoding information.
#:attrstatsinfo: t=description e=1 v=1 l=49 m=49 i=1 s=cis x=ascii
#:attrstatsinfo: t=roomnumber e=7 v=7 l=28 m=4 i=1 s=int
#:attrstatsinfo: t=facsimiletelephonenumber e=7 v=7 l=105 m=15 i=1 s=tel
#:attrstatsinfo: t=telephonenumber e=7 v=7 l=105 m=15 i=1 s=tel
#:attrstatsinfo: t=mail e=7 v=7 l=130 m=19 i=1 s=cis x=mail
#:attrstatsinfo: t=uid e=7 v=7 l=49 m=7 i=1 s=cis x=alphanumeric
#:attrstatsinfo: t=l e=7 v=7 l=71 m=11 i=1 s=cis x=ascii
#:attrstatsinfo: t=ou e=7 v=14 l=144 m=19 i=2 s=cis x=ascii
#:attrstatsinfo: t=objectclass e=7 v=28 l=294 m=20 i=4 s=cis x=alphanumeric
#:attrstatsinfo: t=givenname e=7 v=7 l=36 m=7 i=1 s=cis x=alphanumeric
#:attrstatsinfo: t=sn e=7 v=7 l=42 m=6 i=1 s=cis x=alphanumeric
#:attrstatsinfo: t=cn e=7 v=8 l=96 m=14 i=2 s=cis x=ascii
# Counts of values of specific attribute types
#:attrdomaininfo: t=objectclass v=7 inetOrgPerson
```

EXAMPLE 2 `ldifxform`: Generating Statistical Output *(Continued)*

```
#:attrdomaininfo: t=objectclass v=7 person
#:attrdomaininfo: t=objectclass v=7 top
#:attrdomaininfo: t=objectclass v=7 organizationalPerson
```

Exit Status

The `ldifxform` command exits with status `0` if it completes successfully. Otherwise, it exits with non-zero status.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	Zip distribution only
Stability Level	Evolving

Name

logconv – analyze Directory Server access logs

Synopsis

install-path`/dsrk6/bin/logconv`
`[`*options*`]` *logfile*`...`

Description

The `logconv` command analyzes Directory Server access logs, specified as the *logfile* argument to the command, to extract usage statistics and count occurrences of significant events.

As the `logconv` command depends on the content of the access logs, output depends on the quantity of information present in the access logs. Refer to the Directory Server documentation for instructions on how adjust how much information Directory Server writes to the access logs.

The user running the `logconv` command must have at least read access to the Directory Server log files.

The `logconv` command ignores log files named `access.rotationinfo`.

Options

The `logconv` command supports the following options.

Options specified here without a preceding dash (`-`) may be specified in any order, but must be specified together as a single option such as `-abcefgijlnrtux`.

`-A0` *filename*
Write statistics on client activity based on the number of operations to the specified file.

This option overrides the use of options in the list `-abcefgijlnrtux`.

`-A1` *filename*
Write statistics on client activity based on the number of connections to the specified file.

This option overrides the use of options in the list `-abcefgijlnrtux`.

`-a`
List the most frequently used base DNs.

`-B` *filename*
Write statistics on the most frequently used bind DNs to the specified file.

This option overrides the use of options in the list `-abcefgijlnrtux`.

`-b`
List the most frequently used bind DNs.

`-C` *filename*
Write statistics on the number of operations performed per connection to the specified file.

This option overrides the use of options in the list `-abcefgijlnrtux`.

`-c`
List the number of occurrences for each type of connection code.

`-DELIM`
Generate a field-delimited, formatted report when using the `-B` or `-R` options.

You can import this report into a spreadsheet application.

`-d` *rootDN* Use the specified DN to identify operations performed by Directory Manager.

Default is `cn=Directory Manager`.

`-E` *errorCode* Generate statistics on occurrences of the specified error code.

This option overrides the use of options in the list `-abcefgijlnrtux`.

`-e` List the most frequently occurring error and return codes.

`-f` List the bind DNs with the most failed binds due to invalid credentials.

`-g` List details of all abandoned operations.

`-h` Display the usage message.

`-I` *interval* Use the specified interval for reporting when generating a report using the `-B` or `-R` options. The *interval* may be `MINUTE`, `HOUR`, `DAY`, or `MONTH`.

`-i` List the IP addresses and connection codes for clients opening the most connections.

This option helps detect clients that may attempt to compromise security.

`-j` Generate recommendations based on the data collected.

`-l` List the most frequently occurring search filters.

`-N` Resolve IP addresses to host names.

Using this option may impact performance.

`-n` List the largest and most frequent number of entries per result (`nentries`).

`-P` *filename* Write a report on pending operations to the specified file.

This option overrides the use of options in the list `-abcefgijlnrtux`.

`-R` *filename* Write a report on operations to the specified file.

This option overrides the use of options in the list `-abcefgijlnrtux`.

`-r` List the most frequently requested attributes.

`-s` *number* Return the specified number of results per category.

Default is 20.

`-t` List the longest and most frequent operation times (`etimes`).

`-u` List details about unindexed searches.

`-V` Enable verbose output. Same as `-abcefgijlnrtux`.

`-v` Display version information and exit.

`-X` *IPAddress* Exclude operations originating from clients with the specified IP address, for example when repeated health check operations come from a load balancer.

Repeat this option to exclude multiple addresses.

`-x` List the number and OID of all extended operations requested.

Extended Description

The `logconv` command generates three types of statistics useful for monitoring Directory Server use and optimizing Directory Server configuration:

- Counts of events such as total binds and total searches performed
- Lists of the most frequently occurring parameters in LDAP requests

  For example, the `logconv` command generates lists of the top ten bind DNs, base DNs, filter strings, and attributes returned. As generating such lists is computation intensive, you must explicitly request their generation using the appropriate options.
- Counts of occurrences for error codes such as those defined in `<ldap.h>`

Performance of the `logconv` command is affected by the volume of data in the access logs. To ensure acceptable performance, avoid running the `logconv` command on more than 1 GB of access logs at a time.

Furthermore, some of the data extracted depends on connection and operation numbers reset when you restart Directory Server. To obtain the most accurate counts, avoid analyzing logs that span a server restart.

Examples

Examples in this section use the following conventions:

- The `logconv` command is found in a directory present in the `PATH` used for the examples.
- Directory Server stores access logs in `/var/ds/logs`.
- The current user has read access to the logs.

EXAMPLE 1 `logconv`: Generating Statistics and Recommendations

The following command generates statistics on client connections, binds, abandoned operations, and unindexed searches, and generates recommendations for performance improvements and further investigation:

```
$ logconv -ibgju /var/ds/logs/access*
```

EXAMPLE 2 `logconv`: Examining Binds with Invalid Credentials

The following command counts the number of times clients attempted to bind with invalid credentials, error 49 `LDAP_INVALID_CREDENTIALS`, resolving client IP addresses to host names:

EXAMPLE 2 `logconv`: Examining Binds with Invalid Credentials *(Continued)*

```
$ logconv -N -E 49 /var/ds/logs/access*
```

EXAMPLE 3 `logconv`: Generating a Report

The following command generates a field delimited report on operations, suitable for import into a spreadsheet application:

```
$ logconv -DELIM -R report.txt /var/ds/logs/access
$ cat report.txt
Year|Month|Day|Time|Operations|Results|Performance|Connections|
 Searches|Modifications|Adds|Deletes|Modrdns|Binds|Extended Ops|Compares
2006|Apr|05|07:51:04|18119|18129|100.1%|10|0|0|0|0|0|18119|0|0
2006|Apr|05|08:09:30|12875|12883|100.1%|12878|0|0|0|0|0|12875|0|0
```

Long lines in this example have been wrapped for readability.

Exit Status

The `logconv` command exits with status `0` if it completes successfully. Otherwise it exits with non-zero status.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	Zip distribution only
Stability Level	Evolving

Name makeldif – generate LDIF for import into a directory

Synopsis

```
install-path/dsrk6/bin/makeldif
 [options] -t template -o output.ldif
```

Description The `makeldif` command generates LDAP Data Interchange Format (LDIF) files for import into a Lightweight Directory Access Protocol (LDAP) directory.

Options The `makeldif` command supports the following options:

`-b` *filename* Write bind information to this file when generating LDIF.

Lines of this file include a DN followed by a password, separated by a tab:

DN password

`-C` *delimiter* Use the specified character instead of a comma when reading from a comma-separated format using the `-c` option.

`-c` *filename* Use the specified comma-separated variable format file as input for generating LDIF.

`-D` Run in debug mode, displaying additional information about errors.

`-d` *filename* Write DNs to this file when generating LDIF.

`-F` *filename* Write search filters constructed to find the entries generated to the specified file when generating search filters with the `-T` option.

`-f` *filename* Use the specified file containing a list of first names to use when generating LDIF.

When this option is not used, the `makeldif` command uses the `first.names` file expected in the current directory.

`-H` Display usage information and exit.

`-I` Ignore the first line when reading from a comma-separated format using the `-c` option. Use this option when the initial line is a header not containing data.

`-i` *attribute* Use values of the specified attribute as login IDs when writing login information using the `-L` option.

`-L` *filename* Write login information to this file when generating LDIF.

Lines of this file include a login ID followed by a password, separated by a tab:

loginID   password

Specify the login ID attribute using the `-i` option. Default is `uid`.

`-l` *filename*
Use the specified file containing a list of last names to use when generating LDIF.

When this option is not used, the `makeldif` command uses the `last.names` file expected in the current directory.

`-M`
Generate a separate filter file for each relevant index type when generating search filters with the `-T` option.

`-m` *maximum*
Write no more than the specified maximum number of entries to a single file when generating LDIF.

Default is unlimited.

`-o` *output.ldif*
Create the specified file as output.

`-N` *minimum*
Only create filters that match at least the specified number of entries when generating substring search filters with the `-T` option.

Default is 1.

`-n` *number*
Create substring filters having the specified number of characters when generating substring search filters with the `-T` option.

Default is 3.

`-S`
Skip branch entries (parent entries) when generating LDIF.

`-s` *number*
Use the specified positive integer as a random number generator seed.

Default is to use a seed based on the current time.

You can consistently reproduce the same output by using the same random number generator seed and same templates.

`-T` *attribute*:*types*
Generate search filters of the specified types for the specified attributes.

The *types* is a comma-separated list of the following filter types:

`eq`	Filters matching for equality
`sub`	Filters matching substrings
`subAny`	Filters matching substrings anywhere within the string
`subFinal`	Filters matching substrings at the end of the attribute value
`subInitial`	Filters matching substrings at the beginning of the attribute value

`-t` *template* Use the specified LDIF template file when generating LDIF.

Refer to EXTENDED DESCRIPTION for details.

`-U` Always use UNIX-style newline characters (\n).

`-V` Display version information and exit.

`-w` Wrap long lines when generating LDIF.

Default is to write one attribute type and value per line, potentially resulting in very long lines for some values.

`-X` *maximum* Only create filters that match no more than the specified number of entries when generating substring search filters with the `-T` option.

Default is unlimited.

`-x` *maximum* Write no more than the specified maximum number of entries under each branch for each template when generating LDIF.

Default is unlimited.

Extended Description

The `makeldif` command relies on a template file to customize how entries in the generated LDIF are organized and what they contain. Template files may contain the following definitions:

Global Replacement Definitions — Define strings used to replace variables in the template file itself when generating LDIF

Branch Entry Definitions — Define branches in the directory information tree (DIT) structure

Template Definitions — Define how to generate leaf entries and attribute values

A sample you can customize, `example.template`, is installed with Directory Server Resource Kit.

Global Replacement Definitions

Replacement definitions define strings used to replace variables in the template itself. For example, the following line defines a variable called `suffix` having the value `dc=example,dc=com`:

```
define suffix=dc=example,dc=com
```

Given this definition, all subsequent occurrences of the string `[suffix]` in the template file are replaced with `dc=example,dc=com`. The replacement takes place when lines of the template file are read into memory, with the result that replacements happen even in branch definitions and like places where other tokens are not parsed. Notice that the variable is surrounded by brackets, `[]`. When using brackets for purposes other than delimiting global replacement variables, escape them with a backslash, as in `\[` or `\]`. The backslash characters are removed during LDIF generation.

Branch Entry Definitions

Branch entries are parents for other entries in a suffix. In other words, the branch entry at the root of the suffix for `Example.com` might be defined as:

```
branch: dc=example,dc=com
```

The `makeldif` command can then generate a corresponding branch entry represented in LDIF as follows:

```
dn: dc=example,dc=com
objectclass: top
objectclass: domain
dc=example
```

The `makeldif` command determines which object classes to use by examining the RDN of the entry, recognizing attribute types `c` (country), `dc` (domain component), `l` (location), `o` (organization), and `ou` (organizational unit). When you use RDNs having other attribute types, the `makeldif` command uses the object class `extensibleObject` for the entry.

To customize the branch entry itself, define additional attributes directly below the branch definition. For example, add a `description` attribute for the branch entry as follows:

```
branch: dc=example,dc=com
description: This is the description.
```

The resulting entry generated in LDIF appears as follows:

```
dn: dc=example,dc=com
objectclass: top
objectclass: domain
dc=example
description: This is the description.
```

To enable generation of entries below the branch entry, add `subordinateTemplate` definitions directly below the branch definition. For example, add a 1000 entries using the `person` template below `ou=people,dc=example,dc=com` as follows:

```
branch: ou=people,dc=example,dc=com
subordinateTemplate: person: 1000
```

You can add multiple `subordinateTemplate` definitions. For example, enable addition of 1000 entries using the `person` template and 500 entries using the `personWithCertificate` template below `ou=people,dc=example,dc=com` as follows:

```
branch: ou=people,dc=example,dc=com
subordinateTemplate: person: 1000
subordinateTemplate: personWithCertificate: 500
```

Template Definitions

Template definitions contain prototype entries with special tags allowing the `makeldif` command to generate many unique, custom entries. For example, a `person` template definition might appear as follows:

```
template: person
rdnAttr: uid
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
givenName: <first>
sn: <last>
cn: {givenName} {sn}
uid: {givenName}.{sn}
mail: {uid}@example.com
userPassword: <random:alphanumeric:8>
telephoneNumber: <random:telephone>
```

The first line of a template definition specifies the name of the template, here `person`. The `makeldif` command uses the name to identify the template when creating leaf entries under branch entries, based on `subordinateTemplate` definitions used with the branch entry definition. Each name must be unique.

A template entry may also have an `rdnAttr` line specifying the attribute type for the RDN of the generated entry. The `rdnAttr` takes a single value. Multi-valued RDNs are not supported. The default `rdnAttr` definition is `cn` if you do not provide one.

Other lines in a template definition reflect the attribute types and values to generate in the resulting LDIF. The `makeldif` command generates values for all recognized tokens.

Supported Attribute Value Tokens

The `makeldif` command support the following tokens:

`<ancestordn:{`*depth*`}>`

Replace this value with the DN of the entry's ancestor at the specified depth.

A depth of 1 specifies the parent entry; a depth of 2 specifies the grandparent, and so forth. If the entry does not have an ancestor at the specified depth, the `makeldif` command replaces the value with an empty string.

`<base64:{`*value*`}>`

Replace this value with a base64-encoded representation of the specified value.

The value is decoded to a byte array using the UTF-8 character set, and then the byte array is base64-encoded.

`<base64:{`*charset*`}:{`*value*`}>`

Replace this value with a base64-encoded representation of the specified value.

The value is decoded to a byte array using the specified character set, and then the byte array is base64-encoded.

`<dn>`

Replace this value with the DN of the current entry.

The RDN attribute for the entry must be assigned a value in the template before this token is used.

`<exec:{`*command*`}>`

Replace this value with the information sent to standard output when the specified command is executed on the system.

The replacement invokes a separate process each entry created using this template. Using this token can therefore slow LDIF generation considerably.

`<exec:{`*command*`},{`*arg1*`},{`*arg2*`},…,{`*argN*`}>`

Replace this value with the information sent to standard output when the specified command is executed on the system using the arguments provided.

The replacement invokes a separate process each entry created using this template. Using this token can therefore slow LDIF generation considerably.

`<file:{`*filename*`}>`

Replace this value with a randomly-chosen value from the specified file.

The file must contain one value per line. Weights cannot be assigned to the values in a file. To weight values, repeat their lines multiple times in the file.

`<first>`

Replace this value with a first name from the first name file specified using the `-f` option.

If both a first and last name are included in an entry, the combination of the first and last name is guaranteed to be unique. That is, no two entries in the generated LDIF file have the same combination of first and last name values. In order to guarantee uniqueness, the first and last names must be used in their entirety. You cannot use substrings of the form `{givenName:5}` for example.

`<guid>`

Replace this value with a GUID value in the containing hexadecimal digits in the form `12345678-90ab-cdef-1234-567890abcdef`.

GUID values generated are unique within the LDIF generated.

`<ifabsent:{`*attribute*`}>`

Include this attribute only if the specified attribute is not present on the entry.

The specified attribute must be defined in the template file before it is referenced in the `ifabsent` tag.

`<ifabsent:{`*attribute*`}:{`*value*`}>`
Include this attribute only if the specified attribute is not present on the entry or if it does not have the specified value.

The specified attribute must be defined in the template file before it is referenced in the `ifabsent` tag. If the specified attribute has multiple values, the `makeldif` command checks only the first value.

`<ifpresent:{`*attribute*`}>`
Include this attribute only if the specified attribute is also present on the entry.

The specified attribute must be defined in the template file before it is referenced in the `ifpresent` tag.

`<ifpresent:{`*attribute*`}:{`*value*`}>`
Include this attribute only if the specified attribute is also present on the entry and has the specified value.

The specified attribute must be defined in the template file before it is referenced in the `ifpresent` tag. If the specified attribute has multiple values, the `makeldif` command checks only the first value.

`<last>`
Replace this value with a last name from the last name file specified using the `-l` option.

If both a first and last name are included in an entry, the combination of the first and last name is guaranteed to be unique. That is, no two entries in the generated LDIF file have the same combination of first and last name values. In order to guarantee uniqueness, the first and last names must be used in their entirety. You cannot use substrings of the form `{givenName:5}` for example.

`<list:{`*value1*`},{`*value2*`},…,{`*valueN*`}>`
Replace this value with a randomly-chosen value from the specified, comma-delimited list.

Each value has an equal chance of being chosen.

`<list:{`*value1*`}:{`*weight1*`},{`*value2*`}:{`*weight2*`},…,{`*valueN*`}:{`*weightN*`}>`
Replace this value with a randomly-chosen value from the specified, comma-delimited list.

The weight associated with each list item determines how likely that value is to be chosen. A list item with a weight of 2 is twice as likely to be chosen as an item with a weight of 1. Specified only positive integer weights.

`<loop:{`*start*`}:{`*end*`}>`
Process this definition (*end* - *start* + 1) times, replacing this token each time with a number beginning at {*start*} and incrementing by one until reaching {*end*}.

You may include multiple `loop` tokens on the same line and using different {*start*} values, but only the first {*end*} value is used to determine how many copies of the line to create.

`<parentdn>`
Replace this value with the DN of the parent entry.

`<presence:`{*percent*}`>`
Include the attribute on the specified percentage of entries generated from this template definition. The percentage value is a number between 0 and 100.

Use this token only with attributes not required by the entry's object classes, and include something in the value of the attribute to be generated on entries including the attribute.

`<random:alpha:`{*length*}`>`
Replace this value with a string of {*length*} randomly-chosen alphabetic characters.

`<random:alpha:`{*minlength*}`:`{*maxlength*}`>`
Replace this value with a string of between {*minlength*} and {*maxlength*} randomly-chosen alphabetic characters.

`<random:alphanumeric:`{*length*}`>`
Replace this value with a string of {*length*} randomly-chosen alphanumeric characters.

`<random:alphanumeric:`{*minlength*}`:`{*maxlength*}`>`
Replace this value with a string of between {*minlength*} `and` {*maxlength*} `randomly-chosen alphanumeric characters.`

`<random:base64:`{*length*}`>`
Replace this value with a string of {*length*} randomly-chosen base64 characters.

If the specified length is not a multiple of 4, then the base64 value produced is padded with equal signs so that the total length is a multiple of 4.

`<random:base64:`{*minlength*}`:`{*maxlength*}`>`
Replace this value with a string of between {*minlength*} and {*maxlength*} randomly-chosen base64 characters.

`<random:chars:`{*characters*}`:`{*length*}`>`
Replace this value with a string of {*length*} characters that are randomly-selected from {*characters*}. {*characters*} may be any valid character other than the colon.

`<random:hex:`{*length*}`>`
Replace this value with a string of {*length*} randomly-chosen hexadecimal digits.

`<random:hex:`{*minlength*}`:`{*maxlength*}`>`
Replace this value with a string of between {*minlength*} and {*maxlength*} randomly-chosen hexadecimal digits.

`<random:month>`
Replace this value with the name of a randomly-chosen month. That is, t

The value is one of `January`, `February`, `March`, `April`, `May`, `June`, `July`, `August`, `September`, `October`, `November`, or `December`.

`<random:month:{`*length*`}>`
Replace this value with the first `{`*length*`}` characters of the name of a randomly-chosen month.

`<random:numeric:{`*length*`}>`
Replace this value with a string of `{`*length*`}` randomly-chosen numeric digits.

`<random:numeric:{`*min*`}:{`*max*`}>`
Replace this value with a randomly-chosen number between `{`*min*`}` and `{`*max*`}`, inclusive.

`<random:numeric:{`*min*`}:{`*max*`}:{`*length*`}>`
Replace this value with a randomly-chosen number between `{`*min*`}` and `{`*max*`}`, inclusive.

The value is padded with leading zeros so that it has at least `{`*length*`}` digits.

`<random:telephone>`
Replace this value with a string of randomly-chosen numeric digits in the form 123-456-7890.

This uses a US-format telephone number. You can generate telephone numbers in the format used by other countries by combining other random tags. For example, to generate a telephone number in the UK format, use `+44 <random:numeric:4> <random:numeric:6>`.

`<sequential>`
Replace this value with a sequentially-increasing numeric value.

The first value is zero. Sequential counters are separate on a per-attribute basis, so it is possible to use multiple sequential counters in the different attributes of the same entry without impacting each other.

`<sequential:{`*firstvalue*`}>`
Replace this value with a sequentially-increasing numeric value, where the first number starts at the specified value.

Sequential counters are separate on a per-attribute basis, so it is possible to use multiple sequential counters in different attributes of the same entry without impacting each other.

In addition to supported tokens, you can cause the `makeldif` command to generate attribute values from the values of attributes on the entry previously defined in the template by constructing prototype values using those attribute types in braces. For example, the following excerpt reuses `givenName` and `sn` (surname) values to define `cn` (common name) values:

```
...
givenName: <first>
sn: <last>
cn: {givenName} {sn}
...
```

When generating values from multi-valued attributes, the `makeldif` command uses the first value in the list.

To use only the first few characters of an attribute value to generate a value, add a colon followed by the length of the substring to use. For example, use `{givenName:1}{sn:1}{employeeNumber}` to generate values taking the first letter of the first name, followed by the first letter of the last name, followed by the employee number.

Subordinate Template Definitions

To create entries generated from one template definition below those generated by another template definition, include one or more `subordinateTemplate` definitions in the upper template definition.

Use this functionality with caution, however, as the `makeldif` command does not prevent you from generating circular references throwing the LDIF generation process into an infinite loop.

Inheritance

Template definitions support inheritance, whereby you specify a template definition that builds on a previously defined template, using the `extends` definition.

For example, to generate 10000 entries using the `person` template and an additional 1000 entries having the same structure as those generated from the `person` but also including a value for the `userCertificate` attribute, you create a template definition extending the `person` template as follows:

```
template: certificatePerson
rdnAttr: uid
extends: person
userCertificate: <random:base64:1000>
```

Given the `person` template defined previously, the `certificatePerson` template then has the same effect as the following:

```
template: certificatePerson
rdnAttr: uid
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
givenName: <first>
sn: <last>
cn: {givenName} {sn}
uid: {givenName}.{sn}
mail: {uid}@example.com
userPassword: <random:alphanumeric:8>
telephoneNumber: <random:telephone>
userCertificate: <random:base64:1000>
```

You may use multiple levels of inheritance, but you must make sure both to specify the rdnAttr value for the inherited template as the parent's RDN attribute is not automatically used, and to avoid circular references that cause infinite loops in the LDIF generation process.

Examples

Examples in this section use the following conventions:

- The makeldif command is found in a directory present in the PATH used for the examples.
- The sample files are located in the current directory.

EXAMPLE 1 makeldif: Generating LDIF

The following command generates LDIF using the sample template and other files delivered with Directory Server Resource Kit.

```
$ makeldif -t example.template -o sample.ldif
Processed 1000 entries
Processed 2000 entries
Processed 3000 entries
Processed 4000 entries
Processed 5000 entries
Processed 6000 entries
Processed 7000 entries
Processed 8000 entries
Processed 9000 entries
Processed 10000 entries
Processing complete.
10002 total entries written.
```

EXAMPLE 2 makeldif: Generating Search Filters and LDIF

The following command generates LDIF and corresponding search filters base.

```
$ makeldif -T uid:eq -T cn:eq,sub -F filters.txt -t example.template -o sample.ldif
Processed 1000 entries
Processed 2000 entries
Processed 3000 entries
Processed 4000 entries
Processed 5000 entries
Processed 6000 entries
Processed 7000 entries
Processed 8000 entries
Processed 9000 entries
Processed 10000 entries
Processing complete.
10002 total entries written.
Writing filters to filters.txt
Wrote 10000 equality filters for uid
Wrote 10000 equality filters for cn
```

EXAMPLE 2 `makeldif`: Generating Search Filters and LDIF *(Continued)*

```
Wrote 1827 subInitial filters for cn
Wrote 7328 subAny filters for cn
Wrote 2099 subFinal filters for cn
```

Exit Status

The `makeldif` command exits with status `0` if it completes successfully. Otherwise it exits with non-zero status.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	Zip distribution only
Stability Level	Evolving

Name

mmldif – combine multiple ldif files into a single, authoritative set of entries

Synopsis

install-path`/ds6/bin/mmldif`
 `[-c ] [-D ] [-o` *out.ldif*`]` *files*

Description

The `mmldif` command combines multiple LDIF files into a single authoritative set of entries. Typically each LDIF file is from a master server cooperating in a multi-master replication environment (for example, masters that refuse to sync up). Optionally, the `mmldif` command can generate LDIF change files that could be applied to the original file to bring it up to date with the authoritative version. At least two input files must be specified.

Options

The following options are supported:

`-c` Write a change file (`.delta`) for each input file.

`-D` Print debugging information.

`-o` Write authoritative data to this file. If not specified, the command compares the input files, but does not generate output LDIF files.

files Two or more LDIF files to combine into a single set of entries. For example, `in1.ldif in2.ldif`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

See Also

`insync`(1)

Name modrate – measure modification performance for an LDAP directory

Synopsis

```
install-path/dsrk6/bin/modrate
 [options] -b baseDN -M attribute:length:regexp
```

Description The `modrate` command measures the rate at which an LDAP directory can perform random, user-defined modifications. As with all measures of performance, results depend on many factors, including what options you pass to the `modrate` command, and also how the directory service itself is tuned.

The command uses LDAP v3, and cannot be used to authenticate to an LDAP v2 directory not supporting LDAP v3.

Options The `modrate` command supports the following options:

`-a`
: Run in asynchronous mode, not waiting for results before requesting subsequent modifications. The maximum number of threads the `modrate` command can use is limited by the number of file descriptors the operating system allows the process to use. The time is measured starts when the request is sent and finishes when the result is received.

`-b` *targetDN*
: Use the specified DN for the target entry.

Refer to Random Target Syntax and Random Target Substitution for details.

`-C` *messages*
: Display the specified number of results messages before exiting. Results messages appear by default as output on standard out, similar to the following:

```
Avg r= 272.00/thr ( 54.40/sec), total=   816
```

This shows output for three threads requesting modifications for five seconds. The average modify rate per thread is 54.40 per thread per second for the interval measured. The total shown for all threads is 816.

Default is to continue iterating until the command is interrupted.

`-D` *bindDN*
: Use the specified bind DN to authenticate to the directory.

If the bind DN is not specified, the `modrate` command attempts anonymous authentication.

`-E`
: Display the bind DN of entries for which modifications did not complete successfully.

`-e`
Display the number of attempted modifications that did not complete successfully.

`-h` *hostname*
Connect to the directory on the specified host.

Enclose IPv6 addresses in brackets (`[]`) as described in RFC 2732.

Default is to connect to the local host on the loopback address, `127.0.0.1`.

`-i` *filename*
Use the file specified to generate target entry base DNs at random.

Refer to Random Target Syntax and Random Target Substitution for details.

`-j` *seconds*
Display results each specified number of seconds.

Default is to display results every 5 seconds.

`-K`
Keep connections open and only bind once, measuring only the time required to perform the modify operation.

Default is to measure the duration the connection is active as the modification sequence.

`-k`
Keep connections open, measuring only the time required to perform the bind and modify operations.

Default is to measure the duration the connection is active as the modification sequence.

`-M` *attribute*:*length*:*regexp*
Generate random values for modifications on the specified attribute, having the specified integer length in characters. Generate the values from the specified regular expression, *regexp*, which has the form `(c*(c-c)*)*` where `c` represents an ASCII character.

For example, the *regexp* parameter could be `[A-Z][a-z][0-9]`, or simply `aString`

If the attribute specified does not exist on the target entry, it is added, subject to schema checking.

`-m` *maxIter*
Perform no more than the specified number of modifications per thread.

Default is for each thread to continue iterating until the command is interrupted.

`-O` *maxHops*

Traverse no more than the specified number of hops when following referrals.

Default is 5.

`-p` *port*

Connect to the directory on the specified port.

Default is to connect to the default simple authentication port for LDAP, `389`.

`-q`

Run in quiet mode, not displaying results.

Default is to display results every 5 seconds, which you can adjust using the `-j` option.

`-R`

Do not follow referrals.

Default is to follow referrals.

`-r` *maxRand*

Use the specified maximum to determine the range for random numbers replacing `%d` formatting specifications when modifying random target entries.

When you use this option twice, the first occurrence generates random numbers in the range [0,*maxRand1*-1] for the first `%d`, the second [1,*maxRand2*] for the second `%d`.

Refer to Random Target Syntax and Random Target Substitution for details.

`-S` *randSeed*

Use the specified seed, an `unsigned int`, for random number generation.

Default seed is `0`.

`-t` *threads*

Use the specified number of the threads to connect to the server.

Default is to use one thread.

`-v`

Display verbose output.

`-W` *filename*

Read the bind password from the specified file.

`-w` *password*

Use the specified bind password to authenticate to the directory.

`-w –`

Prompt for the bind password so it does not appear on the command line or in a file.

Extended Description

The modrate command repeatedly requests modification operations of a directory server. Threads may be configured to keep open connections or perform LDAP bind with each operation. The command-line options let you specify the bind credentials.

The command uses LDAP v3, and cannot be used to authenticate to an LDAP v2 directory not supporting LDAP v3. Furthermore, the modrate command uses simple authentication, not secure binding.

The modrate command cannot set a time limit for operations.

By default, the modrate command continues its task indefinitely, displaying results periodically, and displaying any errors encountered as well without interrupting operation.

Random Target Syntax

Include randomly generated numbers by specifying %d and %s placeholders in the base DN. These placeholders are then replaced according to the following rules:

%d
: Replace this placeholder with random integer values depending on the *maxRand* parameter to the -r option.

  The -r option may be used at most two times to generate random target entries. Replacement values for the %d placeholder range over [0,*maxRand1*—1] for the first use of the -r option, and over [1,*maxRand2*] for the second.

%s
: Replace this placeholder with random strings from the file specified using the -i option.

  Replacement values for this placeholder are randomly selected lines of the file specified.

Random Target Substitution

The modrate command requires that you apply the following rules for substitutions, displaying an error message when the used incorrectly:

- Use only one type of placeholder, either %d or %s, per invocation of the modrate command.
- Specify at least as many uses of the -r as %d placeholders used in the base DN.
- Use %%d and %%s to specify literal strings %d and %s, respectively.

In order to use this random modification mechanism, you must populate your directory accordingly. For example, you can measure the modification rate using the following command:

```
$ modrate -D "uid=test%d,ou=test,dc=example,dc=com" -w "auth%d%d" -r 100
```

In order for the modrate command to bind effectively, your directory must contain entries corresponding to the following LDIF excerpt:

```
dn: uid=test0,ou=test,dc=example,dc=com
userPassword: auth00

dn: uid=test1,ou=test,dc=example,dc=com
```

```
userPassword: auth11

dn: uid=test2,ou=test,dc=example,dc=com
userPassword: auth22
...

dn: uid=test10,ou=test,dc=example,dc=com
userPassword: auth1010
...

dn: uid=test99,ou=test,dc=example,dc=com
userPassword: auth9999
```

Examples

Examples in this section use the following conventions:

- The modrate command is found in a directory present in the PATH used for the examples.
- The directory server is located on a system named host.
- The directory server listens on port 389, the default for non-SSL connections.

EXAMPLE 1 modrate: Sample Output

The following command performs modifications until it has displayed five results messages. Notice that each line concerns only the elapsed interval.

```
$ modrate -h host -D uid=hmiller,ou=people,dc=example,dc=com -w - \
-C 5 -b "uid=test%d,ou=test,dc=example,dc=com" -r 100 -M "description:7:aString"
Enter bind password:
Avg r=  74.00/thr ( 14.80/sec), total=    74
Avg r= 118.00/thr ( 23.60/sec), total=   118
Avg r=  68.00/thr ( 13.60/sec), total=    68
Avg r=  39.00/thr (  7.80/sec), total=    39
Avg r=  71.00/thr ( 14.20/sec), total=    71
All threads exited
```

If you read Example.ldif, you see that hmiller's password is hillock.

Notice also that a result message provides the following items of information:

- The average rate of modification per thread of execution
- The average rate of modification per second
- The total number of modification operations performed during the interval the results message concerns

EXAMPLE 2 modrate: Modification Rate Alone

The following command keeping the connection open and binds only once:

EXAMPLE 2 `modrate`: Modification Rate Alone *(Continued)*

```
$ modrate -h host -D uid=hmiller,ou=people,dc=example,dc=com -w - \
-C 5 -b "uid=test%d,ou=test,dc=example,dc=com" -r 100 -M "description:7:aString" -K
Enter bind password:
Avg r= 272.00/thr ( 54.40/sec), total=   272
Avg r= 183.00/thr ( 36.60/sec), total=   183
Avg r= 180.00/thr ( 36.00/sec), total=   180
Avg r= 257.00/thr ( 51.40/sec), total=   257
Avg r= 226.00/thr ( 45.20/sec), total=   226
All threads exited
```

If you read `Example.ldif`, you see that `hmiller`'s password is `hillock`.

Exit Status

The `modrate` command returns the following exit status codes.

`0`	Successful completion.
non-zero	An error occurred.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	Zip distribution only
Stability Level	Evolving

See Also

authrate(1), makeldif(1), searchrate(1)

Name

pwdhash – print the encrypted form of a password by using one of the server's encryption algorithms

Synopsis

install-path`/ds6/bin/pwdhash`
`-D` *instance-path* `[-H ] [-c` *comparepwd* | `-s` *scheme*`]` *password*`...`

Description

The `pwdhash` command prints the encrypted form of a password using one of the encryption algorithms available to the server. If a user cannot log in, you can use this command to compare the user's password with the password stored in the directory.

Options

The following options are supported:

`-c` Specifies the encrypted password with which the user password is to be compared. The result of this comparison is either `OK` or `password does not match`.

`-D` *instance-path* Specifies where the Directory Server instance is located.

`-H` Specifies that the passwords are hex-encoded.

password The clear password from which the encrypted form should be generated (or against which the password in the directory should be compared).

`-s` Generates the encrypted passwords according to the encryption scheme. The available schemes are `SSHA`, `SHA`, `CRYPT`, and `CLEAR`.

Examples

EXAMPLE 1 Encrypting a Password

```
$ pwdhash -D /local/ds -s SSHA mypassword
{SSHA}mtHyZSHfhOZ4FHmvQe09FQjvLZpnW1wbmW05cw==
```

EXAMPLE 2 Comparing Two Passwords

```
$ pwdhash -D /local/ds \
-c "{SSHA}mtHyZSHfhOZ4FHmvQe09FQjvLZpnW1wbmW05cw==" aPassword
pwdhash: password does not match
```

Exit Status

The following exit values are returned:

`0` Successful completion.

`1` An error occurred.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Stable

Name

repldisc – discover a replication topology

Synopsis

```
install-path/ds6/bin/repldisc
 [-D bindDN] [-w password] [-j file ] [-t ] [-n ] [-a ]
 [-p port] [-T timeout] [-J file] [-W keypassword] [-K keydbpath]
 [-N certname] [-P certdbpath] [-e SSL port] [-b ReplicaRoot]
 -s |  -S  HostSpec
```

Description

The `repldisc` command enables the discovery of a replication topology. Topology discovery starts with one server and constructs a graph of all known servers (using the RUVs and Replication Agreements). `repldisc` then prints an adjacency matrix describing the topology.

Options

The following options are supported:

`-a` Specifies that only the arcs between pairs of connected hosts are printed. For more information, see EXAMPLES.

Note – If the total line length of the output exceeds 80 characters, symbolic host names are used, accompanied by a legend. Otherwise, full host names are printed. Using the `-a` option ensures that symbolic host names are not used.

`-b` The suffix (replica root) that has been specified for replication. If `-b` is not specified, the delay for all suffixes is printed.

`-D` Distinguished name with which to bind to the server. This parameter is optional if the server is configured to support anonymous access. If a DN is specified in the *HostSpec* option, this overrides the `-D` option.

`-j` If specifying the default password at the command-line poses a security risk, the password can be stored in a file. The `-j` option specifies this file.

`-n` Specifies that `repldisc` should not run in interactive mode. Running in interactive mode allows you to re-enter the bindDN, password, host and port, if a bind error occurs.

`-p` The TCP port used by the instance. The default port is 389. If a port is specified in the *HostSpec*, this overrides the `-p` option.

`-t` Prints the mode of transport (`SSL` or `CLEAR`).

`-T` Specifies the number of seconds after which `repldisc` times out if the server connection goes down.

`-w` Password associated with the distinguished name specified by the `-D` option. If a password is specified in the *HostSpec*, this overrides the `-w` option.

HostSpec Host specification, which takes the form `[binddn:[password]@]` *host*`[:port]`. The following is an example:

```
cn=admin,cn=Administrators,cn=config:mypword@myserver:1389
```

If you are using SSL, use `-S` in the server specification. In this case, *HostSpec* specifies the certificate name and key password, rather than the bindDN and password.

Ssl Options

You can use the following options to specify that `repldisc` uses LDAPS when communicating with Directory Server. You can also use these options if you want to use certificate-based authentication. These options are valid only when LDAPS has been turned on and configured.

`-e` Default SSL port, 636.

`-J` This option has the same function as the `-j` option, for the key password.

`-K` Specifies the name of the certificate key used for certificate-based client authentication. For example, `-K` *Server-Key*.

`-N` Specifies the certificate name to use for certificate-based client authentication. For example, `- N` *Server-Cert*. If this option is specified, the `-W` option is required.

`-P` Specifies the location of the certificate database.

`-W` Specifies the password for the certificate database identified by the `-P` option. For example, `-W` *serverpassword*.

Examples

EXAMPLE 1 Single Replication Scenario

```
$ repldisc -D cn=admin,cn=Administrators,cn=config -w pwd \
 -b o=rtest -s myserver:1389

Topology for suffix: o=rtest

Legend:

^ : Host on row sends to host on column.

v : Host on row receives from host on column.

x : Host on row and host on column are in MM mode.

H1 : france.example.com:1389

H2 : spain:1389

H3 : portugal:389

   | H1 | H2 | H3 |

===+===============

H1 |    | ^  |    |
```

EXAMPLE 1 Single Replication Scenario *(Continued)*

```
---+---------------

H2 | v  |    | ^  |

---+---------------

H3 |    | v  |    |

---+---------------
```

EXAMPLE 2 Using the -a Option

```
Topology for suffix: o=rtest

Legend:

The direction of the replication is indicated with arrows.

Single-master: suppliers appear on left, consumers on right (->).

Multi-master : servers are shown linked by a double arrow (<->).


france.example.com:1389 -> spain:1389

spain:1389 -> portugal:389
```

Exit Status

The following exit values are returned:

`0` Successful completion.

`1` An error occurred.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Stable

See Also

`insync`(1), `entrycmp`(1)

Notes

The node on which you are running the `entrycmp`, `insync`, and `repldisc` tools must be able to reach all the specified hosts. If these hosts are unavailable, you will encounter difficulties using these tools. Ensure that all servers are up and running before using these tools.

When you identify hosts, you must use either symbolic names or IP addresses for all hosts. The replication monitoring commands do not address resolution between symbolic names and IP addresses. Using a combination of symbolic names and IP addresses can cause problems. Moreover, on multi-homed hosts, referring to the same Directory Server instance using different names may cause unexpected results.

When SSL is enabled, the directory server on which you are running the tools must have a copy of all the certificates used by the other servers in the topology.

`repldisc` takes the host specification from the replication agreement, unless otherwise specified at the command line.

The replication monitoring tools rely on access to `cn=config` to obtain the replication status. This should be taken into account, particularly when replication is configured over SSL.

Name searchrate – measure search performance for an LDAP directory

Synopsis *install-path*`/dsrk6/bin/searchrate`
 `[`*options*`] -b` *baseDN* `-f` *filter*

Description The `searchrate` command measures the rate at which an LDAP directory can perform random, user-defined searches. As with all measures of performance, results depend on many factors, including what options you pass to the `searchrate` command, and also how the directory service itself is tuned.

Options The `searchrate` command supports the following options:

`-A` *attribute* Retrieve only the specified attribute.

Repeat this option to specify multiple attributes.

`-a` Run in asynchronous mode, not waiting for results before requesting subsequent searches. The maximum number of threads the `searchrate` command can use is limited by the number of file descriptors the operating system allows the process to use. The time is measured starts when the request is sent and finishes when the result is received.

`-b` *baseDN* Use the specified base DN for the target entry.

Default is the root DSE, `""`.

Refer to Random Target Syntax and Random Target Substitution for details on number and string substitutions.

`-C` *messages* Display the specified number of results messages before exiting. Results messages appear by default as output on standard out, similar to the following.

```
Avg r=2731.00/thr (1092.40/sec), total=  5462
```

This shows output for two threads searching for five seconds. The average search rate per thread is 2731 searches per thread for the interval measured, for 1092.40 searches per second on average. The total shown for both threads is 5462.

Default is to continue iterating until the command is interrupted.

`-D` *bindDN* Use the specified bind DN to authenticate to the directory.

If the bind DN is not specified, the `searchrate` command attempts anonymous authentication.

`-E` Display the bind DN and filter for searches that failed to retrieve an entry.

`-e` Display the number of attempted searches that failed to retrieve an entry.

`-f` *filter* Use the specified RFC 2254 conformant filter for all searches.

Refer to Random Target Syntax and Random Target Substitution for details on number and string substitutions.

`-h` *hostname* Connect to the directory on the specified host.

Enclose IPv6 addresses in brackets (`[]`) as described in RFC 2732.

Default is to connect to the local host on the loopback address, `127.0.0.1`.

`-i` *filename* Use the file specified to generate target entry base DNs at random.

Refer to Random Target Syntax and Random Target Substitution for details.

`-j` *seconds* Display results each specified number of seconds.

Default is to display results every 5 seconds.

`-K` Keep connections open and only bind once, measuring only the time required to perform the search operation.

Default is to measure the duration the connection is active as the search sequence.

`-k` Keep connections open, measuring only the time required to perform the bind and search operations.

Default is to measure the duration the connection is active as the search sequence.

`-l` *seconds* Set the search time-out at the specified number of seconds for synchronous searches.

Default is 10 seconds.

`-m` *maxIter* Perform no more than the specified number of searches per thread.

Default is for each thread to continue iterating until the command is interrupted.

`-p` *port* Connect to the directory on the specified port.

Default is to connect to the default simple authentication port for LDAP, `389`.

`-q` Run in quiet mode, not displaying results.

Default is to display results every 5 seconds, which you can adjust using the `-j` option.

`-r` *maxRand*
Use the specified maximum to determine the range for random numbers replacing `%d` formatting specifications when searching random target entries.

When you use this option twice, the first occurrence generates random numbers in the range [0,*maxRand1*−1] for the first `%d`, the second [1,*maxRand2*] for the second `%d`.

Refer to Random Target Syntax and Random Target Substitution for details.

`-S` *randSeed*
Use the specified seed, an `unsigned int`, for random number generation.

Default seed is `0`.

`-s` *scope*
Use the specified scope when searching.

The following values are supported for *scope*:

`base`
Examine only the entry specified by the argument to the `-b` option.

`one`
Examine only to the entry specified by the argument to the `-b` option and its immediate children.

`sub`
(Default) Examine the subtree whose root is the entry specified by the argument to the `-b` option.

`-t` *threads*
Use the specified number of the threads to connect to the server.

Default is to use one thread.

`-v`
Display verbose output.

`-W` *filename*
Read the bind password from the specified file.

`-w` *password*
Use the specified bind password to authenticate to the directory.

`-w –`
Prompt for the bind password so it does not appear on the command line or in a file.

Extended Description

The `searchrate` command repeatedly requests search operations of a directory server. Threads may be configured to keep open connections or perform LDAP binds with each operation. The command-line options let you specify the bind credentials.

The command uses LDAP v3, and cannot be used to authenticate to an LDAP v2 directory not supporting LDAP v3. Furthermore, the `searchrate` command uses simple authentication, not secure binding.

By default, the `searchrate` command continues its task indefinitely, displaying results periodically, and displaying any errors encountered as well without interrupting operation.

Random Target Syntax

Include randomly generated numbers by specifying `%d` and `%s` placeholders in the base DN and filters. These placeholders are then replaced according to the following rules:

`%d` Replace this placeholder with random integer values depending on the *maxRand* parameter to the `-r` option.

The `-r` option may be used at most two times to generate random base DNs or filters. Replacement values for the `%d` placeholder range over [0,*maxRand1*–1].

`%s` Replace this placeholder with random strings from the file specified using the `-i` option.

Replacement values for this placeholder are randomly selected lines of the file specified.

Multiple `-r` and `-i` options are matched to the `%d` and `%s` placeholders, respectively, in the order they are used.

Random Target Substitution

The `searchrate` command requires that you apply the following rules for substitutions, displaying an error message when the used incorrectly:

- Use only one type of placeholder, either `%d` or `%s`, per invocation of the `searchrate` command.
- Specify at least as many uses of the `-r` as `%d` placeholders used.
- Use `%%d` and `%%s` to specify literal strings `%d` and `%s`, respectively.

In order to use this random mechanism, you must populate your directory accordingly. For example, you can measure the search rate using the following command:

```
$ searchrate -b "ou=test,dc=example,dc=com" -f "uid=test%d" -r 100
```

In order for the `searchrate` command to find entries, your directory must contain entries corresponding to the following LDIF excerpt:

```
dn: uid=test0,ou=test,dc=example,dc=com
userPassword: auth00

dn: uid=test1,ou=test,dc=example,dc=com
userPassword: auth11

dn: uid=test2,ou=test,dc=example,dc=com
userPassword: auth22
...

dn: uid=test10,ou=test,dc=example,dc=com
userPassword: auth1010
...

dn: uid=test99,ou=test,dc=example,dc=com
userPassword: auth9999
```

Examples

Examples in this section use the following conventions:

- The searchrate command is found in a directory present in the PATH used for the examples.
- The directory server is located on a system named host.
- The directory has been configured to support anonymous access for search and read. Therefore, you do not have to specify bind information.
- The directory server listens on port 389, the default for non-SSL connections.

EXAMPLE 1 searchrate: Sample Output

The following command performs searches until it has displayed five results messages. Notice that each line concerns only the elapsed interval.

```
$ searchrate -h host -b dc=example,dc=com -f "(uid=bjensen)" -C 5
Avg r=1349.00/thr (269.80/sec), total=  1349
Avg r=1312.00/thr (262.40/sec), total=  1312
Avg r=1334.00/thr (266.80/sec), total=  1334
Avg r=1346.00/thr (269.20/sec), total=  1346
Avg r=1340.00/thr (268.00/sec), total=  1340
All threads exited
```

Notice also that a result message provides the following items of information:

- The average search rate per thread of execution
- The average search rate per second
- The total number of search operations performed during the interval the results message concerns

EXAMPLE 2 searchrate: Search Rate Alone

The following command keeping the connection open and binds only once:

```
$ searchrate -h host -b dc=example,dc=com -f "(uid=bjensen)" -C 5 -K
Avg r=2706.00/thr (541.20/sec), total=  2706
Avg r=2706.00/thr (541.20/sec), total=  2706
Avg r=2739.00/thr (547.80/sec), total=  2739
Avg r=2717.00/thr (543.40/sec), total=  2717
Avg r=2731.00/thr (546.20/sec), total=  2731
All threads exited
```

EXAMPLE 3 searchrate: Using a Filter File

The following commands substitute filters from a file to perform searches:

```
$ cat filters
=Jen*
```

```
=Jensen
>=Jensen
<=Jensen
~=Jensen
$ searchrate -h host -b dc=example,dc=com -f "(sn%s)" -i filters -C 5 -K
Avg r=  59.00/thr ( 11.80/sec), total=    59
Avg r=  64.00/thr ( 12.80/sec), total=    64
Avg r=  63.00/thr ( 12.60/sec), total=    63
Avg r=  64.00/thr ( 12.80/sec), total=    64
Avg r=  61.00/thr ( 12.20/sec), total=    61
All threads exited
```

Exit Status

The `searchrate` command returns the following exit status codes.

`0` Successful completion.

non-zero An error occurred.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	Zip distribution only
Stability Level	Evolving

See Also

authrate(1), makeldif(1), modrate(1)

REFERENCE

Administration Commands

Name dpadm – Manage the administration of Directory Proxy Server

Synopsis *install-path*`/dps6/bin/dpadm`
`[`*subcommand*`] [`*global-options*`] [`*subcommand-options*`]`
`[`*subcommand-operands*`]`

Description The `dpadm` command is the administration command for the Directory Proxy Server. Use the `dpadm` command with one of the subcommands described in this man page.

Subcommands The following subcommands are supported:

`dpadm add-cert -i -W` *CERT_PW_FILE INSTANCE_PATH CERT_ALIAS CERT_FILE*
Adds a certificate to the certificate database.

`dpadm add-selfsign-cert [-i] [-W` *CERT_PW_FILE*`] [ -s` *DN* `|`
`--name` *NAME* `[--org` *ORG*`] [--org-unit` *ORG-UNIT*`] [--city` *CITY*`] [--state`
STATE`] [--country` *COUNTRY*`]] [--keyalg` *KEYALG*`] [--sigalg` *SIGALG*`]`
INSTANCE_PATH CERT_ALIAS
Creates a self-signed certificate and adds it to the certificate database.

`dpadm autostart [--off [-i]]` *INSTANCE_PATH*
Enables or disables Directory Proxy Server instance startup at system boot. This command is only available if you installed with Sun Java Enterprise System or native packages, and is not available on Windows.

`dpadm backup` *INSTANCE_PATH ARCHIVE_DIR*
Creates a backup archive of the Directory Proxy Server instance.

`dpadm create [-i] [-p` *PORT*`] [-P` *SECURE_PORT*`] [-u` *USER_NAME* `-g`
GROUP_NAME`] [-D` *DN*`] [-w` *PWD_FILE*`]` *INSTANCE_PATH*
Creates a Directory Proxy Server Instance.

`dpadm delete` *INSTANCE_PATH*
Deletes an instance of Directory Proxy Server.

`dpadm disable-service [-T` *TYPE*`]` *INSTANCE_PATH*
Disables a Directory Proxy Server from being managed as a service. This command is on Windows distributions and Solaris native package distributions only.

`dpadm enable-service [-T` *TYPE*`]` *INSTANCE_PATH* `[`*RESOURCE_GRP*`]`
Enables a Directory Proxy Server instance to be managed as a service. This command is on Windows distributions and Solaris native package distributions only.

`dpadm get-flags` *INSTANCE_PATH* `[`*FLAG*`...]`
Displays the flag values for the Directory Proxy Server instance.

`dpadm import-cert [-i] [-W` *CERT_PW_FILE*`] [-I` *INPUT_PW_FILE*`]`
INSTANCE_PATH CERT_FILE
Imports the public and private keys of a certificate in the certificate database.

`dpadm info` *INSTANCE_PATH*
Displays information about the status and configuration of the Directory Proxy Server instance.

`dpadm list-certs [-i] [-C] [-W` *CERT_PW_FILE*`]` *INSTANCE_PATH*
Lists all certificates in the certificate database.

`dpadm remove-cert [-i] [-W` *CERT_PW_FILE*`]` *INSTANCE_PATH CERT_ALIAS*
Removes a certificate from the certificate database.

`dpadm renew-cert [-i] [-W` *CERT_PW_FILE*`]` *INSTANCE_PATH CERT_ALIAS CERT_FILE*
Renews a certificate in the certificate database.

`dpadm request-cert [-i] [-W` *CERT_PW_FILE*`] [ -s` *DN* `| --name` *NAME* `[--org` *ORG*`] [--org-unit` *ORG-UNIT*`] [--city` *CITY*`] [--state` *STATE*`] [--country` *COUNTRY*`]] [--sigalg` *SIGALG*`] [--keyalg` *KEYALG*`] [-o` *OUTPUT_FILE*`]` *INSTANCE_PATH CERT_ALIAS*
Generates a certificate request.

`dpadm restart [-i] [-W] [`*CERT_PW_FILE*`] [`*INSTANCE_PATH*`]`
Restarts a Directory Proxy Server instance.

`dpadm restore` *INSTANCE_PATH ARCHIVE_DIR*
Restores a Directory Proxy Server instance from a backup archive.

`dpadm set-flags [-i] [-W` *CERT_PW_FILE*`]` *INSTANCE_PATH FLAG=VAL* `[`*FLAG=VAL*`...]`
Sets flag values for a Directory Proxy Server instance.

`dpadm show-cert [-i] [-W` *CERT_PW_FILE*`] [-o` *OUTPUT_FILE*`] [-F` *FORMAT*`]` *INSTANCE_PATH* `[`*CERT_ALIAS*`]`
Displays a certificate.

If no *CERT_ALIAS* is specified, the default server certificate is displayed.

`dpadm split-ldif` *INSTANCE_PATH LDIF_FILEOUTPUT_FILE_DIR*
Splits the LDIF file given by *LDIF_FILE* into multiple LDIF files according to the data distribution configured in Directory Proxy Server. One LDIF file is created for each data view defined in the *LDIF_FILE* file.

The LDIF files are stored in the *OUTPUT_FILE_DIR* directory and are automatically named after the data view, with the following format: *OUTPUT_FILE_DIR.DATA_VIEW_NAME*`.ldif`

The `dpadm` `split-ldif` command can be launched even if the Directory Proxy Server is running.

`dpadm start [-Ei] [-W` *CERT_PW_FILE*`]` *INSTANCE_PATH*
Starts a Directory Proxy Server instance.

`dpadm stop` *INSTANCE_PATH*
Stops a Directory Proxy Server instance.

Global Options

The following options are global, and are applicable to all commands and subcommands.

`--?`
`--help` Displays instructions for accessing help.

`-V`
`--version` Displays the current version of `dpadm`. The version is provided in the format *year.day.time*. So version number `2006.178.0035` was built on the 178th day of 2006 at 00h35. If the components used by `dpadm` are not aligned, the version of each individual component is displayed.

`-v`
`--verbose` Displays instructions for accessing verbose help.

Subcommand Options

The following options are applicable to the subcommands where they are specified.

-C
`--ca` Lists Certificate Authority certificates only. The default is to list server certificates only.

`--city` *CITY* Adds `L=CITY` to the subject DN. Default is none.

`--country` *COUNTRY* Adds `C=COUNTRY` to the subject DN. The default is none.

-D *DN*
`--rootDN` *DN* Defines the Proxy Manager DN. The default is `cn=Proxy Manager`.

`-E`
`--safe` Starts Directory Proxy Server with the configuration used at the last successful startup.

-F *FORMAT*
`--format` *FORMAT* Specifies the output format. The options are `readable` and `ascii`. The default is `readable`.

-g *GROUP_NAME*
`--group` *GROUP_NAME* Specifies the group name for the owner of the server instance. The default is the name of the current group.

-i
`--no-inter` Does not prompt for confirmation before performing the operation.

-I *INPUT_PW_FILE*
`--input-pwd-file`*INPUT_PW_FILE*

Specifies the certificate password. The default is to prompt for a password.

`--keyalg` *KEYALG*

Specifies the key-pair generation algorithm (DSA or RSA).

`--sigalg` *SIGALG*

Specifies the signature algorithm used to sign the certificate. The signature algorithm depends on the underlying key-pair generation algorithm. The default signature algorithm is `SHA1withDSA` when the key algorithm is DSA, and `MD5withRSA` when the key algorithm is RSA.

`--name` *NAME*

Adds `CN=NAME` to the subject DN. The default is the hostname.

-O *OUTPUT_PW_FILE*
`--output-pwd-file` *OUTPUT_PW_FILE*

Reads the output password from the `OUTPUT_FILE` file. The default is a prompt for a password.

`--o` *OUTPUT_FILE*
`--output` *OUTPUT_FILE*

Stores the command results in the `OUTPUT_FILE` file. The default is `stdout`.

`--off`

Disables the autostart of an instance of Directory Proxy Server at system boot

`--org` *ORG*

Adds `O=ORG` to the subject DN. The default is none.

`--org-unit` *ORG-UNIT*

Adds `O=ORG-UNIT` to the subject DN. The default is none.

`--p` *PORT*
`--port` *PORT*

Specifies the port for LDAP traffic. The default is `389` or `1389`.

`--P` *SECURE_PORT*
`--secure-port` *SECURE_PORT*

Specifies the secure SSL port for LDAP traffic. The default is `636` or `1636`.

`--S` *DN*
`--subjectDN` *DN*

Specifies the subject DN. The default is `cn=`*CERT_ALIAS* `cn=`*hostname*.

`--state` *STATE*
: Adds `ST=STATE` to the subject DN. Default is the hostname.

`--T` *TYPE*
`--type` *TYPE*
: Service type. Can be `SMF` when using Solaris 10, or `WIN_SERVICE` when using Windows.

-u *USER_NAME*
`--username` *USER_NAME*
: Specifies the user name for the owner of the server instance. The default is the name of the current user.

`--W` *CERT_PW_FILE*
`--cert-pwd-file` *CERT_PW_FILE*
: Reads the certificate database password from the `CERT_PW_FILE` file. The default is a prompt for password.

`--w` *PW_FILE*
`--pwd-file` *PW_FILE*
: Reads the password from the `PW_FILE` file. The default is a prompt for password.

Subcommand Operands

The following operands are supported:

ARCHIVE_DIR
: Specifies the path to the backup of the Directory Proxy Server instance.

CERT_ALIAS
: Specifies the certificate alias.

CERT_FILE
: Specifies the file that contains the certificate.

FLAG
: Specifies a flag that represents a property operand when using the command `dpadm get-flags`. Possible flags: `cert-pwd-prompt`, `jvm-args`.

FLAG=VALUE
: Specifies a flag and its value. The *FLAG=VALUE* operand can have the following values:

`cert-pwd-prompt=off`
: Sets the certificate database password storage mode to `on`. The certificate database password is stored on the file system. This is the default value.

`cert-pwd-prompt=on`
: Sets the certificate database password storage mode to `off`. The certificate database password is *not* stored on the file system. You are prompted to supply the certificate database password when needed.

`jvm-args="arg1 arg2 ..."`

These values are arguments passed to the Java Virtual Machine (JVM).

The default value is `jvm-args=-Xmx250M -Xms250M`.

`-Xmx`*memory* is the maximum memory size for the JVM. The default value is `-Xmx250M` (250 MB).

`-Xms`*memory* is the startup memory size for the JVM. The default value is `-Xms250M` (250 MB). The startup memory size `-Xms`*memory* should be the same as the maximum memory size `-Xmx`*memory*.

`-XX:NewRatio=`*ratio* is applicable to the Sun Hotspot JVM only, and is the ratio between old and young generation memory. The recommended value is `-XX:NewRatio=1`, which is equal old and young generation memory.

The `-d` flag specifies which JVM is used (32-bits or 64-bits). By default, Directory Proxy Server is launched with a 64-bit JVM, if available, and with a 32-bit JVM otherwise. If you want to override this behavior and specify the JVM, set the `jvm-args` flag to either d-32 or d-64, for example `jvm-args=-Xmx250M -Xms250M -d32`

You can use the `jvm-args` flag to pass a list of arguments to the JVM. For information about JVM arguments not described in this man page, see the java(1) man page.

INSTANCE_PATH Specifies the path to the Directory Proxy Server instance.

LDIF_FILE Specifies the LDIF file that is to be split by using the `split_ldif` subcommand.

OUTPUT_FILE_DIR Specifies the directory where LDIF files are placed after being split by the `split_ldif` subcommand.

Exit Status

The following exit status values are returned:

`0`	Successful completion.
non-zero	An error occurred.

Examples

The following examples show how the `dpadm` command is used.

EXAMPLE 1 Creating a Directory Proxy Server Instance

The following example shows how to create a Directory Proxy Server instance.

```
$ dpadm create /local/dps
```

EXAMPLE 2 Starting a Directory Proxy Server Instance

The following example shows how to start a Directory Proxy Server instance.

```
$ dpadm start /local/dps
```

EXAMPLE 3 Getting Information about a Directory Proxy Server Instance

The following example shows how to get information about a Directory Proxy Server instance.

```
$ dpadm info /local/dps
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

dpconf(1M)

Name dpconf – Manage the configuration of Directory Proxy Server

Synopsis *install-path*`/dps6/bin/dpconf`
subcommand [*global-options*] [*subcommand-options*]
[*subcommand-operands*]

Description The `dpconf` command manages the configuration of Directory Proxy Server. An instance of Directory Proxy Server must be running in order for you to run the `dpconf` command.

Subcommands The following subcommands are supported:

`dpconf add-jdbc-attr [-h` *host*`] [-p` *port*`]` *TABLE_NAME ATTR_NAME COLUMN_NAME*
Add a JDBC attribute by using a SQL table.

`dpconf add-virtual-transformation [-h` *host*`] [-p` *port*`]` *VIEW_NAME MODEL ACTION ATTR_NAME* [*PARAM...*]
Add a virtual transformation to a data view.

`dpconf attach-jdbc-data-source [-h` *host*`] [-p` *port*`]` *POOL_NAME SRC_NAME* [*SRC_NAME...*]
Attach one or more JDBC data sources to a JDBC data source pool.

`dpconf attach-ldap-data-source [-h` *host*`] [-p` *port*`]` *POOL_NAME SRC_NAME* [*SRC_NAME...*]
Attach one or more LDAP data sources to an LDAP data source pool.

`dpconf create-connection-handler [-h` *host*`] [-p` *port*`]` *NAME* [*NAME...*]
Create one or more new connection handlers.

`dpconf create-custom-search-size-limit [-h` *host*`] [-p` *port*`]` *POLICY_NAME LIMIT_NAME* [*LIMIT_NAME...*]
Create one or more new custom search size limits for a resource limits policy.

`dpconf create-jdbc-data-source [-h` *host*`] [-p` *port*`] -b` *DB_NAME* `-B` *DB_URL* `-J` *DRIVER_URL* `[-J` *DRIVER_URL*`]... -S` *DRIVER_CLASS SRC_NAME*
Create a JDBC data source that corresponds to an existing JDBC database.

`dpconf create-jdbc-data-source-pool [-h` *host*`] [-p` *port*`]` *NAME* [*NAME...*]
Create one or more JDBC data source pools.

`dpconf create-jdbc-data-view [-h` *host*`] [-p` *port*`]` *JDBC_VIEW_NAME POOL_NAME SUFFIX_DN*
Create a data view that enables LDAP applications to view JDBC tables.

`dpconf create-jdbc-object-class [-h` *host*`] [-p` *port*`]` *JDBC_VIEW_NAME OBJECTCLASS PRIMARY_TABLE* [*SECONDARY_TABLE...*] *DN_PATTERN*
Create a JDBC object class and attach it to a JDBC data view. At least one JDBC table, the primary table, must be specified. Additional tables can be specified if the JDBC data view is to be a join data view of more than one JDBC table.

`dpconf create-jdbc-table [-h` *host*`] [-p` *port*`]` *TABLE_NAME DB_TABLE*
Create a JDBC table.

`dpconf create-join-data-view [-h` *host*`] [-p` *port*`]` *JOIN_NAME PRIMARY_NAME SECONDARY_NAME SUFFIX_DN*
Create a virtual data view that combines or aggregates two separate data views. One of these data views is the primary data view, and the other the secondary data view. Before you can create a join data view, you must define at least one join rule on the secondary data view. To define join rules, set the `dn-join-rule` or `filter-join-rule` properties of the secondary data view.

`dpconf create-ldap-data-source [-h` *host*`] [-p` *port*`]` *NAME HOST*`:`*PORT*
Create a new LDAP data source.

`dpconf create-ldap-data-source-pool [-h` *host*`] [-p` *port*`]` *NAME* `[`*NAME...*`]`
Create one or more new LDAP data source pools.

`dpconf create-ldap-data-view [-h` *host*`] [-p` *port*`]` *VIEW_NAME POOL_NAME SUFFIX_DN*
Create a new LDAP data view.

`dpconf create-ldif-data-view [-h` *host*`] [-p` *port*`]` *VIEW_NAME LDIF_FILE_NAME SUFFIX_DN*
Create a new LDIF data view.

`dpconf create-request-filtering-policy [-h` *host*`] [-p` *port*`]` *NAME* `[`*NAME...*`]`
Create one or more new request filtering policies.

`dpconf create-resource-limits-policy [-h` *host*`] [-p` *port*`]` *NAME* `[`*NAME...*`]`
Create one or more new resource limits policies.

`dpconf create-search-data-hiding-rule [-h` *host*`] [-p` *port*`]` *POLICY_NAME RULE_NAME* `[`*RULE_NAME...*`]`
Create one or more new search data hiding rules for a request filtering policy.

`dpconf create-user-mapping [-h` *host*`] [-p` *port*`]` *NAME USER_DN USER_PWD_FILE*
Create a new user mapping.

`dpconf delete-connection-handler [-h` *host*`] [-p` *port*`]` *NAME* `[`*NAME...*`]`
Delete existing connection handlers.

`dpconf delete-custom-search-size-limit [-h` *host*`] [-p` *port*`]` *POLICY_NAME LIMIT_NAME* `[`*LIMIT_NAME...*`]`
Delete existing custom search size limit for a resource limits policy.

`dpconf delete-jdbc-data-source [-h` *host*`] [-p` *port*`]` *NAME* `[`*NAME...*`]`
Delete one or more JDBC data sources.

`dpconf delete-jdbc-data-source-pool [-h` *host*`] [-p` *port*`]` *NAME* `[`*NAME...*`]`
Delete one or more JDBC data source pools.

`dpconf delete-jdbc-data-view [-h` *host*`] [-p` *port*`]` *NAME* `[`*NAME...*`]`
Delete one or more JDBC data views.

`dpconf delete-jdbc-object-class [-h` *host*`] [-p` *port*`]` *JDBC_VIEW_NAME* *OBJECTCLASS* `[`*OBJECTCLASS...*`]`
Delete one or more JDBC object classes.

`dpconf delete-jdbc-table [-h` *host*`] [-p` *port*`]` *NAME* `[`*NAME...*`]`
Delete one or more JDBC tables.

`dpconf delete-join-data-view [-h` *host*`] [-p` *port*`]` *JOIN_NAME*
Delete a join data view.

`dpconf delete-ldap-data-source [-h` *host*`] [-p` *port*`]` *NAME* `[`*NAME...*`]`
Delete existing LDAP data sources.

`dpconf delete-ldap-data-source-pool [-h` *host*`] [-p` *port*`]` *NAME* `[`*NAME...*`]`
Delete existing LDAP data source pools.

`dpconf delete-ldap-data-view [-h` *host*`] [-p` *port*`]` *VIEW_NAME* `[`*VIEW_NAME...*`]`
Delete existing LDAP data views.

`dpconf delete-ldif-data-view [-h` *host*`] [-p` *port*`]` *VIEW_NAME* `[`*VIEW_NAME...*`]`
Delete existing LDIF data views.

`dpconf delete-request-filtering-policy [-h` *host*`] [-p` *port*`]` *NAME* `[`*NAME...*`]`
Delete existing request filtering policies.

`dpconf delete-resource-limits-policy [-h` *host*`] [-p` *port*`]` *NAME* `[`*NAME...*`]`
Delete existing resource limits policies.

`dpconf delete-search-data-hiding-rule [-h` *host*`] [-p` *port*`]` *POLICY_NAME* *RULE_NAME* `[`*RULE_NAME...*`]`
Delete an existing search data hiding rule.

`dpconf delete-user-mapping [-h` *host*`] [-p` *port*`]` *NAME* `[`*NAME...*`]`
Delete existing user mappings.

`dpconf detach-jdbc-data-source [-h` *host*`] [-p` *port*`]` *POOL_NAME* *SRC_NAME* `[`*SRC_NAME...*`]`
Detach JDBC data sources from a JDBC data source pool.

`dpconf detach-ldap-data-source [-h` *host*`] [-p` *port*`]` *POOL_NAME* *SRC_NAME* `[`*SRC_NAME...*`]`
Detach LDAP data sources from an LDAP data source pool.

`dpconf get-access-log-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`] [`*PROP...*`]`
View the properties of the access log.

`dpconf get-attached-ldap-data-source-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *POOL_NAME* *SRC_NAME* `[`*PROP...*`]`
View the properties of an attached LDAP data source.

`dpconf get-connection-handler-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *NAME* `[`*PROP...*`]`
View the properties of a connection handler.

`dpconf get-custom-search-size-limit-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *POLICY_NAME* *LIMIT_NAME* `[`*PROP...*`]`
View the properties of custom search size limits for a resource limits policy.

`dpconf get-error-log-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *NAME* `[`*PROP...*`]`
View the properties of the error log.

`dpconf get-jdbc-attr-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *TABLE_NAME* *ATTR_NAME* `[`*PROP...*`]`
View the properties of a JDBC attribute.

`dpconf get-jdbc-data-source-pool-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *NAME* `[`*PROP...*`]`
View the properties of a JDBC data source pool.

`dpconf get-jdbc-data-source-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *NAME* `[`*PROP...*`]`
View the properties of a JDBC data source.

`dpconf get-jdbc-data-view-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *NAME* `[`*PROP...*`]`
View the properties of a JDBC data view.

`dpconf get-jdbc-object-class-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *NAME* `[`*PROP...*`]`
View the properties of a JDBC object class.

`dpconf get-jdbc-table-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *TABLE_NAME* `[`*PROP*`]`
View the properties of a JDBC table.

`dpconf get-join-data-view-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *JOIN_NAME* `[`*PROP...*`]`
View the properties of a join data view.

`dpconf get-ldap-data-source-pool-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *NAME* `[`*PROP...*`]`
View the properties of an LDAP data source pool.

`dpconf get-ldap-data-source-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *NAME* `[`*PROP...*`]`
View the properties of an LDAP data source.

`dpconf get-ldap-data-view-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *VIEW_NAME* `[`*PROP...*`]`
View the properties of an LDAP data view.

`dpconf get-ldap-listener-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *NAME* `[`*PROP...*`]`

View the properties of the LDAP listener.

`dpconf get-ldaps-listener-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`] [`*PROP...*`]`

View the properties of the LDAPS listener.

`dpconf get-ldif-data-view-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *VIEW_NAME* `[`*PROP...*`]`

View the properties of an LDIF data view.

`dpconf get-request-filtering-policy-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *NAME* `[`*PROP...*`]`

View the properties of a request filtering policy.

`dpconf get-resource-limits-policy-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *NAME* `[`*PROP...*`]`

View the properties of the resource limits policy

`dpconf get-search-data-hiding-rule-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *POLICY_NAME RULE_NAME* `[`*PROP...*`]`

View the properties of search data hiding rules for a request filtering policy.

`dpconf get-server-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`] [`*PROP...*`]`

View the properties of a Directory Proxy Server.

`dpconf get-user-mapping-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *NAME* `[`*PROP...*`]`

View the properties of a user mapping.

`dpconf get-virtual-aci-prop [-h` *host*`] [-p` *port*`] [`*PROP...*`]`

View the properties of the data view defined to provide access to virtual ACIs.

`dpconf get-virtual-transformation-prop [-h` *host*`] [-p` *port*`] [-M` *UNIT*`] [-Z` *UNIT*`]` *VIEW_NAME TRANSFORMATION_NAME* `[`*PROP...*`]`

View the properties of a virtual transformation. Virtual transformation properties that can be specified include `action`, `attr-name`, `model`, `internal-value` and `view-value`.

`dpconf help-properties [-r]`

View information about the properties exposed by subcommands.

`dpconf info`

Display information about server configuration.

`dpconf list-attached-jdbc-data-sources [-h` *host*`] [-p` *port*`] [-E] [`*POOL_NAME*`...]`

List JDBC data sources that are attached to a data source pool.

`dpconf list-attached-ldap-data-sources [-h` *host*`] [-p` *port*`] [-E] [`*POOL_NAME*`...]`
List LDAP data sources that are attached to a data source pool.

`dpconf list-connection-handlers [-h` *host*`] [-p` *port*`] [-E]`
List the existing connection handlers.

`dpconf list-custom-search-size-limits [-h` *host*`] [-p` *port*`] [-E] [`*POLICY_NAME*`...]`
List the existing custom search size limits for a resource limits policy.

`dpconf list-jdbc-attrs [-h` *host*`] [-p` *port*`] [-E] [`*TABLE_NAME*`...]`
List the JDBC attributes that have been defined using SQL tables.

`dpconf list-jdbc-data-source-pools [-h` *host*`] [-p` *port*`] [-E]`
List the existing JDBC data source pools.

`dpconf list-jdbc-data-sources [-h` *host*`] [-p` *port*`] [-E]`
List the existing JDBC data sources.

`dpconf list-jdbc-object-classes [-h` *host*`] [-p` *port*`] [-E] [`*JDBC_VIEW_NAME*`...]`
List the JDBC object classes that are attached to a JDBC data view.

`dpconf list-jdbc-tables [-h` *host*`] [-p` *port*`] [-E]`
List all JDBC tables.

`dpconf list-join-data-views [-h` *host*`] [-p` *port*`] [-E]`
List the existing join data views.

`dpconf list-ldap-data-source-pools [-h` *host*`] [-p` *port*`] [-E]`
List the existing LDAP data source pools.

`dpconf list-ldap-data-sources [-h` *host*`] [-p` *port*`] [-E]`
List the existing LDAP data sources.

`dpconf list-ldap-data-views [-h` *host*`] [-p` *port*`] [-E]`
List the existing LDAP data views.

`dpconf list-ldif-data-views [-h` *host*`] [-p` *port*`] [-E]`
List the existing LDIF data views.

`dpconf list-request-filtering-policies [-h` *host*`] [-p` *port*`] [-E]`
List the existing request filtering policies.

`dpconf list-resource-limits-policies [-h` *host*`] [-p` *port*`] [-E]`
List the existing resource limits policies.

`dpconf list-search-data-hiding-rules [-h` *host*`] [-p` *port*`] [-E] [`*POLICY_NAME*`...]`
List the existing search data hiding rules for a request filtering policy.

`dpconf list-user-mappings [-h` *host*`] [-p` *port*`] [-E]`
List the existing user mappings.

dpconf list-virtual-transformations [-h *host*] [-p *port*] [-E] [*VIEW_NAME*...]
List the virtual transformations that are defined on a data view.

dpconf remove-jdbc-attr [-h *host*] [-p *port*] *TABLE_NAME* *ATTR_NAME* [*ATTR_NAME*...]
Delete a JDBC attribute.

dpconf remove-virtual-transformation [-h *host*] [-p *port*] *VIEW_NAME* *TRANSFORMATION_NAME* [*TRANSFORMATION_NAME*...]
Delete a virtual transformation.

dpconf rotate-log-now [-h *host*] [-p *port*] *LOG_TYPE*
Launch the rotation of a log file.

dpconf set-access-log-prop [-h *host*] [-p *port*] *PROP*:*VAL* [*PROP*:*VAL*...]
Change the properties of the access log. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

dpconf set-attached-ldap-data-source-prop [-h *host*] [-p *port*] *POOL_NAME* *SRC_NAME* *PROP*:*VAL* [*PROP*:*VAL*...]
Change the properties of an attached LDAP data source. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

dpconf set-connection-handler-prop [-h *host*] [-p *port*] *NAME* *PROP*:*VAL* [*PROP*:*VAL*...]
Change the properties of a connection handler. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

dpconf set-custom-search-size-limit-prop [-h *host*] [-p *port*] *POLICY_NAME* *LIMIT_NAME* *PROP*:*VAL* [*PROP*:*VAL*...]
Change the properties of custom search size limits for a resource limits policy. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

dpconf set-error-log-prop [-h *host*] [-p *port*] *PROP*:*VAL* [*PROP*:*VAL*...]
Change the properties of the error log. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dpconf set-jdbc-attr-prop [-h` *host*`] [-p` *port*`]` *TABLE_NAME ATTR_NAME PROP*:*VAL* [*PROP*:*VAL*...]

Change the properties of a JDBC attribute. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dpconf set-jdbc-data-source-pool-prop [-h` *host*`] [-p` *port*`]` *NAME PROP*:*VAL* [*PROP*:*VAL*...]

Change the properties of a JDBC data source pool. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dpconf set-jdbc-data-source-prop [-h` *host*`] [-p` *port*`]` *NAME PROP*:*VAL* [*PROP*:*VAL*...]

Change the properties of a JDBC data source. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dpconf set-jdbc-data-view-prop [-h` *host*`] [-p` *port*`]` *VIEW_NAME PROP*:*VAL* [*PROP*:*VAL*...]

Change the properties of a JDBC data view. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dpconf set-jdbc-object-class-prop [-h` *host*`] [-p` *port*`]` *JDBC_VIEW_NAME OBJECTCLASS PROP*:*VAL* [*PROP*:*VAL*...]

Change the properties of a JDBC object class. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dpconf set-jdbc-table-prop [-h` *host*`] [-p` *port*`]` *TABLE_NAME PROP*:*VAL* [*PROP*:*VAL*...]

Change the properties of a JDBC table. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dpconf set-join-data-view-prop [-h` *host*`] [-p` *port*`]` *VIEW_NAME PROP*`:`*VAL* `[`*PROP*`:`*VAL*`...]`

Change the properties of a join data view. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dpconf set-ldap-data-source-pool-prop [-h` *host*`] [-p` *port*`]` *NAME PROP*`:`*VAL* `[`*PROP*`:`*VAL*`...]`

Change the properties of an LDAP data source pool. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dpconf set-ldap-data-source-prop [-h` *host*`] [-p` *port*`]` *NAME PROP*`:`*VAL* `[`*PROP*`:`*VAL*`...]`

Change the properties of an LDAP data source. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`set-ldap-data-view-prop [-h` *host*`] [-p` *port*`]` *VIEW_NAME PROP*`:`*VAL* `[`*PROP*`:`*VAL*`...]`

Change the properties of an LDAP data view. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dpconf set-ldap-listener-prop [-h` *host*`] [-p` *port*`]` *PROP*`:`*VAL* `[`*PROP*`:`*VAL*`...]`

Change the properties of the LDAP listener. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dpconf set-ldaps-listener-prop [-h` *host*`] [-p` *port*`]` *PROP*`:`*VAL* `[`*PROP*`:`*VAL*`...]`

Change the properties of the LDAPS listener. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dpconf set-ldif-data-view-prop [-h` *host*`] [-p` *port*`]` *VIEW_NAME PROP*`:`*VAL* `[`*PROP*`:`*VAL*`...]`

Change the properties of an LDIF data view. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dpconf set-request-filtering-policy-prop [-h` *host*`] [-p` *port*`]` *NAME* *PROP*:*VAL* `[`*PROP*:*VAL*`...]`

Change the properties of a request filtering policy. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dpconf set-resource-limits-policy-prop [-h` *host*`] [-p` *port*`]` *NAME* *PROP*:*VAL* `[`*PROP*:*VAL*`...]`

Change the properties of a resource limits policy. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dpconf set-search-data-hiding-rule-prop [-h` *host*`] [-p` *port*`]` *POLICY_NAME* *RULE_NAME* *PROP*:*VAL* `[`*PROP*:*VAL*`...]`

Change the properties of search data hiding rules for a request filtering policy. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dpconf set-server-prop [-h` *host*`] [-p` *port*`]` *PROP*:*VAL* `[`*PROP*:*VAL*`...]`

Change the properties of a Directory Proxy Server instance. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dpconf set-user-mapping-prop [-h` *host*`] [-p` *port*`]` *NAME* *PROP*:*VAL* `[`*PROP*:*VAL*`...]`

Change the properties of a user mapping. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dpconf set-virtual-aci-prop [-h` *host*`] [-p` *port*`]` *PROP*:*VAL* `[`*PROP*:*VAL*`...]`

Change the properties of the data view defined to provide access to virtual ACIs. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dpconf set-virtual-transformation-prop [-h` *host*`] [-p` *port*`]` *VIEW_NAME TRANSFORMATION_NAME PROP*`:`*VAL* `[`*PROP*`:`*VAL*`...]`

Change the properties of a virtual transformation that was defined on the data view. If you do not specify a *VAL*, the value of the property is reset.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

Global Options

The following options are global to all commands and subcommands:

`-?`
`--help`
Displays help information for a command or subcommand.

`-c`
`--accept-cert`
Does not ask for confirmation before accepting untrusted server certificates.

`-D` *USER_DN*
`--user-dn` *USER_DN*
Binds as *USER_DN*. The `dpconf` command searches for a *USER_DN* value in the following order:

- A *USER_DN* specified in the command line
- A *USER_DN* set by using the `$LDAP_ADMIN_USER` environment variable

If none of these are found, the default is to bind as the `cn=Proxy Manager` user.

`-e`
`--unsecured`
Connects over LDAP with no secure connection. To connect over a clear connection by default, set the `DIR_PROXY_UNSECURED` environment variable.

`-h` *HOST*
`--hostname` *HOST*
Connects to the proxy server on *HOST*. The `dpconf` command searches for a *HOST* value in the following order:

- A *HOST* specified in the command line
- A *HOST* set by using the `$DIR_PROXY_HOST` environment variable

If none of these are found, the default is to use the local host.

`-i`
`--no-inter`
Does not ask for confirmation or passwords.

`-j`
`--reject-cert`
Does not ask for confirmation before rejecting untrusted server certificates in this session.

`-p` *PORT*
`--port` *PORT*

Connects to the proxy on *PORT*. The `dpconf` command searches for a *PORT* value in the following order:

- A *PORT* specified in the command line
- A *PORT* set by using the `$DIR_PROXY_PORT` environment variable

If none of these are found, the default is to use port `389`.

This option is mutually exclusive with `-P,--secure-port`.

`-P` *PORT*
`--secure-port` *PORT*

Connects over SSL to the proxy on *PORT*. The `dpconf` command searches for a *PORT* value in the following order:

- A *PORT* specified in the command line
- A *PORT* set by using the `$DIR_PROXY_PORT` environment variable

If none of these are found, the default is to use port `1636`.

This option is mutually exclusive with `-p,--port`.

`-r`
`--attr-map`

Displays help properties and their corresponding attributes in `cn=config`.

`-v`
`--verbose`

Displays extra information. This option is especially useful in the `list` subcommands. For an example of the use of the verbose option, see Example 5.

`-V--version`

Displays the current version of `dpconf`. The version is provided in the format *year.day.time*. So version number `2006.178.0035` was built on the 178th day of 2006 at 00h35. If the components used by `dpconf` are not aligned, the version of each individual component is displayed.

`-w` *FILE*
`--pwd-file` *FILE*

Specifies that the LDAP password is read from *FILE*. The `dpconf` command searches for a password *FILE* value in the following order:

- A password or password file specified in the command line
- A password file set by using the `$LDAP_ADMIN_PWF` environment variable

If none of these are found, the default is to prompt for the password.

Subcommand Options

The following options can be used with the subcommands:

`-b`
`--db-name`
The name of the JDBC database for which you create a JDBC data source.

`-B`
`--db-url`
The URL to the JDBC database for which you create a JDBC data source.

`-E`
`--record`
Modifies the display output to show one property value per line.

`-J`
`--driver-url`
The URL to the JDBC driver.

`-M` *UNIT*
`--unit-time` *UNIT*
Display time data with `UNIT` unit. The value for *UNIT* can be `M`, `w`, `d`, `h`, `m`, `s`, or `ms` (month, week, day, hour, minute, second, or milisecond).

`-S`
`--driver-class`
The class of the JDBC driver.

`-Z` *UNIT*
`--unit-size` *UNIT*
Display memory size data with UNIT unit. The value for *UNIT* can be `T`, `G`, `M`, `k`, or `b` (Terabyte, Gigabyte, Megabyte, kilobyte, or byte).

Subcommand Operands

The following operands can be used with the subcommands:

ACTION
Describes what a transformation does to its target entry or entries. The following transformation actions are possible:

- `add-attr` Add a new attribute. The value of the new attribute is defined by the `PARAM` operand.
- `add-attr-value` Add a calculated value to an existing attribute. The value that must be added is defined by the `PARAM` operand.
- `attr-value-mapping` Map one attribute to another attribute to provide the attribute value. The value is defined by the `internal-value` and `view-value` `PARAM` operands.
- `def-value` Add a default value to an existing attribute. The value that must be added is defined by the `PARAM` operand.

- `remove-attr` Remove an attribute.
- `remove-attr-value` Remove a value from an existing attribute. This action is usually used in the case of multi-value attributes when one of the values should be removed.

ATTR_NAME

The name of a virtual attribute or JDBC attribute to be added or removed.

COLUMN_NAME

The name of a column in an SQL table.

DB_TABLE

The name of an SQL table.

DN_PATTERN

The pattern that should be used to construct a DN from a JDBC table.

HOST

Contacts the LDAP server on the specified host, which may be a host name or an IP address.

For example, when mapping the IPv4 address `192.168.0.99` to IPv6, pass the `-h` option with its argument as `-h ::ffff:192.168.0.99`.

JDBC_VIEW_NAME

The name of a JDBC data view.

JOIN_NAME

The name of a join data view.

LDIF_FILE_NAME

The name of a file on the Directory Proxy Server that contains the LDIF data.

LIMIT_NAME

The name of a custom search size limit.

LOG_TYPE

The type of log, log type can be `access` or `error`.

MODEL

The direction in which a transformation action will be applied. The transformation model can be one of `mapping`, `read`, or `write`.

A mapping transformation is applied during the request, and its inverse is applied during the response. A write transformation is applied during the request, but not during the response. A write transformation changes the physical data in storage. A read transformation is applied only during the response to a request.

NAME

The name of an object to be created or deleted, or the name of an object for which you are getting or setting properties.

OBJECTCLASS

The name of a JDBC object class.

PARAM

The parameters to be applied to a virtual transformation. Depending on the transformation, *PARAM* can be one or more of the following:

- *value* specifies the value of the virtual attribute for all transformation actions other than `attrValueMapping`.
- internal-value:*value* used only with the `attrValueMapping` transformation action. Specifies the value of the virtual attribute that should be written to the physical data source.
- view-value:*value* used only with the `attrValueMapping` transformation action. Specifies the value of the virtual attribute that should be returned to the client.

POLICY_NAME

The name of the resource limits policy or request filtering policy to which limits or rules are to be applied.

POOL_NAME

The name of an existing LDAP or JDBC data source pool.

PORT

The port number of the object to be created.

PRIMARY_NAME

The name of the primary data view that is the source for a join data view.

PRIMARY_TABLE

The name of the primary table in a JDBC database.

PROP

The name of the property. For a list of property names and values, use this command:

```
dpconf help-properties.
```

The rws and rwd keywords of a property indicate whether changes to the property require the server to be restarted. If a property has an `rws` (read, write, static) keyword, the server must be restarted when the property is changed. If a property has an `rwd` (read, write, dynamic) keyword, modifications to the property are implemented dynamically (without restarting the server).

For multi-valued properties, use the syntax *PROP+*:*VAL* to add a value, and *PROP-*:*VAL* to remove a value.

Multi-valued properties are identified by the `M` keyword. For a list of multi-valued properties, use this command:

```
dpconf help-properties | grep " M "
```

RULE_NAME

The name of a search data hiding rule.

SECONDARY_NAME
The name of the secondary data view that is the source for a join data view.

SECONDARY_TABLE
The name of the secondary table in a JDBC database.

SRC_NAME
The name of an LDAP or JDBC data source.

SUFFIX_DN
The DN of the suffix represented by the data view.

TABLE_NAME
The name of a JDBC table.

TRANSFORMATION_NAME
The name of a virtual transformation.

USER_DN
The DN of the user to be mapped.

USER_PWD_FILE
The name of the password file, or the value - meaning to prompt for the password.

VAL
The new value of the property. For a complete list of property names and values, use the command `dpconf help-properties -v`.

When the `VAL` operand is used for passwords, it can have the following values:

- The name of the password file.
- The value -, meaning to prompt for the password.

VIEW_NAME
The name of a data view.

Description

Syntax values shown in lower case or partly in lower case are literal values.

Those shown in upper case are syntax types, defined as follows:

`ATTR_NAME`
A valid attribute type name such as `cn` or `objectClass`.

`BOOLEAN`
`true` or `false`.

`DN`
A valid distinguished name such as `ou=People,dc=example,dc=com`.

`DURATION`
A duration specified in months (`M`), weeks (`w`), days (`d`), hours (`h`), minutes (`m`), seconds (`s`), and miliseconds (`ms`), or some combination with multiple specifiers. For example, you can specify one week as `1w`, `7d`, `168h`, `10080m`, or `604800s`. You can also specify one week as `1w0d0h0m0s`.

`DURATION` properties typically do not each support all duration specifiers (`Mwdhms`). Examine the output of `dsconf help-properties` for the property to determine which duration specifiers are supported.

`EMAIL_ADDRESS`
A valid e-mail address.

`HOST_NAME`
An IP address or host name.

`INTEGER`
A positive integer value between 0 and the maximum supported integer value in the system address space. On 32-bit systems, `2147483647`. On 64-bit systems, `9223372036854775807`.

`INTERVAL`
An interval value of the form *hhmm*-*hhmm 0123456*, where the first element specifies the starting hour, the next element the finishing hour in 24-hour time format, from `0000-2359`, and the second specifies days, starting with Sunday (`0`) to Saturday (`6`).

`IP_RANGE`
An IP address or range of address in one of the following formats:

- IP address in dotted decimal form.
- IP address and bits, in the form of network number/mask bits.
- IP address and quad, in the form of a pair of dotted decimal quads.
- All address. A catch-all for clients that are note placed into other, higher priority groups.
- `0.0.0.0`. This address is for groups to which initial membership is not considered. For example, for groups that clients switch to after their initial bind.
- IP address of the local host.

`LDAP_URL`
A valid LDAP URL as specified by RFC 2255 (`http://www.ietf.org/rfc/rfc2255.txt`).

`MEMORY_SIZE`
A memory size specified in gigabytes (`G`), megabytes (`M`),kilobytes (`k`), or bytes (`b`). Unlike `DURATION` properties, `MEMORY_SIZE` properties cannot combine multiple specifiers. However, `MEMORY_SIZE` properties allow decimal values, for example, 1.5M.

`NAME`
A valid `cn` (common name).

`OCTAL_MODE`
A three-digit, octal file permissions specifier. The first digit specifies permissions for the server user ID, the second for the server group ID, the last for other users. Each digit consists of a bitmask defining read (`4`), write (`2`), execute (`1`), or no access (`0`) permissions, thus `640` specifies read-write access for the server user, read-only access for other users of the server group, and no access for other users.

`PASSWORD_FILE`
The full path to the file from which the bind password should be read.

`PATH`
A valid, absolute file system path.

`STRING`
A DirectoryString value, as specified by RFC 2252 (`http://www.ietf.org/rfc/rfc2252.txt`).

`SUPPORTED_SSL_CIPHER`
An SSL cipher supported by the server. See the Reference for a list of supported ciphers.

`SUPPORTED_SSL_PROTOCOL`
An SSL protocol supported by the server. See the Reference for a list of supported protocols.

`TIME`
A time of the form *hhmm* in 24-hour format, where *hh* stands for hours and *mm* stands for minutes.

Exit Status

The following exit status values are returned:

`0`	Successful completion
non-zero	An error occurred

Examples

This section contains examples of how the `dpconf` command is used.

EXAMPLE 1 Getting Help With a Subcommand

This example shows how to get help for using a subcommand:

```
$ dpconf create-connection-handler -?
Usage: dpconf create-connection-handler NAME [NAME ...]
Create new connection handlers
For global options, use dpconf --help.
NAME  The name of a connection handler
For more information, see dpconf(1M).
```

EXAMPLE 2 Getting Information About Properties

This example shows how to get information about the properties of the resource limits policy.

- To view the properties exposed by all of the `dpconf` subcommands, run this command:

  ```
  $ dpconf help-properties
  ```

EXAMPLE 3 Getting Properties for Access Logs

This example shows how to get the access log properties, specifying that the `log-rotation-size` property is quoted in bytes.

EXAMPLE 3 Getting Properties for Access Logs *(Continued)*

```
$ dpconf get-access-log-prop -h host -p port
-Z b
default-log-level                  :  info
log-file-name                      :  logs/access
log-file-perm                      :  600
log-level-client-connections       :  -
log-level-client-disconnections    :  -
log-level-client-operations        :  -
log-level-connection-handlers      :  -
log-level-data-sources             :  -
log-level-data-sources-detailed    :  -
log-rotation-frequency             :  1h
log-rotation-policy                :  size
log-rotation-size                  :  104,857,600b
log-rotation-start-day             :  1
log-rotation-start-time            :  0000
log-search-filters                 :  false
max-log-files                      :  10
```

EXAMPLE 4 Customizing Search Limits

This example shows how to define customized limits for search operations, based on the search base and search scope.

1. Create a custom search limit.

   ```
   $ dpconf create-custom-search-size-limit -h host -p port
   POLICY-NAME LIMIT-NAME
   ```

2. Set the criteria for the custom search limit.

   ```
   $ dpconf set-custom-search-size-limit-prop -h host -p port
     POLICY-NAME LIMIT-NAME
     one-level-search-base-dn:VALUE
     subtree-search-base-dn:VALUE
   ```

3. Define the limit for the number of results returned when a search meets one of the above criteria.

   ```
   $ dpconf set-custom-search-size-limit-prop -h host -p port
     POLICY-NAME CUSTOM-SEARCH-LIMIT-NAME
     search-size-limit:VALUE
   ```

4. View the properties of a custom search limit.

   ```
   $ dpconf get-custom-search-size-limit-prop -h host -p port
   POLICY-NAME LIMIT-NAME
   ```

EXAMPLE 5 Comparing Properties of Connection Handlers

This example shows how to view the properties of one connection handler and how to compare the properties of a set of connection handlers.

1. View all of the properties of one connection handler.

   ```
   $ dpconf get-connection-handler-prop -h host -p port
   CONNECTION-HANDLER-NAME
   ```

   These are the default properties of a connection handler:

   ```
   allowed-auth-methods           :  anonymous
   allowed-auth-methods           :  sasl
   allowed-auth-methods           :  simple
   allowed-ldap-ports             :  ldap
   allowed-ldap-ports             :  ldaps
   bind-dn-filters                :  any
   data-view-routing-custom-list  :  -
   data-view-routing-policy       :  all-routable
   description                    :  -
   domain-name-filters            :  any
   enable-data-view-affinity      :  false
   ip-address-filters             :  any
   is-enabled                     :  false
   is-ssl-mandatory               :  false
   priority                       :  99
   request-filtering-policy       :  no-filtering
   resource-limits-policy         :  no-limits
   user-filter                    :  any
   ```

2. View the key properties and relative priorities of all of the connection handlers.

   ```
   $ dpconf list-connection-handlers -v
     Name                           is-enabled  priority  description
   --------------------------  ----------  --------  ---------------------------
   anonymous                   false       99        unauthenticated connections
   myconnectionhandler         true        99        -
   default connection handler  true        100       default connection handler
   ```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also dpadm(1M), dsconf(1M), and dsadm(1M)

Name dsadm – Manages a Directory Server instance

Synopsis *install-path*`/ds6/bin/dsadm`
subcommand options

Description The `dsadm` command is the local administration command for Directory Server instances. Use the `dsadm` command with any of the subcommands described in this man page.

`dsadm` must be used while the server is stopped (except subcommands `dsadm info`, `dsadm stop` and `dsadm restart`). It must be run from the local machine where the server instance is located. This command must be run by the username that is the Operating System owner of the server instance, or by root.

Subcommands The following subcommands are supported:

`dsadm add-cert [-Ci] [-W` *CERT_PW_FILE*`]` *INSTANCE_PATH CERT_ALIAS CERT_FILE*
Adds a certificate to the certificate database.

`dsadm add-selfsign-cert [-i] [-W` *CERT_PW_FILE*`] [-S` *DN*`]` *INSTANCE_PATH CERT_ALIAS*
OR

`dsadm add-selfsign-cert [-i] [-W` *CERT_PW_FILE*`] [--name` *NAME* `] [--org` *ORG*`] [--org-unit` *ORG-UNIT*`] [--city` *CITY*`] [--state` *STATE*`] [--country` *COUNTRY*`]` *INSTANCE_PATH CERT_ALIAS*
Creates a self-signed certificate and adds it to the certificate database.

`dsadm autostart [--off [-i]]` *INSTANCE_PATH*
Enables or disables Directory Server instance startup at system boot. This command is only available if you installed with Sun Java Enterprise System or native packages, and is not available on Windows. This command must be run as root.

`dsadm backup [-f` *FLAG*`] ...` *INSTANCE_PATH ARCHIVE_DIR*
Creates a backup archive of the Directory Server instance.

`dsadm create [-BiG] [-u` *USER_NAME*`] -g` *GROUP_NAME*`] [-h` *HOST_NAME* `[-p` *PORT*`] [-P` *SSL_PORT*`] [-D` *DN*`] [-w` *PW_FILE*`]` *INSTANCE_PATH*
Creates a Directory Server instance.

`dsadm delete` *INSTANCE_PATH*
Deletes a Directory Server instance.

`dsadm disable-service [-T` *TYPE*`]` *INSTANCE_PATH*
Disables a Directory Server instance from being managed as a service. This command is available on Windows distributions and on Solaris native package distributions only. The command must be run as root.

`dsadm enable-service [-T` *TYPE*`]` *INSTANCE_PATH* `[`*RESOURCE_GRP*`]`

Enables a Directory Server instance to be managed as a service. This command is available on Windows distributions and on Solaris native package distributions only. The command must be run as root.

`dsadm export [-biQ] [-s` *DN*`] ... [-x` *DN*`] ... [-f` *FLAG*`] ... [-y [-w` *CERT_PW_FILE*`]]` *INSTANCE_PATH* *SUFFIX_DN* `[` *SUFFIX_DN* `...]` *LIDF_FILE*

Exports suffix to LDIF format.

`dsadm export-cert [-i] [-w` *CERT_PW_FILE*`] [-o` *OUTPUT_FILE*`] [-O` *OUTPUT_PW_FILE*`]` *INSTANCE_PATH* *CERT_ALIAS*

Exports an encrypted copy of the certificate and its public and private keys from the certificate database.

`dsadm generate-legacy-scripts [ -i ]` *INSTANCE_PATH*

Generates legacy scripts in a Directory Server instance. This command is not available on Windows.

`dsadm get-flags` *INSTANCE_PATH* `[`*FLAG* `...]`

Displays the flag values for the Directory Server instance.

`dsadm import [-biK] [-x` *DN*`] ... [-f` *FLAG=VAL*`] ... [-y [-W` *CERT_PW_FILE*`]]` *INSTANCE_PATH* *LDIF_FILE* `[`*LDIF_FILE* `...]` *SUFFIX_DN*

Populates an existing suffix with LDIF data.

`dsadm import-cert [-i] [-W` *CERT_PW_FILE*`] [-I` *INPUT_PW_FILE*`]` *INSTANCE_PATH* *CERT_FILE*

Adds a new certificate and its keys to the certificate database.

`dsadm import-selfsign-cert [-i] [-W` *CERT_PW_FILE*`] [-I` *INPUT_PW_FILE*`]` *INSTANCE_PATH CERT_FILE*

Adds a new self-signed certificate and its keys to the certificate database.

`dsadm info` *INSTANCE_PATH*

Displays Directory Server instance status and some configuration information.

`dsadm list-certs [-Ci] [-W` *CERT_PW_FILE*`]` *INSTANCE_PATH*

Lists all certificates in the certificate database.

`dsadm reindex [-bl] -t` *ATTR_INDEX* `[-t` *ATTR_INDEX* `...]` *INSTANCE_PATH* *SUFFIX_DN*

Regenerates existing indexes.

`dsadm remove-cert [-i] [-W` *CERT_PW_FILE*`]` *INSTANCE_PATH* *CERT_ALIAS*

Removes a certificate from the certificate database. The instance must be stopped before running this command.

`dsadm renew-cert [-i] [-W` *CERT_PW_FILE*`]` *INSTANCE_PATH* *CERT_ALIAS* *CERT_FILE*

Replaces a certificate, but keeps the existing private key. The instance must be stopped before running this command.

`dsadm renew-selfsign-cert [-i] [-W` *CERT_PW_FILE*`]` *INSTANCE_PATH* *CERT_ALIAS*
Renews a self-signed certificate in the certificate database. The instance must be stopped before running this command.

`dsadm repack [-b` *backend*`]` *INSTANCE_PATH* *SUFFIX_DN* `[`*SUFFIX_DN...*`]`
Repacks or compacts an existing suffix. The `-b` option enables you to specify the name of the back end instead of the suffix name. At least one suffix DN or one back end name must be specified. The instance must be stopped before running this command.

`dsadm request-cert [-i] [-W` *CERT_PW_FILE*`] -s` *DN* `[-F` *FORMAT*`] [-o` *OUTPUT_FILE*`]` *INSTANCE_PATH*
Or:
`dsadm request-cert [-i] [-W` *CERT_PW_FILE*`] --name` *NAME* `[--org` *ORG*`] [--org-unit` *ORG-UNIT*`] [--city` *CITY*`] [--state` *STATE*`] [--country` *COUNTRY*`] [-F` *FORMAT*`] [-o` *OUTPUT_FILE*`]` *INSTANCE_PATH*
Generates a certificate request.

`dsadm restart [-i] [-W` *CERT_PW_FILE*`]` *INSTANCE_PATH*
Restarts a Directory Server instance.

`dsadm restore [-i]` *INSTANCE_PATH* *ARCHIVE_DIR*
Restores Directory Server instance from a backup archive.

`dsadm set-flags [-i] [-W` *CERT_PW_FILE*`]` *INSTANCE_PATH* *FLAG=VAL* `[`*FLAG=VAL* `...]`
Sets flags for a Directory Server instance.

`dsadm show-access-log -A` *DURATION* *INSTANCE_PATH*
OR

`dsadm show-access-log -L` *LAST_LINES* *INSTANCE_PATH*
Displays the contents of the access log.

`dsadm show-cert [-i] [-W` *CERT_PW_FILE*`] [-o` *OUTPUT_FILE*`] [-F` *FORMAT*`]` *INSTANCE_PATH* `[`*CERT_ALIAS*`]`
Displays a certificate.

`dsadm show-error-log -A` *DURATION* *INSTANCE_PATH*
OR

`dsadm show-error-log -L` *LAST_LINES* *INSTANCE_PATH*
Displays the contents of the error log.

`dsadm start [-Ei] [-W` *CERT_PW_FILE*`]` *INSTANCE_PATH*
Starts a Directory Server instance.

`dsadm stop` *INSTANCE_PATH*
Stops a Directory Server instance.

Global Options The following options are global, and are applicable to all commands and subcommands.

`--?`
`--help` Displays help information for a command or subcommand.

`-V`
`--version` Displays the current version of `dsadm`. The version is provided in the format *year.day.time*. So version number `2006.178.0035` was built on the 178th day of 2006 at 00h35. If the components used by `dsadm` are not aligned, the version of each individual component is displayed.

Subcommand Options The following options are applicable to the subcommands where they are specified.

`-A` *DURATION*
`--max-age` *DURATION* Specifies the maximum age of lines to be returned from the access log or the error log. For example, to search for all entries younger than 24 hours, use `-A 24h`.

`-B`
`--below` Creates the Directory Server instance in an existing directory, specified by the *INSTANCE_PATH*. The existing directory must be empty. On UNIX machines, the user who runs this command must be root, or must be the owner of the existing directory. If the user is root, the instance will be owned by the owner of the existing directory.

`--C`
`--ca` Specifies a Certificate Authority certificate is to be used, or that the command should display information about CA certificates.

`--city` *CITY* Adds `L=CITY` to the subject DN. Default is none.

`--country` *COUNTRY* Adds `C=COUNTRY` to the subject DN. The default is none.

`-D` *DN*
`--rootDN` *DN* Defines the Directory Manager DN. The default is `cn=Directory Manager`.

`-E`
`--safe` Starts Directory Server with the configuration used at the last successful startup.

`-F` *FORMAT*
`--format` *FORMAT*

Specifies output format. For `dsadm request-cert`, the default is `der`, and the other possible output format is `ascii`. .For `dsadm show-cert`, the default is `readable`, and other possible output formats are `ascii` and `der`.

`-f` *FLAG*
`--flags` *FLAG* or *FLAG=VAL*

Customized values for options.

Possible flags for the `dsadm backup` subcommand are as follows.

`verify-db`	Check database integrity.

Possible flags for the `dsadm export` subcommand are as follows.

`minimal-encode`	Perform minimal base64 encoding.
`multiple-output-file`	Generate multiple LDIF output files.
`not-export-unique-id`	Do not export the unique ID generated on import.
`not-folded-output`	Do not fold long lines.
`no-num-version`	Delete the initial line specifying the LDIF version, `version: 1`, for backward compatibility.
`not-print-entry-ids`	Do not include entry IDs in the LDIF output.
`use-main-db-file`	Only export from the main database file.

Possible flags for the `dsadm import` subcommand are as follows.

`chunk-size`	Merge chunk size.
`incremental-output-file`	Import LDIF generated during incremental import.

`-G`
`--no-legacy-scripts`

Does not create legacy scripts. If you do not use this option, command scripts that are similar to 5.x command scripts are created in the server instance.

`-g` *GROUP_NAME*
`--groupname` *GROUP_NAME*

Sets the server instance owner's group ID. The default is the user's current UNIX group. This option is not available on Windows.

`-h` *HOST_NAME*
`--hostname` *HOST_NAME*

Specifies the hostname. The default is the name of the current host system.

`-I` *INPUT_PW_FILE*
`--input-pwd-file` *INPUT_PW_FILE*

Reads the input file password in the `INPUT_PW_FILE` file. The default is a prompt for password.

`-i`
`--no-inter`

Does not prompt for confirmation before performing the operation.

`-K`
`--incremental`

Specifies that the contents of the imported LDIF file are appended to the existing LDAP entries. If this option is not specified, the contents of the imported file replace the existing entries.

`-L` *LAST_LINES*
`--last-lines` *LAST_LINES*

Specifies the number of lines to be returned from the access log or the error log. *LAST_LINES* must be an integer. For example, to return the last 50 lines, use `-L 50`. If no value is specified, the default number of lines returned is 20.

`--l`
`--vlv`
Specifies VLV (browsing) index.

`--name` *NAME*
Adds `CN=NAME` to the subject DN.

`--O` *OUTPUT_PW_FILE*
`--output-pwd-file` *OUTPUT_PW_FILE*
Reads the output password from the `OUTPUT_FILE` file. The default is a prompt for password.

`--o` *OUTPUT_FILE*
`--output` *OUTPUT_FILE*
Stores the command results in the `OUTPUT_FILE` file. The default is `stdout`, standard output.

`--off`
Disables server instance startup at system boot.

`--org` *ORG*
Adds `O=ORG` to the subject DN. The default is none.

`--org-unit` *ORG-UNIT*
Adds `O=ORG-UNIT` to the subject DN. The default is none.

`--P` *SSL_PORT*
`--ssl-port` *SSL_PORT*
Specifies the secure SSL port for LDAP traffic. The default is `636` if dsadm is run by the root user, or `1636` if dsadm is run by a non-root user.

`--p` *PORT*
`--port` *PORT*
Specifies the port for LDAP traffic. The default is `389` if dsadm is run by the root user, or `1389` if dsadm is run by a non-root user.

`--Q --no-repl`
Specifies that additional data needed for replication is not included in the export.

`--S` *DN*
`--subject` *DN*
Specifies the subject DN. The default depends on the subcommand used, and is either `CN=hostname` or `CN=CERT_ALIAS`.

`--s` *DN*
`--include` *DN*
Exports data from suffix DN.

`--state` *STATE*
Adds `ST=STATE` to the subject DN. Default is none.

`--T` *TYPE*
`--type` *TYPE*

Service type. Can be `CLUSTER` when using Sun Cluster, `SMF` when using Solaris 10, or `WIN_SERVICE` when using Windows.

`--t` *ATTR_INDEX*
`--attr` *ATTR_INDEX*

Specifies attribute index *ATTR_INDEX*

`--u` *USER_NAME*
`--username` *USER_NAME*

Sets the server instance owner user ID. The default is the current UNIX user name. This option is not available on Windows.

`--W` *CERT_PW_FILE*
`--cert-pwd-file` *CERT_PW_FILE*

Reads certificate database password from `CERT_PW_FILE`. The default is to prompt for password.

`--w` *PW_FILE*
`--pwd-file` *PW_FILE*

Sets the password file for the Directory Manager (-D). The default is to prompt for password.

`--x` *DN*
`--exclude` *DN*

Excludes the specified DN from the command.

`--y`
`--decrypt-attr`

Decrypts encrypted attributes.

Operands The following operands are supported:

ARCHIVE_DIR Specifies the path to the backup of the Directory Server instance.

CERT_ALIAS Certificate alias name. A user-specified name that identifies a certificate.

CERT_FILE Specifies the file that contains the certificate.

FLAG Specifies a flag that represents a property operand when using the command `dsadm get-flags`. Possible flag: `cert-pwd-prompt`.

FLAG=VAL Specifies a property flag operand and its value when using the command `dsadm set-flags`.

`cert-pwd-prompt` flag possible values are: `off` `on`. Default: `off`. By default the dsadm command generates a certificate database password when creating a server instance. This password is stored, allowing dsadm to access the certificate database when necessary, for example, when the server starts listening for SSL connections. When the

	`cert-pwd-prompt` flag is changed to `on`, the `dsadm` command prompts for the certificate database password when needed.
INSTANCE_PATH	Path of the Directory Server instance.
LDIF_FILE	Filename of LDIF file.
RESOURCE_GRP	Cluster resource group. Required for `CLUSTER` service, not applicable for other types of services.
SUFFIX_DN	Suffix DN (Distinguished name).

Exit Status

The following exit status values are returned:

`0`	Successful completion.
non-zero	An error occurred.

Examples

The following examples show how the `dsadm` command is used.

EXAMPLE 1 Creating a Directory Server Instance

```
$ dsadm create -p 6389 -P 6636 /local/ds
```

This command creates the server instance files in the directory `/local/ds`. The server instance is owned by the UNIX user who creates the command.

In this example, the LDAP port is specified as 6389, and the secure port is specified as 6636. If you do not specify port numbers, the default port numbers 389 and 636 (for root user) or 1389 and 1636 (for not-root user) are used. If you do not specify port numbers and the default port numbers are already being used, the `dsadm create` command aborts.

EXAMPLE 2 Starting a Directory Server Instance

The server instance path is `/local/ds`.

```
$ dsadm start /local/ds
```

EXAMPLE 3 Getting Information About a Directory Server instance

This command shows information such as the owner, ports, and current state of the server instance. The instance path is `/local/ds`.

```
$ dsadm info /local/ds
```

EXAMPLE 4 Importing an LDIF File

Import an LDIF file, specifying that no user confirmation is required, and giving the suffix DN.

```
$ dsadm import -i /local/ds /local/ds/ldif/example.ldif \
dc=example,dc=com
```

EXAMPLE 5 Exporting an LDIF File

Export a suffix to an LDIF file.

```
$ dsadm export -x ou=People,dc=example,dc=com /local/ds \
dc=example,dc=com /local/ds/ldif/export.ldif
```

This command shows all data in the suffix `dc=example,dc=com`, excluding data in the subsuffix `ou=People,dc=example,dc=com`

EXAMPLE 6 Backing Up a Directory Server Instance

This command backs up the suffix data and the configuration data. The instance path is `/local/ds` and the archive directory is `/local/dsbackup/20060722`.

```
$ dsadm backup /local/ds /local/dsbackup/20060722
```

EXAMPLE 7 Regenerating Attribute Indexes

To regenerate the existing `cn` and `uid` indexes:

```
$ dsadm reindex -t cn -t uid /local/ds dc=example,dc=com
```

EXAMPLE 8 Renewing a Certificate

Use the following command to renew an existing server certificate with a new server certificate from your Certificate Authority.

```
$ dsadm renew-cert /local/ds cert_alias /local/certfiles/new-cert
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

See Also dsconf(1M)

Name dsccmon – Monitor servers registered with Directory Service Control Center

Synopsis *install-path*`/dscc6/bin/dsccmon`

```
 [subcommand] [options]
```

Description The `dsccmon` command is used to monitor servers registered with Directory Service Control Center. Use the `dsccmon` command with the subcommands described in this man page.

Subcommands The following subcommands are supported:

`dsccmon view-repl-agmts`
Show monitoring information about the replication agreements between Directory Server instances.

The format of this subcommand is:

`dsccmon view-repl-agmts [-d` *seconds*`] [-b] [-s` *suffix-dn*`]...`

`dsccmon view-servers`
Show monitoring information about registered servers.

The format of this subcommand is:

`dsccmon view-servers [-d` *seconds*`] [-t] [-E]`

`dsccmon view-suffixes`
Show monitoring information about suffixes supported by registered servers.

The usage of this subcommand is:

`dsccmon view-suffixes [-d` *seconds*`] [-b] [-G] [-s` *suffix-dn*`]...`

Global Options The following options apply to all commands and subcommands:

`-?`
`--help`
Display usage for the command or for the specified subcommand.

`-D` *user-dn*
`--user-dn` *user-dn*
Bind using the specified *user-dn*.

By default, the value of the environment variable `LDAP_ADMIN_USER` is used. If `LDAP_ADMIN_USER` is not defined, `cn=admin,cn=Administrators,cn=dcc` is used.

`-a`
`--all`
Display hidden suffixes or servers, such as the server and suffixes used by Directory Service Control Center to manage metainformation about the directory service.

`-h` *hostname*
`--hostname` *hostname*
Connect to the Directory Service Control Center registry on the specified host or IP address.

By default, the value of the environment variable `DSCC_HOST` is used. If `DSCC_HOST` is not defined, `localhost` is used.

For example, when mapping the IPv4 address `192.168.0.99` to IPv6, pass the `-h` option with its argument as `-h ::ffff:192.168.0.99`.

`-p` *port-number*
`--port` *port-number*

Connect to the Directory Service Control Center registry on the specified port.

By default, the value of the environment variable `DSCC_PORT` is used. If `DSCC_PORT` is not defined, `3998` is used.

`-u` *uid*
`--username` *uid*

Bind using `cn=`*uid*`,cn=Administrators,cn=dcc`.

By default, the value of the environment variable `LDAP_ADMIN_USER` is used. If `LDAP_ADMIN_USER` is not defined, `cn=admin,cn=Administrators,cn=dcc` is used.

`-V`
`--version`

Displays the current version of `dsccmon`. The version is provided in the format *year.day.time*. So version number `2006.178.0035` was built on the 178th day of 2006 at 00h35. If the components used by `dsccmon` are not aligned, the version of each individual component is displayed.

`-v`
`--verbose`

Display extra information for debugging purposes.

`-w` *file*
`--pwd-file` *file*

Bind using the password specified in *file*.

By default, the value of the environment variable `LDAP_ADMIN_PWF` is used. If `LDAP_ADMIN_PWF` is not defined, `dsccmon` prompts for a password.

Subcommand Options

The following options apply to the subcommands where they are specified:

`-E`
`--error`

Display detailed server error information.

`-G`
`--genid`

Display generation IDs.

`-b`
`--brief`

Do not display nonessential data, such as headers and notes.

`-d` *seconds*
`--period` *seconds* — Update monitoring information each specified number of seconds.

`-s` *suffix-dn*
`--suffix` *suffix-dn* — Display information for the specified suffix only.

`-t`
`--ipath` — Display the server instance path.

Environment Variables

The following environment variables are supported:

`DSCC_HOST`	Bind to the registry on this host.
`DSCC_PORT`	Bind to the registry on this port number.
`LDAP_ADMIN_PWF`	Read the bind password from this file.
`LDAP_ADMIN_USER`	Bind with this user DN or uid.

Exit Status

The following exit status values are returned:

`0`	Successful completion
non-zero	An error occurred.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-console-cli
Stability Level	Evolving

See Also

dsccreg(1M)

Name

dsccreg – Register servers with Directory Service Control Center

Synopsis

```
install-path/dscc6/bin/dsccreg
 [subcommand] [options]
```

Description

The `dsccreg` command is used to register server instances on the local system with Directory Service Control Center, which may be remote. Use the `dsccreg` command with the subcommands described in this man page.

Subcommands

The following subcommands are supported:

`dsccreg add-server`

Add a server instance to the Directory Service Control Center registry.

The format of this subcommand is:

```
dsccreg add-server [-B instance-user-dn] [-G instance-pwd-file]
 [-d desc] [-H local-host] instance-path
```

`dsccreg list-servers`

List server instances registered with Directory Service Control Center.

The format of this subcommand is:

```
dsccreg list-servers [-a] [-C]
```

`dsccreg remove-server`

Remove a server instance from the Directory Service Control Center registry.

The usage of this subcommand is:

```
dsccreg remove-server [-B instance-user-dn] [-G instance-pwd-file]
 [-H local-host] instance-path
```

Global Options

The following options apply to all commands and subcommands:

`-?`
`--help`

Display usage for the command or for the specified subcommand.

`-D` *user-dn*
`--user-dn` *user-dn*

Bind using the specified *user-dn*.

By default, the value of the environment variable `LDAP_ADMIN_USER` is used. If `LDAP_ADMIN_USER` is not defined, `cn=admin,cn=Administrators,cn=dcc` is used.

`-h` *hostname*
`--hostname` *hostname*

Connect to the Directory Service Control Center registry on the specified host or IP address.

By default, the value of the environment variable `DSCC_HOST` is used. If `DSCC_HOST` is not defined, `localhost` is used.

For example, when mapping the IPv4 address `192.168.0.99` to IPv6, pass the `-h` option with its argument as `-h ::ffff:192.168.0.99`.

`-i`
`--no-inter`
: Do not prompt for confirmation before restarting servers.

`-p` *port-number*
`--port` *port-number*
: Connect to the Directory Service Control Center registry on the specified port.

By default, the value of the environment variable `DSCC_PORT` is used. If `DSCC_PORT` is not defined, `3998` is used.

`-u` *uid*
`--username` *uid*
: Bind using `cn=`*uid*`,cn=Administrators,cn=dcc`.

By default, the value of the environment variable `LDAP_ADMIN_USER` is used. If `LDAP_ADMIN_USER` is not defined, `cn=admin,cn=Administrators,cn=dcc` is used.

`-V`
`--version`
: Displays the current version of `dsccreg`. The version is provided in the format *year.day.time*. So version number `2006.178.0035` was built on the 178th day of 2006 at 00h35. If the components used by `dsccreg` are not aligned, the version of each individual component is displayed.

`-v`
`--verbose`
: Display extra information for debugging purposes.

`-w` *file*
`--pwd-file` *file*
: Bind using the password specified in *file*.

By default, the value of the environment variable `LDAP_ADMIN_PWF` is used. If `LDAP_ADMIN_PWF` is not defined, `dsccreg` prompts for a password.

Subcommand Options

The following options apply to the subcommands where they are specified:

`-B` *instance-user-dn*
`--inst-user-dn` *instance-user-dn*
: Use the specified bind DN to bind to the instance specified by *instance-path*.

By default, the `dsccreg` command uses `cn=Directory Manager`.

`-C`
`--check-access`

Verify that each registered server instance is accessible from Directory Service Control Center.

`-G` *instance-pwd-file*
`--inst-pwd-file` *instance-pwd-file*

Use the password in the specified file to bind to the instance specified by *instance-path*.

By default, the `dsccreg` command prompts for the password.

`-H` *hostname*
`--current-host` *hostname*

Use the specified host name as the local host.

By default, the `dsccreg` command uses the local host name returned by the operating system.

`-a`
`--all`

Display hidden servers, such as the server used by Directory Service Control Center to manage metainformation about the directory service.

`-d` *desc*
`--description` *desc*

Use the specified text *desc* as the description for the server instance.

Operands

The following subcommand operands are supported:

instance-path Full path to the server instance.

Environment Variables

The following environment variables are supported:

`DSCC_HOST`	Bind to the registry on this host.
`DSCC_PORT`	Bind to the registry on this port number.
`LDAP_ADMIN_PWF`	Read the bind password from this file.
`LDAP_ADMIN_USER`	Bind with this user DN or uid.

Exit Status

The following exit status values are returned:

`0`	Successful completion
non-zero	An error occurred.

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-console-cli
Stability Level	Evolving

See Also

dsccmon(1M)

Name

dsccsetup – Set up Directory Service Control Center

Synopsis

```
install-path/dscc6/bin/dsccsetup
 [subcommand] [options]
```

Description

The `dsccsetup` command is used to register Directory Service Control Center with Sun Java Web Console (DSCC), and to register local agents of the administration framework. Use the `dsccsetup` command with the subcommands described in this man page.

Subcommands

The following subcommands are supported:

`dsccsetup ads-create [-w` *file*`]`

Initialize the DSCC registry, a local Directory Server instance for private use by DSCC to store configuration information. DSCC requires that this instance reside locally on the host where you run DSCC. Therefore, if you replicate the data in the instance for high availablity, set up one DSCC per replica host.

If you do not provide the Directory Manager password for the DSCC registry in the file passed to the `-w` option, the command prompts for the password.

The default port numbers used by the instance are 3998 for LDAP, and 3999 for LDAPS.

The default instance path is `/var/opt/SUNWdsee/dscc6/dcc/ads` on Solaris systems, `/var/opt/sun/dscc6/dcc/ads` on HP-UX and Red Hat systems, and `C:\Program Files\Sun\DSEE\var\dscc6\dcc\ads` on Windows systems.

The base DN for the suffix containing configuration information is `cn=dscc`. Use the `dsccsetup status` subcommand to read actual values for the DSCC registry instance.

`dsccsetup ads-delete`

Delete the Directory Server instance used by DSCC to store configuration information.

Use the `-i` when not using the command interactively.

`dsccsetup cacao-reg [-t]`

Register the local DSCC agent with the Common Agent Container, `cacao`.

Use the `-t` option if you want to restart the Common Agent Container manually at a later time.

`dsccsetup cacao-unreg`

Remove the local DSCC agent registration information from `cacao`.

`dsccsetup console-reg [-t]`

Register DSCC with the web application container, Sun Java Web Console.

Use the `-i` when not using the command interactively.

Use the `-t` option if you want to restart Sun Java Web Console manually at a later time.

`dsccsetup console-unreg [-t]`

Remove DSCC from Sun Java Web Console.

Use the `-i` when not using the command interactively.

Use the `-t` option if you want to restart Sun Java Web Console manually at a later time.

`dsccsetup dismantle [-t]`

Dismantle the DSCC administration framework, running the `cacao-unreg`, `console-unreg`, and `ads-delete` subcommands.

Use the `-i` when not using the command interactively.

Use the `-t` option if you want to restart Sun Java Web Console, and the Common Agent Container manually at a later time.

`dsccsetup initialize [-t] [-w` *file*`]`

Initialize the DSCC administration framework, running the `ads-create`, `console-reg`, and `cacao-reg` subcommands.

Use the `-i` when not using the command interactively.

Use the `-t` option if you want to restart Sun Java Web Console, or the Common Agent Container manually at a later time.

If you do not provide the Directory Manager password for the DSCC registry in the file passed to the `-w` option, the command prompts for the password.

`dsccsetup status`
Display whether DSCC has been registered with Sun Java Web Console, and with the Common Agent Container. Also, display whether the DSCC registry has been initialized.

`dsccsetup mfwk-reg [-t]`
Register the local Directory Server monitoring agent for Java Enterprise System Monitoring Framework with the Common Agent Container, `cacao`.

Use the `-t` option if you want to restart the Common Agent Container manually at a later time.

`dsccsetup mfwk-unreg`
Remove the local Directory Server monitoring agent registration information from `cacao`.

Global Options

The following options apply to all commands and subcommands:

`-?`
`--help`
Display usage for the command or for the specified subcommand.

`-i`
`--no-inter`
Do not prompt for confirmation before performing the operation.

`-V`
`--version`
Displays the current version of `dsccsetup`. The version is provided in the format *year.day.time*. So version number `2006.178.0035` was built on the 178th day of 2006 at 00h35. If the components used by `dsccsetup` are not aligned, the version of each individual component is displayed.

`-v`
`--verbose`
Display extra information for debugging purposes.

Subcommand Options

The following options apply to the subcommands where they are specified:

`-t`
`--norestart`
Do not restart the Common Agent Container or Sun Java Web Console after performing the operation.

You can restart the Common Agent Container using the `cacaoadm` command. You can restart the Sun Java Web Console using the `smcwebserver` command.

`-w` *file*
`--pwd-file` *file* Use the Directory Service Manager password specified in *file*.

By default, `dsccsetup` prompts for a password.

Exit Status

The following exit status values are returned:

`0`	Successful completion
non-zero	An error occurred.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-console-agent
Stability Level	Unstable

See Also

`cacaoadm`(1M), `smcwebserver`(1M)

Name dsconf – Manages Directory Server configuration

Synopsis *install-path*`/ds6/bin/dsconf`
subcommand options

Description The `dsconf` command manages Directory Server configuration. It enables you to modify the configuration entries in `cn=config`.

The server must be running in order for you to run `dsconf`.

Subcommands The following subcommands are supported:

`dsconf accord-repl-agmt [-h` *host*`] [-p` *port*`] [-I` *dest-bind-dn*`] [-W` *dest-pw-file*`]` *SUFFIX_DN HOST*:*PORT* `[`*HOST*:*PORT* `...]`
Ensures the authentication properties of the destination suffix are in accord with those of the replication agreement.

`dsconf backup [-h` *host*`] [-p` *port*`] [-a]` *ARCHIVE_DIR*
Backs up Directory Server data (configuration data excluded).

`dsconf change-repl-dest [-h` *host*`] [-p` *port*`] [-A` *NEW_PROTOCOL*`]` *SUFFIX_DN HOST*:*PORT NEW_HOST*:*NEW_PORT*
Changes the remote replica pointed to by an existing replication agreement. The suffix DN and configuration of the existing agreement remain the same.

`dsconf create-encrypted-attr [-h` *host*`] [-p` *port*`] [--desc` *DESC*`]` *SUFFIX_DN ATTR_NAME* `[`*ATTR_NAME* `...]` *ENCRYPTION_ALGO*
Declares that the values for an attribute are encrypted.

`dsconf create-index [-h` *host*`] [-p` *port*`]` *SUFFIX_DN ATTR_NAME* `[`*ATTR_NAME* `...]`
Declares that an attribute is indexed. The default index types for the attribute are equality and presence.

`dsconf create-plugin [-h` *host*`] [-p` *port*`] -H` *LIB_PATH* `-F` *INIT_FUNCT* `-Y` *TYPE* `[-G` *ARG*`]...` *PLUGIN_NAME*
Declares a new client plugin. The plugin state is disabled.

`dsconf create-repl-agmt [-h` *host*`] [-p` *port*`] [-A` *PROTOCOL*`]` *SUFFIX_DN HOST*:*PORT* `[`*HOST*:*PORT* `...]`
Creates a replication agreement for existing suffix.

`dsconf create-repl-priority [-h` *host*`] [-p` *port*`]` *SUFFIX_DN PRIORITY_NAME PROP*:*VAL* `[`*PROP*:*VAL* `...]`
Creates a prioritized replication rule on a master.

`dsconf create-suffix [-h` *host*`] [-p` *port*`] [-B` *NAME*`] [-L` *FILE*`] [-N]` *SUFFIX_DN* `[`*SUFFIX_DN* `...]`
Creates a suffix.

`dsconf delete-encrypted-attr [-h` *host*`] [-p` *port*`]` *SUFFIX_DN ATTR_NAME* `[`*ATTR_NAME* `...]`
Declares that the values for an attribute are no longer encrypted.

`dsconf delete-index [-h` *host*`] [-p` *port*`]` *SUFFIX_DN ATTR_NAME* `[`*ATTR_NAME* `...]`
Declares that an attribute is no longer indexed.

`dsconf delete-plugin [-h` *host*`] [-p` *port*`]` *PLUGIN_NAME* `[`*PLUGIN_NAME* `...]`
Declares that a plugin can not be used by the server any more.

`dsconf delete-repl-agmt [-h` *host*`] [-p` *port*`]` *SUFFIX_DN HOST*`:`*PORT* `[`*HOST*`:`*PORT* `...]`
Deletes a replication agreement.

`dsconf delete-repl-priority [-h` *host*`] [-p` *port*`]` *SUFFIX_DN PRIORITY_NAME* `[`*PRIORITY_NAME* `...]`
Deletes a prioritized replication rule.

`dsconf delete-suffix [-h` *host*`] [-p` *port*`]` *SUFFIX_DN* `[`*SUFFIX_DN* `...]`
Deletes suffix configuration and data.

`dsconf demote-repl [-h` *host*`] [-p` *port*`]` *SUFFIX_DN* `[`*SUFFIX_DN* `...]`
Demotes the role of an existing replicated suffix. A master is demoted to a hub, a hub is demoted to a consumer. To demote a master to a consumer, run the command twice.

`dsconf disable-plugin [-h` *host*`] [-p` *port*`]` *PLUGIN_NAME* `[`*PLUGIN_NAME* `...]`
Disables a plugin.

`dsconf disable-repl [-h` *host*`] [-p` *port*`]` *SUFFIX_DN* `[`*SUFFIX_DN* `...]`
Disables replication for a replicated suffix.

`dsconf disable-repl-agmt [-h` *host*`] [-p` *port*`]` *SUFFIX_DN HOST*`:`*PORT* `[`*HOST*`:`*PORT* `...]`
Disables replication with another Directory Server.

`dsconf enable-plugin [-h` *host*`] [-p` *port*`]` *PLUGIN_NAME* `[`*PLUGIN_NAME* `...]`
Enables a plugin.

`dsconf enable-repl [-h` *host*`] [-p` *port*`] [-d` *REPL_ID*`]` *ROLE SUFFIX_DN* `[`*SUFFIX_DN* `...]`
Enables replication by assigning a role to an existing suffix.

`dsconf enable-repl-agmt [-h` *host*`] [-p` *port*`]` *SUFFIX_DN HOST*`:`*PORT* `[`*HOST*`:`*PORT* `...]`
Enables replication with another Directory Server.

`dsconf export [-h` *host*`] [-p` *port*`] [-aQ] [-f` *FLAG*`] ... [[-s` *DN*`] ... | [-x` *DN*`] ...]` *SUFFIX_DN* `[`*SUFFIX_DN*`...]` *LDIF_FILE*
Exports suffix data to LDIF format.

`dsconf get-index-prop [-h` *host*`] [-p` *port*`] [-T]` *SUFFIX_DN ATTR_NAME* `[`*PROP* `...]`
Displays the value of an index configuration property.

`dsconf get-log-prop [-h` *host*`] [-p` *port*`] [-T] [-Z` *UNIT*`]` *LOG_TYPE* `[`*PROP* `...]`
Displays server log property values.

`dsconf get-plugin-prop [-h` *host*`] [-p` *port*`] [-T]` *PLUGIN_NAME* `[`*PROP* `...]`
Displays plugin property values.

`dsconf get-repl-agmt-prop [-h` *host*`] [-p` *port*`] [-T]` *SUFFIX_DN HOST*`:`*PORT* `[`*PROP* `...]`
Displays replication agreement property values.

`dsconf get-server-prop [-h` *host*`] [-p` *port*`] [-T] [-M` *UNIT*`] [-Z` *UNIT*`] [`*PROP* `...]`
Displays server property values.

`dsconf get-suffix-prop [-h` *host*`] [-p` *port*`] [-T] [-M` *UNIT*`] [-Z` *UNIT*`]` *SUFFIX_DN* `[`*PROP* `...]`
Displays suffix property values.

`dsconf help-properties [-r]`
Lists properties exposed by subcommands.

`dsconf import [-h` *host*`] [-p` *port*`] [-aK] [-f` *FLAG*`=`*VAL*`] ... [-x` *DN*`] ...` *LDIF_FILE* `[`*LDIF_FILE* `...]` *SUFFIX_DN*
Populates existing suffixes with LDIF data.

`dsconf info`
Displays information about server configuration such as port number, suffix name, server mode and task states.

`dsconf init-repl-dest [-h` *host*`] [-p` *port*`] [-a]` *SUFFIX_DN HOST*`:`*PORT* `[`*HOST*`:`*PORT* `...]`
Launches a total update of the remote replica from a local suffix.

`dsconf list-encrypted-attrs [-h` *host*`] [-p` *port*`] [-E] [-v] [`*SUFFIX_DN* `...]`
Lists encrypted attributes. When used with `-v`, this command displays additional information related to encrypted attributes.

`dsconf list-indexes [-h` *host*`] [-p` *port*`] [-E] [-v] [`*SUFFIX_DN* `...]`
Lists indexed attribute configuration. When used with `-v`, this command displays additional information related to indexes.

`dsconf list-plugins [-h` *host*`] [-p` *port*`] [-E] [-v]`
Lists plugins. When used with `-v`, this command displays additional information related to plugins.

`dsconf list-repl-agmts [-h` *host*`] [-p` *port*`] [-E] [-v] [`*SUFFIX_DN* `...]`
Lists replication agreements. When used with `-v`, this command displays additional information related to replication agreements.

`dsconf list-repl-priorities [-h` *host*`] [-p` *port*`] [-E] [-v] [`*SUFFIX_DN* `...]`
Lists prioritized replication rules. When used with `-v`, this command displays additional information related to prioritized replication rules.

`dsconf list-suffixes [-h` *host*`] [-p` *port*`] [-E] [-v]`
Lists suffixes. When used with `-v`, this command displays additional information related to suffixes. This includes the number of entries, the suffix role and the number of replication agreements, replication priority rules, indexes and encrypted attributes.

`dsconf promote-repl [-h` *host*`] [-p` *port*`] [-d` *REPL_ID*`]` *SUFFIX_DN* `[`*SUFFIX_DN* `...]`
Promotes the role of an existing replicated suffix. A consumer is promoted to a hub, a hub is promoted to a master. To promote a consumer to a master, run the command twice.

`dsconf pwd-compat [-h` *host*`] [-p` *port*`] [-a]` *NEW_MODE*
Changes Directory Server password compatibility state.

`dsconf reindex [-h` *host*`] [-p` *port*`] [-a] [-t` *ATTR*`] ...` *SUFFIX_DN* `[`*SUFFIX_DN* `...]`
Rebuilds index(es) of an existing suffix.

`dsconf restore [-h` *host*`] [-p` *port*`] [-a]` *ARCHIVE_DIR*
Restores Directory Server data from backup archive.

`dsconf rotate-log-now [-h` *host*`] [-p` *port*`] [-a]` *LOG_TYPE*
Closes and renames current log and creates fresh log.

`dsconf set-index-prop [-h` *host*`] [-p` *port*`]` *SUFFIX_DN ATTR_NAME PROP*:*VAL* `[`*PROP*:*VAL* `...]`
Sets the index property value.

For multi-valued properties, use *PROP+*:*VAL* to add a value, and *PROP-*:*VAL* to remove a value.

`dsconf set-log-prop [-h` *host*`] [-p` *port*`]` *LOG_TYPE PROP*:*VAL* `[`*PROP*:*VAL* `...]`
Sets server log property value.

For multi-valued properties, use *PROP+*:*VAL* to add a value, and *PROP-*:*VAL* to remove a value.

`dsconf set-plugin-prop [-h` *host*`] [-p` *port*`]` *PLUGIN_NAME PROP*:*VAL* `[`*PROP*:*VAL* `...]`
Sets plugin property value.

For multi-valued properties, use *PROP+*:*VAL* to add a value, and *PROP-*:*VAL* to remove a value.

`dsconf set-repl-agmt-prop [-h` *host*`] [-p` *port*`]` *SUFFIX_DN HOST*:*PORT PROP*:*VAL* `[`*PROP*:*VAL* `...]`
Sets replication agreement property value.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dsconf set-server-prop [-h` *host*`] [-p` *port*`]` *PROP*:*VAL* `[`*PROP*:*VAL* `...]`

Sets server property value.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dsconf set-suffix-prop [-h` *host*`] [-p` *port*`]` *SUFFIX_DN* *PROP*:*VAL* `[`*PROP*:*VAL* `...]`

Sets suffix property value.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

`dsconf show-repl-agmt-status [-h` *host*`] [-p` *port*`] [-I` *dest-bind-dn*`] [-W` *dest-pw-file*`]` *SUFFIX_DN* *HOST*:*PORT*

Displays a comparison of a source and destination suffix configuration and the status of the replication agreement. When used with v, this command displays additional replication agreement information such as pending changes and delayed maximum duration.

`dsconf show-task-status [-h` *host*`] [-p` *port*`]`

Displays status of current directory server tasks. When used with v, this command displays additional information related to the task type.

`dsconf update-repl-dest-now [-h` *host*`] [-p` *port*`]` *SUFFIX_DN* *HOST*:*PORT* `[`*HOST*:*PORT* `...]`

Restarts replication updates after the destination server has been down by forcing updates to the remote replica from the local suffix.

Global Options The following options are global, and are applicable to all commands and subcommands.

`-?`
`--help` Displays help information for a command or subcommand.

`-c`
`--accept-cert` Does not ask for confirmation before accepting non-trusted server certificates.

`-D` *USER_DN*
`--user-dn` *USER_DN* Binds as *USER_DN*. `dsconf` searches for a *USER_DN* value in the following order: First a a *USER_DN* specified in the command line, then a *USER_DN* set by using the environment variable $LDAP_ADMIN_USER. If none of these are found, the default is to bind as the user `cn=Directory Manager`.

`-e`

`--unsecured`

Connects over LDAP with no secure connection. To connect over a clear connection by default, set the `DIRSERV_UNSECURED` environment variable.

`-h` *HOST*

`--hostname` *HOST*

Connects to the directory on *HOST*. dsconf contacts the LDAP server on the specified host, which may be a host name or an IP address. `dsconf` searches for a *HOST* value in the following order: First a *HOST* specified on the command line, then a *HOST* set by using the environment variable $DIRSERV_HOST. If none of these are found, the default is to use the local host.

For example, when mapping the IPv4 address `192.168.0.99` to IPv6, specify the HOST:PORT as follows: `::ffff:192.168.0.99`.

`-i`

`--no-inter`

Does not prompt for confirmation before performing the operation.

`-j`

`--reject-cert`

Does not ask for confirmation before rejecting non-trusted server certificates (for current session only).

`-p` *PORT*

`--port` *PORT*

Connects to directory on *PORT*. `dsconf` searches for a *PORT* value in the following order: First a*PORT* specified in the command line, then a *PORT* set by using the environment variable $DIRSERV_PORT. If none of these are found, the default is to use port 389.

This option is mutually exclusive with `-P`,`--secure-port`.

`-P` *PORT*

`--secure-port` *PORT*

Connects over SSL to the directory on *PORT*. The `dpconf` command searches for a *PORT* value in the following order:

- A *PORT* specified in the command line
- A *PORT* set by using the `$DIR_SERV_PORT` environment variable

If none of these are found, the default is to use port `636`.

This option is mutually exclusive with `-p`,`--port`.

`-v`

`--verbose`

Displays extra information.

`-V --version`
Displays the current version of `dsconf`. The version is provided in the format *year.day.time*. So version number `2006.178.0035` was built on the 178th day of 2006 at 00h35. If the components used by `dsconf` are not aligned, the version of each individual component is displayed.

`-w` *FILE*
`--pwd-file` *FILE*
Binds using an LDAP password is read from *FILE*. `dsconf` searches for a password *FILE* value in the following order:
A password or password file specified in the command line.
A password file set by using the environment variable $LDAP_ADMIN_PWF.
If none of these are found, the default is to prompt for the password.

Subcommand Options

The following options are applicable to the subcommands where they are specified.

`-A` *PROTOCOL*
`--auth-protocol` *PROTOCOL*
Sets authentication protocol for replication agreements to *PROTOCOL*. For the `create-repl-dest` subcommand, the default value is clear. Other possible values are `ssl-simple` and `ssl-client`. For the `change-repl-dest` subcommand, the default value is the same as that of the HOST:PORT to which you are changing.

`-a`
`--async`
Launches a task and returns the command line accessible immediately.

`-B` *NAME*
`--db-name` *NAME*
Specifies a database name.

`-d` *REPL_ID*
`--repl-id` *REPL_ID*
Specifies a replication ID for a master. It is only used when *ROLE* = master.

`--desc` *DESC*
Specifies a description *DESC*.

`-E`
`--record`
Modifies the display output to show one property value per line.

`-F` *INIT_FUNC*
`--init-func` *INIT_FUNC*
Sets initialization function for a plugin to *INIT_FUNC*.

`-f` *FLAG* or `-f` *FLAG=VAL*
`--flags` *FLAG* or `--flags` *FLAG=VAL*

Customizes imported or exported LDIF.

Import flags:

chunk-size=*INTEGER*

Sets the merge chunk size. Overrides the detection of when to start a new pass during import.

incremental-output

Specifies whether an output file will be generated for later use in importing to large replicated suffixes. Default is `yes`. Possible values are `yes` and `no`. This flag can only be used when the `-K` option is used. If this flag is not used, an output file will automatically be generated.

incremental-output-file=*PATH*

Sets the path of the generated output file for an incremental (appended) import. The output file is used for updating a replication topology. It is an LDIF file containing the difference between the replicated suffix and the LDIF file, and replication information.

Export flags:

multiple-output-file	Exports each suffix to a separate file.
use-main-db-file	Exports the main database file only.
not-export-unique-id	Does not export unique id values.
output-not-folded	Does not wrap long lines.
not-print-entry-ids	Does not export entry IDs.

`-G` *ARG*
`--arguments` *ARG*

Sets plugin argument property to *ARG*.

`-H` *LIB_PATH*
`--lib-path` *LIB_PATH*

Sets plugin library path to *LIB_PATH*.

`-I` *USER_DN*
`--dest-bind-dn` *USER_DN*

Binds as *USER_DN* on destination suffix (Default: same as the DN used for source suffix)

`-K`
`--incremental`

Specifies that the contents of the imported LDIF file are appended to the existing LDAP entries. If this option is not specified, the contents of the imported file replace the existing entries.

`-L` *FILE*
`--db-path` *FILE*
Specifies database directory and path.

`-M` *UNIT*
`--unit-time` *UNIT*
Displays time in *UNIT*, where *UNIT* is one of: w, d, h, m, s (week, day, hour, minute, second).

`-N`
`--no-top-entry`
Does not create a top entry for the suffix. By default, a top-level entry is created when a new suffix is created (on the condition that the suffix starts with `dc=`, `c=`, `o=` or `ou=`). This option changes the default behavior.

`-Q`
`--no-repl`
Does not export additional data needed for replication.

`-r`
`--attr-map`
Displays help properties and their corresponding attributes in `cn=config`.

`-s` *DN*
`--include` *DN*
Exports all data under specified DN.

`-T`
`--tab`
Displays information in a table format.

`-t` *ATTR*
`--attr` *ATTR*
Reindexes the attribute *ATTR* (Default: All attributes).

`-W` *FILE*
`--dest-pwd-file` *FILE*
Binds on a destination suffix using the password read from *FILE*. The default is the same *FILE* used for the source suffix.

`-x` *DN*
`--exclude` *DN*
Does not import or export data contained under the specified DN.

`-Y` *TYPE*
`--type` *TYPE*
Sets plugin type to *TYPE*, where *TYPE* is one of: database, extendedop, preoperation, postoperation, matchingrule, syntax, internalpreoperation, internalpostoperation,

object, pwdstoragescheme, reverpwdstoragescheme, ldbmentryfetchstore, beprecommit, archive2ldbm.

`-Z` *UNIT*
`--unit-size` *UNIT*
Displays memory size data in *UNIT*, where *UNIT* is one of: G, M, k, b (Gigabyte, Megabyte, kilobyte, byte).

Operands The following operands are supported:

ARCHIVE_DIR Directory Server instance backup archive directory.

ATTR_NAME Attribute name.

ENCRYPTION_ALGO Algorithm to use for encryption. Possible values are: `des`, `des3`, `rc2`, `rc4`. These values signify respectively DES block cipher, Triple DES block cipher, RC2 block cipher, RC4 stream cipher.

HOST:*PORT* Destination replicated suffix, defined by *HOST* and destination *PORT*.

LDIF_FILE Path and filename for file in LDIF format.

LOG_TYPE Type of log, where *LOG_TYPE* is one of: access, error, audit.

NEW_MODE Desired mode for password compatibility policy. The default mode is `DS5–compatible-mode`. You can change it to `to-DS6-migration-mode` and then to`to-DS6-mode`.

PLUGIN_NAME Plugin name. The plugin name is defined when the plugin is created.

PRIORITY_NAME Name used to define or identify a prioritized replication rule.

PROP Property name. For a list of PROP names and default values, use the command `dsconf help-properties -v`.

PROP:*VAL* Property and corresponding value. For a list of PROP names and default values, use the command `dsconf help-properties -v`.

For multi-valued properties, use *PROP*+:*VAL* to add a value, and *PROP*-:*VAL* to remove a value.

Multi-valued properties are identified by the `M` keyword. For a list of multi-valued properties, use the command `dsconf help-properties | grep " M "`

Allowed values that are too wide for the help-properties output are listed below:

LOG level (Access): acc-internal | default | acc-default_plus_referrals | acc-timing. For definitions of log levels, see the man page log(5dsconf).

LOG level (Error): default | err-function-calls | err-search-args | err-connection | err-packets | err-search-filter | err-config-file | err-acl | err-ldbm | err-entry-parsing | err-housekeeping | err-replication | err-entry-cache | err-plugins | err-dsml | err-dsml-advanced. For definitions of log levels, see the man page log(5dsconf).

PLG type and depends-on-type: database | extendedop | preoperation | postoperation | matchingrule | syntax | internalpreoperation | internalpostoperation | object | pwdstoragescheme | reverpwdstoragescheme | ldbmentryfetchstore | beprecommit | archive2ldbm

RAG transport-compression: no-compression | default-compression | best-speed | best-compression

SER dsml-client-auth-mode: client-cert-first | http-basic-only | client-cert-only

ROLE
Role of the replicated suffix , where *ROLE* is one of: master, hub, consumer.

SUFFIX_DN
Suffix DN (Distinguished Name)

Description

Syntax values shown in lower case or partly in lower case are literal values.

Those shown in upper case are syntax types, defined as follows:

`ATTR_NAME`
A valid attribute type name such as `cn` or `objectClass`.

`BOOLEAN`
`true` or `false`.

`DN`
A valid distinguished name such as `ou=People,dc=example,dc=com`.

`DURATION`
A duration specified in months (`M`), weeks (`w`), days (`d`), hours (`h`), minutes (`m`), seconds (`s`), and miliseconds (`ms`), or some combination with multiple specifiers. For example, you can specify one week as `1w`, `7d`, `168h`, `10080m`, or `604800s`. You can also specify one week as `1w0d0h0m0s`.

DURATION properties typically do not each support all duration specifiers (Mwdhms). Examine the output of dsconf help-properties for the property to determine which duration specifiers are supported.

EMAIL_ADDRESS
A valid e-mail address.

HOST_NAME
An IP address or host name.

INTEGER
A positive integer value between 0 and the maximum supported integer value in the system address space. On 32-bit systems, 2147483647. On 64-bit systems, 9223372036854775807.

INTERVAL
An interval value of the form *hhmm*-*hhmm 0123456*, where the first element specifies the starting hour, the next element the finishing hour in 24-hour time format, from 0000-2359, and the second specifies days, starting with Sunday (0) to Saturday (6).

IP_RANGE
An IP address or range of address in one of the following formats:

- IP address in dotted decimal form.
- IP address and bits, in the form of network number/mask bits.
- IP address and quad, in the form of a pair of dotted decimal quads.
- All address. A catch-all for clients that are note placed into other, higher priority groups.
- 0.0.0.0. This address is for groups to which initial membership is not considered. For example, for groups that clients switch to after their initial bind.
- IP address of the local host.

LDAP_URL
A valid LDAP URL as specified by RFC 2255 (http://www.ietf.org/rfc/rfc2255.txt).

MEMORY_SIZE
A memory size specified in gigabytes (G), megabytes (M),kilobytes (k), or bytes (b). Unlike DURATION properties, MEMORY_SIZE properties cannot combine multiple specifiers. However, MEMORY_SIZE properties allow decimal values, for example, 1.5M.

NAME
A valid cn (common name).

OCTAL_MODE
A three-digit, octal file permissions specifier. The first digit specifies permissions for the server user ID, the second for the server group ID, the last for other users. Each digit consists of a bitmask defining read (4), write (2), execute (1), or no access (0) permissions, thus 640 specifies read-write access for the server user, read-only access for other users of the server group, and no access for other users.

`PASSWORD_FILE`
The full path to the file from which the bind password should be read.

`PATH`
A valid, absolute file system path.

`STRING`
A DirectoryString value, as specified by RFC 2252 (`http://www.ietf.org/rfc/rfc2252.txt`).

`SUPPORTED_SSL_CIPHER`
An SSL cipher supported by the server. See the Reference for a list of supported ciphers.

`SUPPORTED_SSL_PROTOCOL`
An SSL protocol supported by the server. See the Reference for a list of supported protocols.

`TIME`
A time of the form *hhmm* in 24-hour format, where *hh* stands for hours and *mm* stands for minutes.

Exit Status

The following exit status values are returned:

`0`	Successful completion.
non-zero	An error occurred.

Examples

The following examples show how the `dsconf` command is used.

EXAMPLE 1 Create a Suffix

```
$ dsconf create-suffix -h host -p port
dc=example,dc=com
```

In this example, non-default ports are specified.

Check to see if the suffix has been created.

```
$ dsconf list-suffixes -h host -p port -v
```

EXAMPLE 2 Import LDIF Data into the Suffix

```
$ dsconf import -h host -p port
/local/ds/ldif/example.ldif dc=example,dc=com
```

EXAMPLE 3 Index an Attribute

In this example, the `preferredLanguage` attribute is going to be indexed.

1. Create an index entry for the attribute. By default, the index matching types are equity and presence.

EXAMPLE 3 Index an Attribute *(Continued)*

```
$ dsconf create-index -h host -p port
dc=example,dc=com preferredLanguage
```

2. Check that the index entry has been created

```
$ dsconf get-index-prop -h host -p port
dc=example,dc=com preferredLanguage
```

3. Generate the index for the attribute.

```
$ dsconf reindex -h host -p port
-t preferredLanguage dc=example,dc=com
```

EXAMPLE 4 Back Up the Directory Server Data

```
$ dsconf backup -h host -p port
/tmp/backupArchiveDir
```

For complete backup procedures, see the *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

EXAMPLE 5 Monitor and Change Cache Size for a Suffix

1. Search for the string `cache` within the `dsconf` help properties:

```
$ dsconf help-properties | grep cache
```

2. Determine which property is most applicable and request more information. In the results of the preceding step, `cache-mem-size` seems to correspond. For additional information, use the verbose option:

```
$ dsconf help-properties -v | grep entry-cache-size
SUF  entry-cache-size  rw MEMORY_SIZE (Ex: 3G,2m,200k,10000b)
nsslapd-cachememsize
Cache size in term of memory space: (Default: 10M)
```

Use the following information to interpret the results above:

`SUF`	This property applies to a suffix.
`entry-cache-size`	The name of the property
`rw`	You have read and write access to the property when using `get-suffix-prop` and `set-suffix-prop`.
`MEMORY_SIZE`	Use memory size values as described in this man page.
`nsslapd-cachememsize`	The attribute under `cn=config` to which this property applies.
`(Default: 10M)`	The default value of this property

EXAMPLE 5 Monitor and Change Cache Size for a Suffix *(Continued)*

3. Determine the current value of entry-cache-size:

   ```
   $ dsconf get-suffix-prop -h host -p port
   dc=example,dc=com entry-cache-size
   entry-cache-size : 10M
   ```

4. Change the value of entry-cache-size to 12M:

   ```
   $ dsconf set-suffix-prop -h host -p port
   dc=example,dc=com entry-cache-size:12M
   ```

5. Check that the value has been changed:

   ```
   $ dsconf get-suffix-prop -h host -p port
   dc=example,dc=com entry-cache-size
   entry-cache-size : 12M
   ```

EXAMPLE 6 Export to LDIF While Using Filters

```
$ dsconf export -h host -p port
-f not-print-entry-ids -s ou=people,dc=example,dc=com
 -s ou=contractors,dc=example,dc=com dc=example,dc=com
/local/ds/ldif/export.ldif
```

This example shows a command that:

- Uses the flag `not-print-entry-ids` to request that entry IDs are not exported.
- Exports data from two suffixes `ou=people,dc=example,dc=com` and `ou=contractors,dc=example,dc=com` into one LDIF file `/local/ds/ldif/export.ldif`.

EXAMPLE 7 Rotate the Access Log and Modify the Rotation Delay for the Access Log

If you have a log which is getting very large, you can rotate the log. Rotation backs up the existing log file and creates a fresh log file. In this example, the access log is rotated.

1. Rotate the access log by using the command:

   ```
   $ dsconf rotate-log-now -h host -p port access
   ```

2. You can now modify the delay between log rotations for the access log.

   Find the property which sets maximum log size:

   ```
   $ dsconf help-properties -v | grep LOG
   ```

   The output from the previous command shows that the required property is `rotation-interval`.

3. To see the default setting for `rotation-interval`:

EXAMPLE 7 Rotate the Access Log and Modify the Rotation Delay for the Access Log *(Continued)*

```
$ dsconf get-log-prop -h host -p port
access rotation-interval
```

The default is one day `1d`.

4. To increase the rotation delay to two days, use the command:

```
$ dsconf set-log-prop -h host -p port
access rotation-interval:2d
```

EXAMPLE 8 Configure Replication in a Two-Master Topology

This procedure configures replication on a topology with two severs, and both are masters. Replication is configured first on one master, then on the second master. Master 1 is located on `server1.example:1389`. Master 2 is located on `server2.example:2389`.

1. On server 1: Create a suffix

```
$ dsconf create-suffix -h server1.example -p 1389
dc=example,dc=com
```

2. On Server 1: Populate the suffix with LDIF data

```
$ dsconf import -a -h server1.example -p 1389
/opt/SUNWdsee/ds6/ldif/Example.ldif dc=example,dc=com
```

If the import takes a long time, you can obtain status on the import operation using:

```
$ dsconf info -h server1.example -p 1389
```

or

```
$ dsconf show-task-status -h server1.example -p 1389 -v
```

Alternatively, you can view the status of the task while it is running by omitting the `-a` option in the command.

3. On Server 1: Enable replication on Master 1. This step assigns a replication role and ID to an existing suffix. It also sets the replication manager bind DN to the default replication manager DN.

```
$ dsconf enable-repl -h server1.example -p 1389
-d 1 master dc=example,dc=com
```

4. On server 2: Create a suffix

```
$ dsconf create-suffix -h server2.example -p 2389 dc=example,dc=com
```

EXAMPLE 8 Configure Replication in a Two-Master Topology *(Continued)*

5. On Server 2: Enable replication on Master 2. This step assigns a replication role and ID to an existing suffix. It also sets the replication manager bind DN to the default replication manager DN.

   ```
   $ dsconf enable-repl -h server2.example -p 2389
   -d 2 master dc=example,dc=com
   ```

6. On Server 1: Create a replication agreement from Master 1 to Master 2.

   ```
   $ dsconf create-repl-agmt -h server1.example -p 1389
   dc=example,dc=com server2.example:2389
   ```

7. On Server 2: Create a replication agreement from Master 2 to Master 1

   ```
   $ dsconf create-repl-agmt -h server2.example -p 2389
   dc=example,dc=com server1.example:1389
   ```

8. On Server 1: Check that the replication agreement status is OK.

   ```
   $ dsconf show-repl-agmt-status -h server1.example -p 1389
   dc=example,dc=com server2.example:2389
   ```

   If the status is not OK, then accord the replication agreement.

   ```
   $ dsconf accord-repl-agmt -h server1.example -p 1389
   dc=example,dc=com server2.example:2389
   ```

9. On Server 1: From Master 1, initialize replication on Master 2. This step initializes Master 2 with the data contained in the suffix on Master 1 and starts replication.

   ```
   $ dsconf init-repl-dest -h server1.example -p 1389
   dc=example,dc=com server2.example:2389
   ```

The replication agreements in both directions are now active and replication is running.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Evolving

See Also

dsadm(1M)

Name dsee_deploy – deploy Directory Server Enterprise Edition software

Synopsis

```
./dsee_deploy install -c component -i install_path
 [OPTIONS]
```

```
install-path/dsee6/bin/dsee_deploy
 uninstall -c component -i install_path [OPTIONS]
```

Description The `dsee_deploy` command installs Directory Server Enterprise Edition software from zip distributions rather than native packages, and registers server software with the Cacao common agent container to allow remote administration. The `dsee_deploy` command also removes registration information from the Cacao common agent container, and removes Directory Server Enterprise Edition software installed from the zip distribution.

Software installed from a zip distribution does not require that you have super user or administrator access to the system. The software is self-contained and need not have dependencies outside the install path you choose.

Subcommands The following subcommands are supported:

`install` Install component software.

Use the command unpacked with the product distribution.

`uninstall` Remove component software.

Use the command placed under *install-path*`/dsee6/bin/` by the `install` subcommand.

Options The following options are supported:

`-c` *component*
`--component` *component*

Install or remove the specified combination of Directory Server Enterprise Edition component products. The *component* may be one of the following values. The default value is `ALL`.

`ALL` Install or remove Directory Proxy Server and Directory Server software, including server administration, and LDAP client command-line tools, and Directory Server Resource Kit software.

`DPS` Install or remove Directory Proxy Server software, including command-line administration tools.

`DPSCONFIG` Install or remove Directory Proxy Server command-line administration tools.

`DPSSERVER` Install or remove Directory Proxy Server software.

`DS` Install or remove Directory Server software, including server administration and LDAP client command-line tools, and Directory Server Resource Kit software.

`DSCONFIG` Install or remove Directory Server command-line administration tools.

`DSRK` Install or remove Directory Server Resource Kit software, including LDAP client command-line tools.

`DSSERVER` Install or remove Directory Server software.

`LDAPTOOLS` Install or remove LDAP client command-line tools.

`-h`
`--help`
Display the usage message for the command.

`-I`
`--no-inter`
Install in non-interactive mode, accepting the license text without confirmation. This mode is useful for silent installation.

`-i` *install_path*
`--install-path` *install_path*
Install or remove Directory Server Enterprise Edition software under the specified file system directory.

If the specified file system directory does not exist at installation time, the `dsee_deploy` command attempts to create it.

`-N`
`--no-cacao`
Do not use or configure the Cacao common agent container.

If specified, you may use the `dsconf`(1M) command to manage Directory Server and the `dpconf`(1M) command to manage Directory Proxy Server, but not Directory Service Control Center.

`-O`
`--non-overwrite`
Never overwrite files during installation.

`-p` *cacao_port*
`--cacao-port` *cacao_port*

Configure the Cacao common agent container used for remote management to listen for JMX management communications on the specified port number.

If specified, the port must not be in use.

If no Cacao common agent contain port is specified, the default value is `11162`.

`-v`
`--verbose`

Display extra messages during software installation and removal.

Exit Status The following exit values are returned:

`0` Successful completion.

`1` The `unzip` command could not be found.

`2` The *install_path* file system directory could not be created.

`3` The *install_path* is not a file system directory.

`4` Permission was denied to create the *install_path* file system directory.

`5` A *component_product* name was not recognized.

`6` The specified *cacao_port* could not be used.

`7` There was an internal memory error.

`8` The `unzip` command returned an error.

`9` The server(s) installed could not be registered with the Cacao common agent container.

`10` A required zip file, normally located in the `dsee_data/` file system directory next to the `dsee_deploy` command, could not be found.

`11` The `cacaoadm` command issued to configure the Cacao common agent container failed.

`12` The number of parameters was invalid.

Make sure you have specified at least all mandatory options.

`13` The `dsee_deploy` command failed to configure the Cacao common agent container.

`14` The `dsee_deploy` command failed to start the Cacao common agent container.

`15` The specified subcommand was not valid.

16 The Cacao common agent container could not be removed.

17 The specified Cacao common agent container port is already in use.

18 An invalid option was specified.

19 An option was incorrectly specified more than once.

20 Permission to the specified file system directory was denied.

21 The `dsee_deploy` command, necessary for uninstallation, could not be copied to under the specified *install_path*.

22 A subcommand was missing. The `dsee_deploy` requires that you specify a subcommand (`install` | `uninstall`).

23 The `-N` option is not for use with the `uninstall` subcommand.

24 The `-O` option is not for use with the `uninstall` subcommand.

25 The `-p` option is not for use with the `uninstall` subcommand.

26 The Cacao common agent container is already configured. Use the `-N` option.

27 The specified component is not installed in the specified location, and therefore cannot be removed.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	Zip distributions only
Stability Level	Evolving

See Also `cacaoadm`(1M), `unzip`(1)

Name dsmig – Migrates a Directory Server Instance

Synopsis

```
install-path/ds6/bin/dsmig
subcommand [options] [operands]
```

Description The `dsmig` command is the migration command for a single Directory Server instance. Use the `dsmig` command with any of the subcommands described in this man page.

`dsmig` migrates a Directory Server 5.1 instance to a Directory Server 6.2 instance.

`dsmig` must be run from the local machine on which the new instance will be located. If the new instance exists, migration subcommands are carried out on that instance. If the new instance does not exist, `dsmig` creates the new instance with the parameters specified in the global options.

Subcommands The following subcommands are supported.

`dsmig info`

Displays information on the status of each migration step.

The format of the subcommand is:

```
dsmig info NEW_INSTANCE_PATH
```

`dsmig migrate-all`

Migrates the old instance to the new instance in a single step. This subcommand essentially combines the functionality of all the other subcommands.

The format of the subcommand is:

```
dsmig migrate-all [-R] [-N] [-c] [-j] [-e | -Z]
 [-D USER_DN] [-w PWD_FILE] [-v] OLD_INSTANCE_PATH
 NEW_INSTANCE_PATH
```

`dsmig migrate-config`

Migrates the configuration from the old instance to the new instance.

The format of the subcommand is:

```
dsmig migrate-config [-R] [-N] [-c] [-j] [-e | -Z]
 [-D USER_DN] [-w PWD_FILE] [-v] OLD_INSTANCE_PATH
 NEW_INSTANCE_PATH
```

`dsmig migrate-data`

Migrates the data from the old instance to the new instance. Migrating the change logs of the old instance is optional. Migration of the NetscapeRoot database must be specified as this database is not migrated by default.

The format of the subcommand is:

```
dsmig migrate-data [-R] [-N] [-v] OLD_INSTANCE_PATH
 NEW_INSTANCE_PATH
```

`dsmig migrate-schema` Migrates the schema from the old instance to the new instance.

The format of the subcommand is:

`dsmig migrate-schema [-v]` *OLD_INSTANCE_PATH NEW_INSTANCE_PATH*

`dsmig migrate-security` Migrates the security files from the old instance to the new instance.

The format of the subcommand is:

`dsmig migrate-security [-v]` *OLD_INSTANCE_PATH NEW_INSTANCE_PATH*

Global Options

The following options are global, and are applicable to all commands and subcommands.

`--?`
`---help` Displays help information for a command or subcommand.

`--i`
`---no-inter` Does not request confirmation before executing the command.

`--p` *PORT*
`---port` *PORT* The port used for LDAP traffic. The default LDAP port is 389 or 1389.

`--P` *SSL_PORT*
`---secure-port` *SSL_PORT* The port used for secure LDAP traffic. The default secure LDAP port is 636 or 1636.

Subcommand Options

The following options are applicable to the subcommands where they are specified.

`--c`
`---accept-cert` Specifies that confirmation should not be requested before accepting non-trusted server certificates.

`-D` *USER_DN*
`--user-dn` *USER_DN* Defines the Directory Manager DN. The default is `cn=Directory Manager`.

`--e`
`---unsecured` Specifies an unsecured connection over LDAP. If this option is not used, a secure LDAP connection using StartTLS is made by default.

`--j`
`---reject-cert` Specifies that confirmation should not be requested before rejecting non-trusted server certificates (for this session only.)

`--N`
`---netscapeRoot` Specifies that data for the “o=netscapeRoot“ suffix must be migrated. If this option is used with the `migrate-config` subcommand, it refers to the suffix configuration data. If this option is used with the `migrate-data` subcommand, it refers to the netscapeRoot database. Using the option with the `migrate-all` subcommand means that neither the configuration data nor the database is migrated.

`--R`
`---replication` Specifies that replication data should be migrated. If this option is used with the `migrate-config` subcommand, it refers to replication configuration data. If this option is used with the `migrate-data` subcommand, it refers to replication changelogs. Using the option with the `migrate-all` subcommand means that both replication configuration data and changelogs are migrated.

`--v`
`---verbose` Specifies that additional messages are displayed.

`--w` *PWD_FILE*
`---pwd-file` *PWD_FILE* The file from which the Directory Manager password should be read. If this option is not specified, the command prompts for the password.

`-Z`
`--secured` Specifies an SSL connection over LDAP.

Subcommand Operands

The following operands are applicable to the subcommands where they are specified.

`-OLD_INSTANCE_PATH` Specifies the path to the 5.1 instance.

`-NEW_INSTANCE_PATH` Specifies the path to the 6.0 instance.

Exit Status

The following exit status values are returned:

`0` Successful completion.

non-zero An error occurred.

Examples

The following examples show how the `dsmig` command is used.

EXAMPLE 1 Migrating the schema

```
$ dsmig migrate-schema -p 6389 -P 6636
  /local/ds52pX/slapd-old_52_instance /local/new_ds61_instance/
```

EXAMPLE 1 Migrating the schema *(Continued)*

This command migrates the schema from the old Directory Server instance to the new 6.0 instance.

In this example, the LDAP port is specified as 6389, and the secure port is specified as 6636. If you do not specify port numbers, the default port numbers 389 and 636 (for root user) or 1389 and 1636 (for not-root user) are used. If you do not specify port numbers and the default port numbers are already being used, the `dsmig` command aborts.

EXAMPLE 2 Migrating the configuration

```
$ dsmig migrate-config -N /local/ds52pX/slapd-old_52_instance
 /local/new_ds61_instance/
```

This command migrates the configuration from the old Directory Server instance to the new instance.

In this example, configuration data for the “o=netscapeRoot“ suffix and replication configuration data are migrated.

EXAMPLE 3 Migrating the data

```
$ dsmig migrate-data -R -N /local/ds52pX/slapd-old_52_instance
/local/new_ds61_instance/
```

This command migrates the data from the old Directory Server instance to the new instance.

In this example, the replication change logs are not migrated. The NetscapeRoot database is migrated.

EXAMPLE 4 Migrating everything in a single step

```
$ dsmig migrate-all -R -N /local/ds52pX/slapd-old_52_instance
/local/new_ds61_instance/
```

In this example, replication configuration data is not migrated. Data for the “o=netscapeRoot“ suffix is migrated.

EXAMPLE 5 Obtaining migration status information

```
$ dsmig info /local/new_ds61_instance/

Old instance path   :  /local/ds52pX/slapd-old_52_instance
New instance path   :  /local/new_ds61_instance
```

EXAMPLE 5 Obtaining migration status information *(Continued)*

```
Schema Migration    :  Completed
Security Migration  :  Not completed
Config Migration    :  Completed except NetscapeRoot and Replication configuration
Data Migration      :  Not completed
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Evolving

See Also

dsconf(1M)

Name dsrepair – repair replicated directory entries

Synopsis *install-path*`/ds6/support_tools/bin/dsrepair`
subcommand `[`*options*`]` *arguments*

Description The `dsrepair` command makes it possible to repair entries that prevent replication from preceeding normally. You must enable the replication repair plug-in to use the `dsrepair` command.

Use the `dsrepair` command only under the supervision of qualified support personnel.

The `dsrepair` command functions only in non-secure mode, with simple authentication.

The `dsrepair` command is not available on Windows systems, though it can be run against a Directory Server instance on a Windows system.

Subcommands The following subcommands are supported:

`dsrepair add-entry [`*options*`]` *suffix entry.ldif*
Adds the entry specified in the *entry.ldif* file to the specified *suffix*.

If an entry or tombstone entry having the same DN or `nsUniqueID` already exists, or if the parent entry does not exist, `add-entry` fails.

`dsrepair begin-repair-mode [`*options*`]` *suffix*
Puts the specified *suffix* in repair mode such that the only modify operations allowed are those performed using the `dsrepair` command.

Read operations continue normally while the suffix is in repair mode.

`dsrepair delete-entry [`*options*`]` *suffix entry.ldif*
Deletes the entry specified in the *entry.ldif* file from the specified *suffix*, and any tombstone associated with the entry.

If no entry or tombstone entry having the same DN or `nsUniqueID` already exists, or the specified entry has child entries, `delete-entry` fails.

`dsrepair end-repair-mode [`*options*`]` *suffix*
Returns the specified *suffix* from repair mode to its normal replication mode.

`dsrepair replace-entry [`*options*`]` *suffix entry.ldif*
Replaces an entry in the directory with the content specified in the *entry.ldif* file.

If no entry having the DN or `nsUniqueID` exists, or the entries returned for based on the DN and `nsUniqueID` are different, `replace-entry` fails.

`dsrepair update-ruv [`*options*`]` *suffix csn*
Replaces the maximum change sequence number (CSN) in a replication update vector (RUV) element with the specified *csn* string.

Options The following options are supported:

`-D` *bindDN*
`--bind-dn` *bindDN*
: Use the specified bind DN to authenticate to the directory server.

  The default is `cn=Directory Manager`.

`-h` *host*
`--hostname` *host*
: Contact the LDAP server on the specified host, which may be a host name or an IP address.

  For example, when mapping the IPv4 address `192.168.0.99` to IPv6, pass the `-h` option with its argument as `-h ::ffff:192.168.0.99`.

  The default is `localhost`.

`-p` *port*
`--port` *port*
: Contact the LDAP server on the specified port.

  The default is `389`.

`-w` *file*
`--pwd-file` *file*
: Use the bind password in the specified *file*.

  If this option is not specified, the `dsrepair` command prompts for the password.

Exit Status

The following exit values are returned:

`0`	Successful completion.
non-zero	An error occurred.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Evolving

Name idsktune – generate system tuning recommendations for running Directory Server Enterprise Edition server software

Synopsis `./idsktune  [-q] [-D] [-v] [-c] [-i` *install-path*`]`

Description The `idsktune` command checks patch levels and kernel parameter settings for the system on which Directory Server or directory client applications run, making tuning recommendations as it performs the checks. Run the command as super user to obtain the widest range of tuning recommendations.

The `idsktune` command is delivered next to the `dsee_deploy` command with zip distribution software only.

The `idsktune` command suggests changes you make to the system, but does not itself make any changes. You must fix at least all `ERROR` conditions identified by the `idsktune` command.

The `idsktune` command reports as missing *all* patches recommended at the time of release and not installed on the system, even patches for packages not installed on the system.

Options The `idsktune` command supports the following options.

`-c` Display tuning recommendations only for directory client applications.

Default is to display recommendations for both directory client applications and for Directory Server.

`-D` Run in debug mode, displaying messages to showing commands the `idsktune` command runs internally, preceded by `DEBUG`.

`-i` *install-path* Check the specified installation directory to ensure enough space is available.

`-q` Run in quiet mode, reporting only information about key system prerequisites and essential settings.

`-v` Display the version information about the build and exit.

Extended Description The `idsktune` command verifies and reports on the following settings depending on the underlying system.

Operating system and kernel versions

- Solaris™ and Red Hat version numbers
- Solaris kernel build date
- Solaris, and HP-UX patches

Memory and disk space

- Physical memory size
- Swap space or swap partition size
- Memory resource limits
- File descriptor resource limits

Scheduler settings

- Maximum threads per process for HP-UX
- Maximum files for HP-UX

TCP settings

Many of the following are system-specific TCP tuning settings.

- Listen backlog queue size
- `tcbhashsize`, `tcbhashnum` and `tcp_msl`
- `sominconn` and `somaxconn`
- `ipport_userreserved_min`
- `tcp_close_wait_interval` and `tcp_time_wait_interval`
- `tcp_keepalive_interval`
- `tcp_max_listen`
- `tcp_conn_request_max`
- `tcp_conn_req_max_q` and `tcp_conn_req_max_q0`
- `tcp_rexmit_interval_initial`
- `net.inet.ip.portrange.hifirst` and `tcp_smallest_anon_port`
- `tcp_slow_start_initial`
- `net.inet.tcp.delayed_ack` and `tcp_deferred_ack_interval`
- `link_speed` on `/dev/hme`

Tuning system settings, especially network stack settings, involves considering potentially not just directory applications and Directory Server, but also other applications running on the system and in the environment. In general, however, implementing the recommendations optimizes directory performance whether the system is dedicated to Directory Server or shared with other applications.

Exit Status

The `idsktune` command exits with status `0` if it completes successfully and no `ERROR`s are found. Otherwise, it exists with non-zero status.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	Zip distribution only
Stability Level	Evolving

Name

ns-accountstatus – show whether an account is active

Synopsis

install-path`/ds6/bin/ns-accountstatus`
`[-D` *rootDN*`] {-w` *password* `| -w - | -j` *filename*`} [-p` *port*`]`
`[-h` *host*`] -I` *accountDN*

Description

The `ns-accountstatus` command shows whether the account corresponding to an entry is active. The command can also be used to show whether the accounts corresponding to a role are active.

Options

The following options are supported:

`-?` Display the usage message.

`-D` *rootDN* Bind using the Directory Manager (directory super user) *rootDN*.

When this option is not specified, the default bind DN, `cn=Directory Manager`, is used.

`-h` *host* Bind to the specified *host* on which the Directory Server instance runs.

Default: `localhost`.

`-I` *accountDN* Determine account status for the entry or role having Distinguished Name *accountDN*.

`-j` *filename* Read the bind password for simple authentication from *filename*.

`-p` *port* Bind to the specified *port* on which the Directory Server instance listens.

Default: `389`.

`-w –` Bind with simple authentication, specifying the password interactively.

`-w` *password* Bind with simple authentication using the specified *password*.

Exit Status

The following exit values are returned:

`0` Successful completion.

`1` An error occurred.

On error, verbose error messages are displayed on standard output.

Examples

The examples in this section use sample data from the `Example-roles.ldif` file.

EXAMPLE 1 Examining Status of an Entry

The following command checks the status of Barbara Jensen's entry.

```
$ ./ns-accountstatus -D "cn=Directory Manager" -j /tmp/pwd.txt \
> -I uid=bjensen,ou=people,dc=example,dc=com
uid=bjensen,ou=people,dc=example,dc=com  activated.
```

EXAMPLE 2 Examining Status of a Role

The following command checks the status of the Directory Administrators role.

```
$ ./ns-accountstatus -D "cn=Directory Manager" -j /tmp/pwd.txt \
> -I "cn=Directory Administrators,dc=example,dc=com"
cn=Directory Administrators,dc=example,dc=com  activated.
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Stable

See Also

`ns-activate`(1M), `ns-inactivate`(1M)

Name

ns-activate – activate accounts

Synopsis

install-path`/ds6/bin/ns-activate`
 `[-D` *rootDN*`] {-w` *password* `|  -w - |  -j` *filename*`} [-p` *port*`]`
 `[-h` *host*`] -I` *accountDN*

Description

The `ns-activate` command activates an account corresponding to an entry. The command can also be used to activate accounts sharing a role.

Options

The following options are supported:

`-?` Display the usage message.

`-D` *rootDN* Bind using the Directory Manager (directory super user) *rootDN*.

When this option is not specified, the default bind DN, `cn=Directory Manager`, is used.

`-h` *host* Bind to the specified *host* on which the Directory Server instance runs.

Default: `localhost`.

`-I` *accountDN* Activate the account for the entry or accounts corresponding to the role having Distinguished Name *accountDN*.

`-j` *filename* Read the bind password for simple authentication from *filename*.

`-p` *port* Bind to the specified *port* on which the Directory Server instance listens.

Default: `389`.

`-w –` Bind with simple authentication, specifying the password interactively.

`-w` *password* Bind with simple authentication using the specified *password*.

Exit Status

The following exit values are returned:

`0` Successful completion.

`1` An error occurred.

On error, verbose error messages are displayed on standard output.

Examples

The examples in this section use sample data from the `Example-roles.ldif` file.

EXAMPLE 1 Activating an Inactive Account Entry

The following command activates Barbara Jensen's account.

```
$ ./ns-activate -D "cn=Directory Manager" -j /tmp/pwd.txt \
> -I uid=bjensen,ou=people,dc=example,dc=com
uid=bjensen,ou=people,dc=example,dc=com activated.
```

EXAMPLE 2 Activating an Inactive Account Role

The following command activates the Directory Administrators role.

```
$ ./ns-activate -D "cn=Directory Manager" -j /tmp/pwd.txt \
> -I "cn=Directory Administrators,dc=example,dc=com"
cn=Directory Administrators,dc=example,dc=com activated.
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Stable

See Also

`ns-accountstatus`(1M), `ns-inactivate`(1M)

Name

ns-inactivate – inactivate accounts

Synopsis

install-path`/ds6/bin/ns-inactivate`
 `[-D` *rootDN*`] {-w` *password* `| -w - | -j` *filename*`} [-p` *port*`]`
 `[-h` *host*`] -I` *accountDN*

Description

The `ns-inactivate` command inactivates an account corresponding to an entry. The command can also be used to inactivate accounts sharing a role.

Options

The following options are supported:

`-?` Display the usage message.

`-D` *rootDN* Bind using the Directory Manager (directory super user) *rootDN*.

When this option is not specified, the default bind DN, `cn=Directory Manager`, is used.

`-h` *host* Bind to the specified *host* on which the Directory Server instance runs.

Default: `localhost`.

`-I` *accountDN* Inactivate the account for the entry or accounts corresponding to the role having Distinguished Name *accountDN*.

`-j` *filename* Read the bind password for simple authentication from *filename*.

`-p` *port* Bind to the specified *port* on which the Directory Server instance listens.

Default: `389`.

`-w` – Bind with simple authentication, specifying the password interactively.

`-w` *password* Bind with simple authentication using the specified *password*.

Exit Status

The following exit values are returned:

`0` Successful completion.

`1` An error occurred.

On error, verbose error messages are displayed on standard output.

Examples

The examples in this section use sample data from the `Example-roles.ldif` file.

EXAMPLE 1 Inactivating an Account Entry

The following command inactivates Barbara Jensen's account.

```
$ ./ns-activate -D "cn=Directory Manager" -j /tmp/pwd.txt \
> -I uid=bjensen,ou=people,dc=example,dc=com
uid=bjensen,ou=people,dc=example,dc=com inactivated.
```

EXAMPLE 2 Inactivating an Account Role

The following command inactivates the Directory Administrators role.

```
$ ./ns-activate -D "cn=Directory Manager" -j /tmp/pwd.txt \
> -I "cn=Directory Administrators,dc=example,dc=com"
cn=Directory Administrators,dc=example,dc=com inactivated.
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Stable

See Also

`ns-accountstatus`(1M), `ns-activate`(1M)

Name replcheck – diagnose and repair some replication failures

Synopsis *install-path*`/ds6/support_tools/bin/replcheck`
subcommand options

Description The `replcheck` command allows you to diagnose and repair a replication halt. Use the `replcheck` command with one of the options described in this man page.

Subcommands The following subcommands are supported:

`replcheck diagnose [-D` *DN*`] [-w` *PW_FILE*`] [-L` *LOG_DIR*`] [-v]` *TOPOLOGY_FILE*
Diagnoses the cause of the replication breakage and summarizes the proposed repair actions.

`replcheck fix [-D` *DN*`] [-w` *PW_FILE*`] [-L` *LOG_DIR*`] [-v]` *TOPOLOGY_FILE*
Fixes the replication breakage.

Global Options The following options are global, and are applicable to all commands and subcommands.

`--?`
`--help` Displays help information for a command or subcommand.

`-V--version` Displays the current version of `replcheck`. The version is provided in the format *year.day.time*. So version number `2006.178.0035` was built on the 178th day of 2006 at 00h35. If the components used by `replcheck` are not aligned, the version of each individual component is displayed.

Subcommand Options The following options are applicable to the subcommands where they are specified.

`-D` *bindDN*
`--bind-dn` *bindDN* Use the specified bind DN to authenticate to the directory server.

The default is `cn=Directory Manager`.

`-L` *dir-path*
`--log-dir` *dir-path* Creates a `replcheck.log` log file in this directory.

If this option is not specified, the `replcheck.log` log file will be created in the home directory.

`-v`
`--verbose` Displays additional information.

`-w` *password-file*
`--pwd-file` *file* Use the bind password in the specified *password-file*.

If this option is not specified, the `replcheck` command prompts for the password.

Operands The following operands are supported:

TOPOLOGY_FILE Specifies the path to the file that describes the replication topology.

This file contains one record for each line in the following format: *hostname*:*port*:*suffix_dn*[:*label*]. The optional *label* field provides a name that appears in any messages that are displayed or logged. If you do not specify a *label*, the *hostname*:*port* are used instead.

For example, the following topology file describes a replication topology consisting of two hosts:

```
host1:389:dc=example,dc=com:Paris
host2:489:dc=example,dc=com:New York
```

Note – The `replcheck` command must access the servers in the topology using their non-secure ports. The topology file can not specify an SSL port.

Extended Description

I

The `replcheck` command diagnoses and repairs a replication halt. The `replcheck diagnose` subcommand compares the RUVs for each of the servers in your replication topology to determine if the masters are synchronized. If the search results show that all of the consumer replica in-memory RUVs are evolving on time or not evolving but equal to those on the supplier replicas, the tool will conclude that a replication halt is not occurring.

However, if the command determines that the consumer RUVs do not change at all over time, then the `replcheck diagnose` subcommand displays the repair operation it would do and exits without making the repair. Then, you can launch the `replcheck fix` subcommand to repair the replication halt. For example, the command determines that replication is blocked on the entry associated with CSN 24 if a supplier has a CSN of 40, while the consumer has a CSN of 23 that does not evolve at all over time.

The `replcheck` command can repair two types of replication halt:

- The entry at which replication is halted, in our previous example CSN 24, exists on the supplier but not on the consumer. The `replcheck` command takes the entry from the instance that is at least more up-to-date than the consumer and then pushes it to the consumer.
- The entry at which replication is halted, CSN 24, is unknown to supplier A. This can occur if a server is reinitialized or a replication agreement is deleted, resulting in a consumer becoming out of date and breaking replication . The `replcheck` command looks at other servers in the topology to see if the CSN is recognized. If it finds the CSN on a new supplier, such as supplier B, it creates a replication agreement with supplier B and lets replication send the entry, CSN 24, to the consumer.

Exit Status

The following exit status values are returned:

0 Successful completion.

non-zero An error occurred.

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Evolving

Name schema_push – ensure manually modified schema are replicated to consumers

Synopsis *install-path*`/ds6/bin/schema_push`
instance-path

Description When schema modifications are made manually by editing the `.ldif` files such as `99user.ldif` directly, the `schema_push` command should be run to update the modification time used by replication. This ensures that the modified schema are replicated to the consumers.

The *instance-path* argument is the path to the instance where you updated schema files, such as /local/ds.

Note – When using the command on Windows systems, you may need to include Perl in your `PATH`, as shown in the following example.

```
C:\ds6\bin>set PATH=%PATH%;C:\dsee6\perl5\bin
C:\ds6\bin>perl schema_push C:\servers\ds\
```

Once the script has been run, you must restart the server to trigger the schema replication.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Stable

See Also dsadm(1M)

REFERENCE

Directory Server Configuration

Name

all-ids-threshold – Maximum number of values per index key in an index list

Description

Syntax	INTEGER or INTEGER | inherited
Default Value	4000 or inherited
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property defines the maximum number of values per index key that the server maintains in an index list. It can be set for an entire server instance, for an entire suffix, and for an individual attribute type. You can also set individual thresholds for equality, presence, and substring indexes.

When you do not set specific threshold values, the values at each level are inherited from the more global values. Thus the default suffix threshold value is inherited from the setting for the server instance; the default attribute type value from the setting for the suffix. In addition to inheritance of default settings, this property handles settings as follows.

`inherited` The threshold value is inherited from the more global setting.

`<2000` The threshold value is rounded up to `2000`.

`>2000` The setting is used as a guaranteed minimum threshold. Because of internal mechanisms, the real value can be slightly more than the specified value.

After you modify this property for an entire server instance or an entire suffix, import all data from LDIF to reinitialize all indexes.

If you modify this property only for a specific attribute, it is usually most expedient to use the `dsconf reindex` command on the attribute for which you changed the threshold. The `dsconf reindex` command runs a directory task to reindex the attribute while the server instance is online.

Description

Syntax values shown in lower case or partly in lower case are literal values.

Those shown in upper case are syntax types, defined as follows:

`ATTR_NAME`
A valid attribute type name such as `cn` or `objectClass`.

`BOOLEAN`
`true` or `false`.

`DN`
A valid distinguished name such as `ou=People,dc=example,dc=com`.

DURATION

A duration specified in months (M), weeks (w), days (d), hours (h), minutes (m), seconds (s), and miliseconds (ms), or some combination with multiple specifiers. For example, you can specify one week as 1w, 7d, 168h, 10080m, or 604800s. You can also specify one week as 1w0d0h0m0s.

DURATION properties typically do not each support all duration specifiers (Mwdhms). Examine the output of dsconf help-properties for the property to determine which duration specifiers are supported.

EMAIL_ADDRESS

A valid e-mail address.

HOST_NAME

An IP address or host name.

INTEGER

A positive integer value between 0 and the maximum supported integer value in the system address space. On 32-bit systems, 2147483647. On 64-bit systems, 9223372036854775807.

INTERVAL

An interval value of the form *hhmm*-*hhmm 0123456*, where the first element specifies the starting hour, the next element the finishing hour in 24-hour time format, from 0000-2359, and the second specifies days, starting with Sunday (0) to Saturday (6).

IP_RANGE

An IP address or range of address in one of the following formats:

- IP address in dotted decimal form.
- IP address and bits, in the form of network number/mask bits.
- IP address and quad, in the form of a pair of dotted decimal quads.
- All address. A catch-all for clients that are note placed into other, higher priority groups.
- 0.0.0.0. This address is for groups to which initial membership is not considered. For example, for groups that clients switch to after their initial bind.
- IP address of the local host.

LDAP_URL

A valid LDAP URL as specified by RFC 2255 (http://www.ietf.org/rfc/rfc2255.txt).

MEMORY_SIZE

A memory size specified in gigabytes (G), megabytes (M),kilobytes (k), or bytes (b). Unlike DURATION properties, MEMORY_SIZE properties cannot combine multiple specifiers. However, MEMORY_SIZE properties allow decimal values, for example, 1.5M.

NAME

A valid cn (common name).

`OCTAL_MODE`
A three-digit, octal file permissions specifier. The first digit specifies permissions for the server user ID, the second for the server group ID, the last for other users. Each digit consists of a bitmask defining read (`4`), write (`2`), execute (`1`), or no access (`0`) permissions, thus `640` specifies read-write access for the server user, read-only access for other users of the server group, and no access for other users.

`PASSWORD_FILE`
The full path to the file from which the bind password should be read.

`PATH`
A valid, absolute file system path.

`STRING`
A DirectoryString value, as specified by RFC 2252 (`http://www.ietf.org/rfc/rfc2252.txt`).

`SUPPORTED_SSL_CIPHER`
An SSL cipher supported by the server. See the Reference for a list of supported ciphers.

`SUPPORTED_SSL_PROTOCOL`
An SSL protocol supported by the server. See the Reference for a list of supported protocols.

`TIME`
A time of the form *hhmm* in 24-hour format, where *hh* stands for hours and *mm* stands for minutes.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Evolving

See Also

dsconf(1M), index(5dsconf), server(5dsconf), suffix(5dsconf)

Name controls – LDAP controls handled by Directory Server

Description LDAPv3 controls specify extension information sent as part of a request. An explanation of what an LDAPv3 control is can be found in RFC 2251.

Directory Server handles the LDAP controls listed here according to their interface stability. See `attributes`(5) for descriptions of interface stability.

INTERFACE STABILITY: STANDARD

`1.2.840.113556.1.4.473`	Server-side sort request, described in RFC 2891
`2.16.840.1.113730.3.4.2`	Manage DSA IT control, described in RFC 3296
`2.16.840.1.113730.3.4.15`	Authorization bind identity response control, described in RFC 3829
`2.16.840.1.113730.3.4.16`	Authorization bind identity request control, described in RFC 3829
`2.16.840.1.113730.3.4.18`	Proxied authorization (version 2) control, described in RFC 4370.

INTERFACE STABILITY: EXTERNAL

`1.3.6.1.4.1.42.2.27.8.5.1`	Password policy control
`2.16.840.1.113730.3.4.3`	Persistent search control
`2.16.840.1.113730.3.4.9`	Virtual list view request control

INTERFACE STABILITY: STABLE

`1.3.6.1.4.1.42.2.27.9.5.2`	Get effective rights request control
`1.3.6.1.4.1.42.2.27.9.5.8`	Account usability control
`2.16.840.1.113730.3.4.4`	Password expired notification control
`2.16.840.1.113730.3.4.5`	Password expiring notification control
`2.16.840.1.113730.3.4.14`	Specific backend search request control
`2.16.840.1.113730.3.4.17`	Real attributes only request control
`2.16.840.1.113730.3.4.19`	Virtual attributes only request control

INTERFACE STABILITY: PRIVATE

`1.3.6.1.4.1.1466.29539.12`	Chained request control
`1.3.6.1.4.1.42.2.27.9.5.6`	Directory Server initialization control
`2.16.840.1.113730.3.4.13`	Replication update information control

INTERFACE STABILITY: DEPRECATED

The following control is scheduled for removal.

`2.16.840.1.113730.3.4.12`	Proxied authorization (version 1) control

Name

db-path – Path to Directory Server database files

Description

Syntax	`PATH`
Default Value	*instance-path*`/db`
Is readable	Yes
Is modifiable	No (server instance level), Yes (suffix level)
Is multi-valued	No

This property specifies the default file system directory containing the server database files.

This property is modifiable at the suffix level. At server instance level, the property is set when the server instance is created, and cannot be modified.

When changing this property, you must stop the server, delete the existing database, and reimport all suffixes from LDIF, before restarting the server.

Description

Syntax values shown in lower case or partly in lower case are literal values.

Those shown in upper case are syntax types, defined as follows:

`ATTR_NAME`
A valid attribute type name such as `cn` or `objectClass`.

`BOOLEAN`
`true` or `false`.

`DN`
A valid distinguished name such as `ou=People,dc=example,dc=com`.

`DURATION`
A duration specified in months (`M`), weeks (`w`), days (`d`), hours (`h`), minutes (`m`), seconds (`s`), and miliseconds (`ms`), or some combination with multiple specifiers. For example, you can specify one week as `1w`, `7d`, `168h`, `10080m`, or `604800s`. You can also specify one week as `1w0d0h0m0s`.

`DURATION` properties typically do not each support all duration specifiers (`Mwdhms`). Examine the output of `dsconf help-properties` for the property to determine which duration specifiers are supported.

`EMAIL_ADDRESS`
A valid e-mail address.

`HOST_NAME`
An IP address or host name.

`INTEGER`
A positive integer value between 0 and the maximum supported integer value in the system address space. On 32-bit systems, `2147483647`. On 64-bit systems, `9223372036854775807`.

`INTERVAL`
An interval value of the form *hhmm-hhmm 0123456*, where the first element specifies the starting hour, the next element the finishing hour in 24-hour time format, from `0000-2359`, and the second specifies days, starting with Sunday (`0`) to Saturday (`6`).

`IP_RANGE`
An IP address or range of address in one of the following formats:

- IP address in dotted decimal form.
- IP address and bits, in the form of network number/mask bits.
- IP address and quad, in the form of a pair of dotted decimal quads.
- All address. A catch-all for clients that are note placed into other, higher priority groups.
- `0.0.0.0`. This address is for groups to which initial membership is not considered. For example, for groups that clients switch to after their initial bind.
- IP address of the local host.

`LDAP_URL`
A valid LDAP URL as specified by RFC 2255 (`http://www.ietf.org/rfc/rfc2255.txt`).

`MEMORY_SIZE`
A memory size specified in gigabytes (`G`), megabytes (`M`),kilobytes (`k`), or bytes (`b`). Unlike `DURATION` properties, `MEMORY_SIZE` properties cannot combine multiple specifiers. However, `MEMORY_SIZE` properties allow decimal values, for example, 1.5M.

`NAME`
A valid `cn` (common name).

`OCTAL_MODE`
A three-digit, octal file permissions specifier. The first digit specifies permissions for the server user ID, the second for the server group ID, the last for other users. Each digit consists of a bitmask defining read (`4`), write (`2`), execute (`1`), or no access (`0`) permissions, thus `640` specifies read-write access for the server user, read-only access for other users of the server group, and no access for other users.

`PASSWORD_FILE`
The full path to the file from which the bind password should be read.

`PATH`
A valid, absolute file system path.

`STRING`
A DirectoryString value, as specified by RFC 2252 (`http://www.ietf.org/rfc/rfc2252.txt`).

`SUPPORTED_SSL_CIPHER`
An SSL cipher supported by the server. See the Reference for a list of supported ciphers.

`SUPPORTED_SSL_PROTOCOL`
An SSL protocol supported by the server. See the Reference for a list of supported protocols.

`TIME`
A time of the form *hhmm* in 24-hour format, where *hh* stands for hours and *mm* stands for minutes.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Evolving

See Also

dsconf(1M), server(5dsconf), suffix(5dsconf)

Name

desc – Optional description of configuration element

Description

Syntax	STRING
Default Value	None
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

Use this optional property to provide a short description of the configuration element.

Description

Syntax values shown in lower case or partly in lower case are literal values.

Those shown in upper case are syntax types, defined as follows:

`ATTR_NAME`
A valid attribute type name such as `cn` or `objectClass`.

`BOOLEAN`
`true` or `false`.

`DN`
A valid distinguished name such as `ou=People,dc=example,dc=com`.

`DURATION`
A duration specified in months (`M`), weeks (`w`), days (`d`), hours (`h`), minutes (`m`), seconds (`s`), and miliseconds (`ms`), or some combination with multiple specifiers. For example, you can specify one week as `1w`, `7d`, `168h`, `10080m`, or `604800s`. You can also specify one week as `1w0d0h0m0s`.

`DURATION` properties typically do not each support all duration specifiers (`Mwdhms`). Examine the output of `dsconf help-properties` for the property to determine which duration specifiers are supported.

`EMAIL_ADDRESS`
A valid e-mail address.

`HOST_NAME`
An IP address or host name.

`INTEGER`
A positive integer value between 0 and the maximum supported integer value in the system address space. On 32-bit systems, `2147483647`. On 64-bit systems, `9223372036854775807`.

INTERVAL
An interval value of the form *hhmm*-*hhmm 0123456*, where the first element specifies the starting hour, the next element the finishing hour in 24-hour time format, from 0000-2359, and the second specifies days, starting with Sunday (0) to Saturday (6).

IP_RANGE
An IP address or range of address in one of the following formats:

- IP address in dotted decimal form.
- IP address and bits, in the form of network number/mask bits.
- IP address and quad, in the form of a pair of dotted decimal quads.
- All address. A catch-all for clients that are note placed into other, higher priority groups.
- 0.0.0.0. This address is for groups to which initial membership is not considered. For example, for groups that clients switch to after their initial bind.
- IP address of the local host.

LDAP_URL
A valid LDAP URL as specified by RFC 2255 (http://www.ietf.org/rfc/rfc2255.txt).

MEMORY_SIZE
A memory size specified in gigabytes (G), megabytes (M),kilobytes (k), or bytes (b). Unlike DURATION properties, MEMORY_SIZE properties cannot combine multiple specifiers. However, MEMORY_SIZE properties allow decimal values, for example, 1.5M.

NAME
A valid cn (common name).

OCTAL_MODE
A three-digit, octal file permissions specifier. The first digit specifies permissions for the server user ID, the second for the server group ID, the last for other users. Each digit consists of a bitmask defining read (4), write (2), execute (1), or no access (0) permissions, thus 640 specifies read-write access for the server user, read-only access for other users of the server group, and no access for other users.

PASSWORD_FILE
The full path to the file from which the bind password should be read.

PATH
A valid, absolute file system path.

STRING
A DirectoryString value, as specified by RFC 2252 (http://www.ietf.org/rfc/rfc2252.txt).

SUPPORTED_SSL_CIPHER
An SSL cipher supported by the server. See the Reference for a list of supported ciphers.

`SUPPORTED_SSL_PROTOCOL`
An SSL protocol supported by the server. See the Reference for a list of supported protocols.

`TIME`
A time of the form *hhmm* in 24-hour format, where *hh* stands for hours and *mm* stands for minutes.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Evolving

See Also

dsconf(1M), encryption(5dsconf), index(5dsconf), plugin(5dsconf), repl-agmt(5dsconf)

Name ds5AgreementEnable – Whether replication is enabled

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	`on` | `off`
Default Value	`on`
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

Specifies whether the replication agreement is enabled.

Examples

```
ds5AgreementEnable: on
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

ds5BeginReplicaAcceptUpdates – Accept, rather than refer, update operations

Description

PROPERTY	VALUE
Entry DN	`cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	`start`|`stop`
Default Value	None
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5Replica` entries.

When this attribute is set to `start`, the server accepts client updates rather than referring them to another server.

Examples

```
ds5BeginReplicaAcceptUpdates: start
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

ds5LastInitTimeStamp – Time stamp for last initialization

Description

PROPERTY	VALUE
Entry DN	`cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	N/A
Default Value	N/A
Syntax	Integer

This attribute is part of replica configuration for `nsDS5Replica` entries.

This attribute is reserved for internal use.

Examples

```
ds5LastInitTimeStamp: 0
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name ds5ReferralDelayAfterInit – Accept update operations after the specified delay

Description

PROPERTY	VALUE
Entry DN	`cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	`0` to any 64-bit integer (seconds)
Default Value	Not set (unlimited)
Syntax	Integer

This attribute is part of replica configuration for `nsDS5Replica` entries.

When this attribute is set, the server starts accepting client updates after waiting the number of seconds you specify.

Examples

```
ds5ReferralDelayAfterInit: 100
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also replication(5dsconf)

Name

ds5ReplicaAutomaticInit – Automatically initialize dedicated consumer

Description

PROPERTY	VALUE
Entry DN	cn=*agreement name*,cn=replica,cn=*suffix name*,cn=mapping tree,cn=config
Valid Range	on | off
Default Value	N/A
Syntax	DirectoryString

This attribute is part of replica configuration for nsDS5ReplicationAgreement entries.

When set to on, this attribute causes the server to perform a total update of the consumer replica as soon as replication fails to proceed normally. Use this attribute only in agreements toward read-only, dedicated consumer replica.

Examples

```
ds5ReplicaAutomaticInit: on
```

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

ds5ReplicaConsumerTimeout – Timeout for replication operations

Description

PROPERTY	VALUE
Entry DN	`cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	`0` maximum integer (seconds)
Default Value	`300` (seconds)
Syntax	Integer

This attribute is part of replica configuration for `nsDS5Replica` entries.

When set, this attribute causes the consumer to close a replication connection with the current supplier, allowing it to open a replication session with another supplier.

This attribute takes effect under the following conditions on the consumer.

- The supplier sending updates on the current replication session connection has been idle for `ds5ReplicaConsumerTimeout` seconds.
- No local operation on the consumer is currently replaying updates from the current supplier.
- Another supplier is currently attempting to start a replication session.

Examples

Note – Do not change the value of this attribute unless requested to do so by qualified support personnel.

```
ds5ReplicaConsumerTimeout: 300
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name ds5ReplicaForce51Protocol – Force use of DS 5.1 replication protocol

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	`on` | `off`
Default Value	N/A
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

When set to `on`, this attribute forces the supplier to use the DS 5.1 replication protocol with the consumer. Use of this attribute is *not* necessary to replicate with a DS 5.1 replica.

Examples `ds5ReplicaForce51Protocol: on`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also replication(5dsconf)

Name

ds5ReplicaTransportCompressionLevel – Compression used for replication

Description

PROPERTY	VALUE
Entry DN	cn=*agreement name*,cn=replica,cn=*suffix name*,cn=mapping tree,cn=config
Valid Range	0 | 1 | 2 | 3
Default Value	0
Syntax	Integer

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

Specifies the type of compression used for replication protocol messages.

This attribute takes the following values:

`0` No compression

`1` Default zlib compression (zlib numeric value = `-1`)

`2` Fastest zlib compression (zlib numeric value = `1`)

`3` Strongest zlib compression (zlib numeric value = `9`)

If the bottleneck for replication in your environment is network bandwidth, this attribute can potentially help you tune the replication protocol for better performance.

Examples

```
ds5ReplicaTransportCompressionLevel: 2
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

ds5ReplicaTransportConcurrencyLevel – Throttle replication concurrency

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	`0|1|2`
Default Value	`2`
Syntax	Integer

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

This attribute lets you force the supplier to replay replicated updates on the consumer in the order they occurred on the supplier, or even by a single thread so that the replayed updates happen sequentially. The following values are supported:

`0` Replay updates in sequential order using a single thread. This can reduce throughput.

`1` Replay updates in sequential order using multiple threads. This can still reduce throughput compared to the default behavior.

`2` (Default) Replay unrelated updates in parallel to increase throughput.

Avoid tweaking this attribute as in most cases all you will manage to do is reduce replication performance.

Examples

```
ds5ReplicaTransportConcurrencyLevel: 2
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

ds5ReplicaTransportGroupSize – Grouping size for replication updates

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	`1` to `255`
Default Value	`1`
Syntax	Integer

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

When this attribute is set, the supplier groups updates, for an incremental update, or entries, for total update, before sending those updates or entries to the consumer.

If the bottleneck for replication in your environment is network bandwidth, this attribute can potentially help you tune the replication protocol for better performance.

Examples

```
ds5ReplicaTransportGroupSize: 10
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

ds5ReplicaTransportGrpPktSize – Effective group packet size

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	`0` to `65536`
Default Value	N/A
Syntax	Integer

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

This attribute governs the effective packet size sent by the supplier such that maximum BER size is not exceeded on the consumer.

Do not modify this attribute unless told to do so by qualified support personnel.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

ds5ReplicaTransportWindowSize – Window size for replication updates

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	`1` to `65535`
Default Value	`10`
Syntax	Integer

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

The supplier sends up to the specified number of replication messages to the consumer before waiting for a response from the consumer to continue.

If the bottleneck for replication in your environment is network latency or network bandwidth, this attribute can potentially help you tune the replication protocol for better performance.

Examples

```
ds5ReplicaTransportWindowSize: 100
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

ds6ruv – Replication update vector, version 6

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	N/A
Default Value	N/A
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

This attribute is responsible for managing the internal state of the replica via the replication update vector. It is always present and must not be changed.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

dsChangelogMaxAge – Maximum age of change log entries

Description

PROPERTY	VALUE
Entry DN	`cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	See the description that follows.
Default Value	`7d` (one week)
Syntax	IntegerTimeUnit

This attribute is part of replica configuration for `nsDS5Replica` entries.

The change log contains a record for each directory modification and is used when synchronizing consumer servers. Each record contains a timestamp. Any record with a timestamp that is older than the value specified in this attribute is removed.

When this attribute is set, the server purges change log entries older than the time you specify. Age is specified as a number followed by a letter `s` for seconds, `m` for minutes, `h` for hours, `d` for days, or `w` for weeks. If this attribute is set to 0, entries are not removed according to their age. If this attribute is not present, the default age limit on change log records is one week (`7d`).

Examples

```
dsChangelogMaxAge: 7d
```

See Also

replication(5dsconf)

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name

dsChangelogMaxentries – Maximum number of change log records

Description

PROPERTY	VALUE
Entry DN	`cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	`-1` to maximum integer
Default Value	`-1` (unlimited)
Syntax	Integer

This attribute is part of replica configuration for `nsDS5Replica` entries.

The change log contains a record for each directory modification and is used when synchronizing consumer servers. Each record contains a timestamp. Any record with a timestamp that is older than the value specified in this attribute is removed.

When this attribute is set, the server purges entries from the change log after the maximum you specify is reached. If this attribute is absent, or if it is set to `-1`, the server does no limit the number of entries in the change log.

Examples

```
dsChangelogMaxentries: 5000
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

dsFilterSPConfigchecksum – Checksum for partial replication

Description

PROPERTY	VALUE
Entry DN	cn=*agreement name*,cn=replica,cn=*suffix name*,cn=mapping tree,cn=config (supplier), cn=replica,cn=*suffix name*,cn=mapping tree,cn=config (consumer)
Valid Range	N/A
Default Value	N/A
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5Replica` and `nsDS5ReplicationAgreement` entries.

This read-only attribute is reserved for internal use. Do not modify its value.

Examples

```
dsFilterSPConfigchecksum: 0
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name ds-hdsml-clientauthmethod – DSML SSL client authentication

Description Defines how the server will identify a client on a secure (SSL) connection.

Entry DN `cn=DSMLv2-SOAP-HTTP,cn=frontends,cn=plugins, cn=config`

Valid Range `clientCertOnly`: the server uses the credentials from the client certificate to identify the client.

`httpBasicOnly`: the server uses the credentials from the HTTP authorization header to identify the client.

`clientCertFirst`: the server attempts to use the client certificate credentials to identify the client. If there are no client certificate credentials, credentials from the HTTP authorization header are used.

Default Value `clientCertFirst`

Syntax DirectoryString

Example `ds-hdsml-clientauthmethod: clientCertFirst`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name ds-hdsml-dsmlschemalocation – DSMLv2 schema location

Description The path to the DSMLv2 schema. This is generated automatically and should not be changed.

Entry DN	`cn=DSMLv2-SOAP-HTTP,cn=frontends,cn=plugins, cn=config`
Valid Range	Any valid path to the directory storing the DSML schema.
Default Value	*install-path*`/ds6/lib/DSMLv2.xsd`
Syntax	DirectoryString
Example	`ds-hdsml-dsmlschemalocation: /opt/SUNWdsee/ds6/lib/DSMLv2.xsd`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name ds-hdsml-iobuffersize – buffer size for DSML requests

Description The size of the buffer in which the DSML request is stored. If Directory Server receives many large DSML requests, such as large modify requests, then increasing this value may allow fewer buffers to be passed from the HTTP front end to the DSML parsers.

Entry DN	`cn=DSMLv2-SOAP-HTTP,cn=frontends,cn=plugins, cn=config`
Valid Range	1 to an appropriate upper limit for your deployment, with a maximum of 2147483647 (2^31-1). The value must be a multiple of 256.
Default Value	`8192`
Syntax	Integer
Example	`ds-hdsml-buffersize: 8192`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name ds-hdsml-poolmaxsize – maximum number of DSML parsers

Description The maximum number of DSML parsers kept ready to handle DSML requests. If you expect sustained traffic of many concurrent DSML requests, you may choose to increase the value of this attribute.

Entry DN	`cn=DSMLv2-SOAP-HTTP,cn=frontends, cn=plugins,cn=config`
Valid Range	1 to an appropriate upper limit for your deployment, with a maximum of 2147483647 (2^{31}-1).
Default Value	`10`
Syntax	Integer
Example	`ds-hdsml-poolmaxsize: 10`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name ds-hdsml-poolsize – default number of DSML parsers

Description The minimum, default number of DSML parsers kept ready to handle DSML requests. If you expect sustained traffic of many concurrent DSML requests, you may choose to increase the value of this attribute.

Entry DN	`cn=DSMLv2-SOAP-HTTP,cn=frontends,cn=plugins, cn=config`
Valid Range	1 to an appropriate upper limit for your deployment, with a maximum of 2147483647 (2^{31}-1).
Default Value	`5`
Syntax	Integer
Example	`ds-hdsml-poolsize: 5`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name ds-hdsml-port – DSML port number

Description The HTTP port used for DSML communications. The selected port must be unique on the host system; make sure no other application is attempting to use the same port number. Specifying a port number of less than 1024 requires Directory Server to run as super user.

You must restart the server for a port number change to be taken into account.

Entry DN	`cn=DSMLv2-SOAP-HTTP,cn=frontends,cn=plugins, cn=config`
Valid Range	1-65535
Default Value	`disabled`
Syntax	Integer
Example	`ds-hdsml-port: 8080`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name ds-hdsml-requestmaxsize – maximum DSML request size

Description The maximum size of a DSML request. If the request is larger than this value, the server responds with the error message `REQUEST_ENTITY_TOO_LARGE` and closes the connection to prevent the client from continuing the request.

Entry DN	`cn=DSMLv2-SOAP-HTTP,cn=frontends,cn=plugins, cn=config`
Valid Range	1-2147483647 (2^{31}-1)
Default Value	`32768`
Syntax	Integer
Example	`ds-hdsml-requestmaxsize: 32768`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name

ds-hdsml-responsemsgsize – maximum size of DSML response

Description

The maximum size of a server response to a DSML request, or a fraction of the maximum response size in the case of intermediate search responses. If the response is larger than the size specified here.

Entry DN	`cn=DSMLv2-SOAP-HTTP,cn=frontends,cn=plugins, cn=config`
Valid Range	1-2147483647 (2^31-1)
Default Value	`65536`
Syntax	Integer
Example	`ds-hdsml-responsemsgsize: 65536`

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

dse.ldif(4)

Name ds-hdsml-rooturl – root URL for DSML

Description The root URL used in the HTTP POST request to indicate the request is DSML. On the client side, this corresponds to the first line of the post, such as:

```
POST /dsml HTTP/1.1
```

Client applications must post to the value of this attribute.

Entry DN	`cn=DSMLv2-SOAP-HTTP,cn=frontends,cn=plugins, cn=config`
Valid Range	Any valid URL.
Default Value	`/dsml`
Syntax	DirectoryString
Example	`ds-hdsml-rooturl: /dsml`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name ds-hdsml-secureport – DSML SSL port number

Description The port number used for secure DSML communications (over SSL). The selected port must be unique on the host system; make sure no other application is attempting to use the same port number. Specifying a port number of less than 1024 requires Directory Server to run as super user.

You must restart the server for a port number change to be taken into account.

Entry DN	`cn=DSMLv2-SOAP-HTTP,cn=frontends,cn=plugins, cn=config`
Valid Range	1-65535
Default Value	None
Syntax	Integer
Example	`ds-hdsml-secureport: 1443`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name ds-hdsml-soapschemalocation – SOAP schema location for DSML

Description The path to the SOAP schema. This is generated automatically and should not be changed.

Entry DN	`cn=DSMLv2-SOAP-HTTP,cn=frontends,cn=plugins, cn=config`
Valid Range	Any valid path to the directory storing the SOAP schema.
Default Value	*install-path*`/ds6/lib/soap-env.xsd`
Syntax	DirectoryString

Example	`ds-hdsml-soapschemalocation:`
	`/opt/SUNWdsee/ds6/lib/soap-eng.xsd`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name

ds-maxheaphigh, ds-maxheaplow – Specify soft and hard thresholds for heap memory use

Description

PROPERTY	VALUE
Entry DN	`cn=config,cn=ldbm database,cn=plugins,cn=config`
Valid Range	Range depends on the deployment. The value is checked against the run-time values.
Default Value	Not set by default.
Syntax	Integer

These attributes specify threshold values for dynamic memory footprint. When the memory threshold is reached, Directory Server attempts to free memory from the entry caches, and limit memory use.

- When `ds-maxheaplow` is reached, Directory Server attempts to free memory concurrently with other operations.
- When `ds-maxheaphigh` is reached, Directory Server prevents operations on the cache while memory is freed.

These attributes safeguard against sudden increases of memory footprint due to changes in allocation patterns. As such, the memory thresholds should be higher than the sum of all entry caches, plus the memory footprint at startup.

`ds-maxheaphigh` and `ds-maxheaplow` must be configured in conjunction with each other, as follows.

- If `ds-maxheaphigh` is zero or is not set, `ds-maxheaplow` is ignored.
- If `ds-maxheaphigh` is set, its value must be at least one gigabyte.
- If `ds-maxheaphigh` is set, the value of `ds-maxheaplow` must be less than that of `ds-maxheaphigh`.
- If `ds-maxheaphigh` is set to a value other than zero, `ds-maxheaplow` is automatically set by default to 7/8 of the value of `ds-maxheaphigh`.
- If `ds-maxheaphigh` and `ds-maxheaplow` are both set to a value other than zero, `ds-maxheaplow` must be greater than or equal to `(ds-maxheaphigh + minheap)/2`, where `minheap` is the amount of memory used by the server at startup. If this condition is not met, `ds-maxheaplow` is automatically set by default to 7/8 of the value of `ds-maxheaphigh`.

The number of times the memory thresholds have been exceeded can be monitored by using the `heapmaxhighhits` and `heapmaxlowhits` attributes.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

heapmaxhighhits(5dsconf)

Name

dsReplFractionalExclude – Attribute types to exclude from replication

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	Any valid attribute type
Default Value	N/A
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

When this multi-valued fractional replication configuration attribute is set, the supplier does not send updates for the specified attribute types when replicating to the consumer. This attribute is mutually exclusive with dsReplFractionalInclude(5dsconf).

Examples

```
dsReplFractionalExclude: userPassword
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

dsReplFractionalInclude – Attribute types to include in replication

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	Any valid attribute type
Default Value	N/A
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

When this multi-valued fractional replication configuration attribute is set, the supplier send updates only for the specified attribute types when replicating to the consumer. This attribute is mutually exclusive with dsReplFractionalExclude(5dsconf).

Examples

```
dsReplFractionalInclude: cn
dsReplFractionalInclude: mail
dsReplFractionalInclude: objectClass
dsReplFractionalInclude: sn
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

enabled – Whether the configuration element is operational

Description

Syntax	`on	off`
Default Value	Depends on the configuration element
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

Use this property to turn on the configuration element.

Description

Syntax values shown in lower case or partly in lower case are literal values.

Those shown in upper case are syntax types, defined as follows:

`ATTR_NAME`
A valid attribute type name such as `cn` or `objectClass`.

`BOOLEAN`
`true` or `false`.

`DN`
A valid distinguished name such as `ou=People,dc=example,dc=com`.

`DURATION`
A duration specified in months (`M`), weeks (`w`), days (`d`), hours (`h`), minutes (`m`), seconds (`s`), and miliseconds (`ms`), or some combination with multiple specifiers. For example, you can specify one week as `1w`, `7d`, `168h`, `10080m`, or `604800s`. You can also specify one week as `1w0d0h0m0s`.

`DURATION` properties typically do not each support all duration specifiers (`Mwdhms`). Examine the output of `dsconf help-properties` for the property to determine which duration specifiers are supported.

`EMAIL_ADDRESS`
A valid e-mail address.

`HOST_NAME`
An IP address or host name.

`INTEGER`
A positive integer value between 0 and the maximum supported integer value in the system address space. On 32-bit systems, `2147483647`. On 64-bit systems, `9223372036854775807`.

`INTERVAL`
An interval value of the form *hhmm*-*hhmm 0123456*, where the first element specifies the starting hour, the next element the finishing hour in 24-hour time format, from `0000-2359`, and the second specifies days, starting with Sunday (`0`) to Saturday (`6`).

`IP_RANGE`
An IP address or range of address in one of the following formats:

- IP address in dotted decimal form.
- IP address and bits, in the form of network number/mask bits.
- IP address and quad, in the form of a pair of dotted decimal quads.
- All address. A catch-all for clients that are note placed into other, higher priority groups.
- `0.0.0.0`. This address is for groups to which initial membership is not considered. For example, for groups that clients switch to after their initial bind.
- IP address of the local host.

`LDAP_URL`
A valid LDAP URL as specified by RFC 2255 (`http://www.ietf.org/rfc/rfc2255.txt`).

`MEMORY_SIZE`
A memory size specified in gigabytes (`G`), megabytes (`M`),kilobytes (`k`), or bytes (`b`). Unlike `DURATION` properties, `MEMORY_SIZE` properties cannot combine multiple specifiers. However, `MEMORY_SIZE` properties allow decimal values, for example, 1.5M.

`NAME`
A valid `cn` (common name).

`OCTAL_MODE`
A three-digit, octal file permissions specifier. The first digit specifies permissions for the server user ID, the second for the server group ID, the last for other users. Each digit consists of a bitmask defining read (`4`), write (`2`), execute (`1`), or no access (`0`) permissions, thus `640` specifies read-write access for the server user, read-only access for other users of the server group, and no access for other users.

`PASSWORD_FILE`
The full path to the file from which the bind password should be read.

`PATH`
A valid, absolute file system path.

`STRING`
A DirectoryString value, as specified by RFC 2252 (`http://www.ietf.org/rfc/rfc2252.txt`).

`SUPPORTED_SSL_CIPHER`
An SSL cipher supported by the server. See the Reference for a list of supported ciphers.

`SUPPORTED_SSL_PROTOCOL`
An SSL protocol supported by the server. See the Reference for a list of supported protocols.

`TIME`
A time of the form *hhmm* in 24-hour format, where *hh* stands for hours and *mm* stands for minutes.

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Evolving

See Also dsconf(1M), log(5dsconf), plugin(5dsconf), repl-agmt(5dsconf), suffix(5dsconf)

Name

encryption, algorithm – DS attribute encryption (ETA) properties

Description

Directory Server allows you to encrypt individual attributes to protect sensitive information stored in the directory. The encryption does not prevent client applications from reading the attributes. Instead it works at the database index file level to prevent users with access to read database index files from being able to search through the indexes for sensitive information.

For example, before attribute encryption is configured for `uid` attributes, a user with read access to database index files could easily find out that `bjensen` is a `uid` attribute value:

```
$ strings example_uid.db3 | grep bjensen
=bjensen
$
```

Once `uid` attributes are encrypted, the job is not so easy:

```
$ strings example_uid.db3 | grep bjensen
$
```

Notice however that encrypted RDN values are not fully hidden. Instead they appear in clear in the DN index:

```
$ strings example_entrydn.db3 | grep bjensen
=uid=bjensen,ou=people,dc=example,dc=com
=uid=bjensen,ou=people,dc=example,dc=com
$
```

PROPERTY: `algorithm`

Syntax	`des|des3|rc2|rc4`
Default Value	None
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

Directory Server uses a cipher to encrypt a specified attribute in a given suffix. This property specifies the cipher used.

The following property values are supported:

`des`	DES block cipher
`des3`	Triple-DES block cipher
`rc2`	RC2 block cipher
`rc4`	RC4 stream cipher

Description

Syntax values shown in lower case or partly in lower case are literal values.

Those shown in upper case are syntax types, defined as follows:

`ATTR_NAME`
A valid attribute type name such as `cn` or `objectClass`.

`BOOLEAN`
`true` or `false`.

`DN`
A valid distinguished name such as `ou=People,dc=example,dc=com`.

`DURATION`
A duration specified in months (`M`), weeks (`w`), days (`d`), hours (`h`), minutes (`m`), seconds (`s`), and miliseconds (`ms`), or some combination with multiple specifiers. For example, you can specify one week as `1w`, `7d`, `168h`, `10080m`, or `604800s`. You can also specify one week as `1w0d0h0m0s`.

`DURATION` properties typically do not each support all duration specifiers (`Mwdhms`). Examine the output of `dsconf help-properties` for the property to determine which duration specifiers are supported.

`EMAIL_ADDRESS`
A valid e-mail address.

`HOST_NAME`
An IP address or host name.

`INTEGER`
A positive integer value between 0 and the maximum supported integer value in the system address space. On 32-bit systems, `2147483647`. On 64-bit systems, `9223372036854775807`.

`INTERVAL`
An interval value of the form *hhmm*-*hhmm 0123456*, where the first element specifies the starting hour, the next element the finishing hour in 24-hour time format, from `0000-2359`, and the second specifies days, starting with Sunday (`0`) to Saturday (`6`).

`IP_RANGE`
An IP address or range of address in one of the following formats:

- IP address in dotted decimal form.
- IP address and bits, in the form of network number/mask bits.
- IP address and quad, in the form of a pair of dotted decimal quads.
- All address. A catch-all for clients that are note placed into other, higher priority groups.
- `0.0.0.0`. This address is for groups to which initial membership is not considered. For example, for groups that clients switch to after their initial bind.
- IP address of the local host.

LDAP_URL
A valid LDAP URL as specified by RFC 2255 (http://www.ietf.org/rfc/rfc2255.txt).

MEMORY_SIZE
A memory size specified in gigabytes (G), megabytes (M),kilobytes (k), or bytes (b). Unlike DURATION properties, MEMORY_SIZE properties cannot combine multiple specifiers. However, MEMORY_SIZE properties allow decimal values, for example, 1.5M.

NAME
A valid cn (common name).

OCTAL_MODE
A three-digit, octal file permissions specifier. The first digit specifies permissions for the server user ID, the second for the server group ID, the last for other users. Each digit consists of a bitmask defining read (4), write (2), execute (1), or no access (0) permissions, thus 640 specifies read-write access for the server user, read-only access for other users of the server group, and no access for other users.

PASSWORD_FILE
The full path to the file from which the bind password should be read.

PATH
A valid, absolute file system path.

STRING
A DirectoryString value, as specified by RFC 2252 (http://www.ietf.org/rfc/rfc2252.txt).

SUPPORTED_SSL_CIPHER
An SSL cipher supported by the server. See the Reference for a list of supported ciphers.

SUPPORTED_SSL_PROTOCOL
An SSL protocol supported by the server. See the Reference for a list of supported protocols.

TIME
A time of the form *hhmm* in 24-hour format, where *hh* stands for hours and *mm* stands for minutes.

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Evolving

See Also

dsconf(1M), desc(5dsconf)

Name

extended-operations – LDAP extended operations handled by Directory Server

Description

LDAPv3 extended operations allow definition of additional LDAP operations not defined in RFC 2251.

Directory Server handles the LDAP extended operations described here according to their interface stability. See `attributes`(5) for descriptions of interface stability.

INTERFACE STABILITY: STANDARD

`1.3.6.1.4.1.1466.20037`	Start TLS extended operation request, described in RFC 2849
`1.3.6.1.4.1.4203.1.11.1`	Password modify extended operation, described in RFC 3062
`1.3.6.1.4.1.4203.1.11.3`	Who am I? extended operation, described in RFC 4532

INTERFACE STABILITY: PRIVATE

`1.3.6.1.4.1.42.2.27.9.6.1`	Replication protocol private extended operation
`1.3.6.1.4.1.42.2.27.9.6.2`	Replication protocol private extended operation
`1.3.6.1.4.1.42.2.27.9.6.3`	Replication protocol private extended operation
`1.3.6.1.4.1.42.2.27.9.6.4`	Replication protocol private extended operation
`1.3.6.1.4.1.42.2.27.9.6.5`	Replication protocol private extended operation
`1.3.6.1.4.1.42.2.27.9.6.6`	Replication protocol private extended operation
`1.3.6.1.4.1.42.2.27.9.6.7`	Replication protocol private extended operation
`1.3.6.1.4.1.42.2.27.9.6.8`	Replication protocol private extended operation
`1.3.6.1.4.1.42.2.27.9.6.9`	Replication protocol private extended operation
`1.3.6.1.4.1.42.2.27.9.6.11`	Replication protocol private extended operation
`1.3.6.1.4.1.42.2.27.9.6.12`	Replication protocol private extended operation
`1.3.6.1.4.1.42.2.27.9.6.13`	Replication protocol private extended operation
`1.3.6.1.4.1.42.2.27.9.6.14`	Replication protocol private extended operation
`1.3.6.1.4.1.42.2.27.9.6.15`	Replication protocol private extended operation
`1.3.6.1.4.1.42.2.27.9.6.16`	Replication protocol private extended operation
`1.3.6.1.4.1.42.2.27.9.6.17`	Replication protocol private extended operation
`1.3.6.1.4.1.42.2.27.9.6.18`	Replication protocol private extended operation
`1.3.6.1.4.1.42.2.27.9.6.19`	Replication protocol private extended operation
`1.3.6.1.4.1.42.2.27.9.6.21`	Replication protocol private extended operation
`1.3.6.1.4.1.42.2.27.9.6.22`	Replication protocol private extended operation
`1.3.6.1.4.1.42.2.27.9.6.23`	Replication protocol private extended operation
`2.16.840.1.113730.3.5.3`	Replication protocol private extended operation

2.16.840.1.113730.3.5.4	Replication protocol private extended operation
2.16.840.1.113730.3.5.5	Replication protocol private extended operation
2.16.840.1.113730.3.5.6	Replication protocol private extended operation
2.16.840.1.113730.3.5.7	Bulk import start extended operation
2.16.840.1.113730.3.5.8	Bulk import finished extended operation

Name

heapmaxhighhits, heapmaxlowhits – Counts the number of times `ds-maxheaphigh` or `ds-maxheaplow` has been exceeded

Description

PROPERTY	VALUE
Entry DN	`cn=monitor`
Valid Range	N/A
Default Value	N/A
Syntax	Integer

This read-only attribute counts the number of times that the `heapmaxhighhits` attribute or the `heapmaxlowhits` attribute has been exceeded:

- `heapmaxhighhits` counts the number of times `ds-maxheaphigh` has been exceeded
- `heapmaxlowhits` counts the number of times `ds-maxheaplow` has been exceeded

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Unstable

See Also

ds-maxheaphigh(5dsconf)

Name

index, all-ids-threshold-eq, all-ids-threshold-pres, all-ids-threshold-sub, approx-enabled, eq-enabled, matching-rule, pres-enabled, sub-enabled, system – DS attribute indexing (IDX) properties

Description

Directory Server can index attributes, making them faster to search. The `dsconf` command helps you configure five of the six supported index types:

1. *Equality indexes* to determine expediently whether an attribute value is equal to a specified value
2. *Presence indexes* to determine whether a specified attribute has any values
3. *Substring indexes* to determine whether a specified attribute has values containing a specified string, also used to compare regular expressions to attribute values
4. *Approximate indexes*, based on metaphone approximation and useful for English language strings only, to determine whether a specified attribute has any values that sound like the specified string
5. *International indexes*, also called matching rule indexes, to expedite sorting and searching in accordance with the language rules of a particular locale

The `dsconf` command does not help you configure virtual list view, also known as browsing, indexes.

The `dsconf` command does help you assign *all IDs threshold* values to indexes. As the number of entries and attribute values grows in a directory, the number of attribute values to index also grows, as does therefore the size of the indexes. In some deployments a server can end up maintaining index lists so large that the cost of rebuilding an index when attributes are modified or added outweighs the benefit the index provides for searches. All IDs thresholds limit the growth of large indexes by definining the maximum number of entry identifiers Directory Server maintains in an index list. You can define all IDs thresholds for individual indexes and for some types of indexes.

Some indexes are maintained by the server for its own use. These are called *system indexes*. In general, do not modify or remove systems indexes; such modifications could have severe repercussions on performance.

See *Directory Server Indexing* in *Sun Java System Directory Server Enterprise Edition Reference* for further details about indexing.

PROPERTY: `all-ids-threshold`

Syntax	`INTEGER	inherited`
Default Value	`inherited`
Is readable	Yes
Is modifiable	Yes

Is multi-valued	No

This property defines the maximum number of entry IDs the server maintains in an index list for the specified attribute type. By default its value is inherited from the `all-ids-threshold` setting for the suffix, whose default value in turn is inherited from the `all-ids-threshold` setting for the server, which by default is `4000`. In addition to inheritance of default settings, this property handles settings as follows:

`inherited` The threshold is inherited from the more global setting.

`<2000` The threshold value is rounded up to `2000`.

`>2000` The setting is used as a guaranteed minimum threshold. Because of internal mechanisms, the real value can be slightly more than the specified value.

After you modify this property, reindex the attribute for which you changed the threshold. For example:

```
$ dsconf set-index-prop dc=example,dc=com uid all-ids-threshold:5000
$ dsconf reindex -t uid dc=example,dc=com

## example: Indexing attribute: uid
## example: Finished indexing.

Task completed (slapd exit code: 0).
$
```

PROPERTY: `all-ids-threshold-eq`

Syntax	`INTEGER	inherited`
Default Value	`inherited`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property defines the all IDs threshold for equality indexes of the specified attribute. By default its value is inherited from the `all-ids-threshold` setting for the attribute type. See all-ids-threshold(5dsconf) for more information.

PROPERTY: `all-ids-threshold-pres`

Syntax	`INTEGER	inherited`
Default Value	`inherited`
Is readable	Yes

Is modifiable	Yes
Is multi-valued	No

This property defines the all IDs threshold for presence indexes of the specified attribute. By default its value is inherited from the `all-ids-threshold` setting for the attribute type. See all-ids-threshold(5dsconf) for more information.

PROPERTY: `all-ids-threshold-sub`

Syntax	`INTEGER	inherited`
Default Value	`inherited`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property defines the all IDs threshold for substring indexes of the specified attribute. By default its value is inherited from the `all-ids-threshold` setting for the attribute type. See all-ids-threshold(5dsconf) for more information.

PROPERTY: `approx-enabled`

Syntax	`on|off`
Default Value	`off`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property determines whether approximate indexes are maintained for the specified attribute type. You cannot set an all IDs threshold value for approximate indexes.

PROPERTY: `desc`

Syntax	`STRING`
Default Value	None
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

Use this optional property to provide a short description of the index configuration.

PROPERTY: `eq-enabled`

Syntax	`on|off`
Default Value	`on`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property determines whether equality indexes are maintained for the specified attribute type.

PROPERTY: `matching-rule`

Syntax	`STRING`
Default Value	None
Is readable	Yes
Is modifiable	Yes
Is multi-valued	Yes

This property specifies the matching rule indexes maintained for the specified attribute type.

Values for this property must be valid collation order object identifiers (OIDs). See *Directory Server Internationalized Directory* in *Directory Server Enterprise Edition Reference* for the OIDs corresponding to supported locales.

PROPERTY: `pres-enabled`

Syntax	`on|off`
Default Value	`on`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property determines whether presence indexes are maintained for the specified attribute type.

PROPERTY: `sub-enabled`

Syntax	`on|off`
Default Value	`off`

Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property determines whether substring indexes are maintained for the specified attribute type.

PROPERTY: `system`

Syntax	`true|false`
Default Value	`false`
Is readable	Yes
Is modifiable	No
Is multi-valued	No

This property identifies whether the specified index is a system index, and therefore should be left alone.

Description Syntax values shown in lower case or partly in lower case are literal values.

Those shown in upper case are syntax types, defined as follows:

`ATTR_NAME`
A valid attribute type name such as `cn` or `objectClass`.

`BOOLEAN`
`true` or `false`.

`DN`
A valid distinguished name such as `ou=People,dc=example,dc=com`.

`DURATION`
A duration specified in months (`M`), weeks (`w`), days (`d`), hours (`h`), minutes (`m`), seconds (`s`), and miliseconds (`ms`), or some combination with multiple specifiers. For example, you can specify one week as `1w`, `7d`, `168h`, `10080m`, or `604800s`. You can also specify one week as `1w0d0h0m0s`.

`DURATION` properties typically do not each support all duration specifiers (`Mwdhms`). Examine the output of `dsconf help-properties` for the property to determine which duration specifiers are supported.

`EMAIL_ADDRESS`
A valid e-mail address.

`HOST_NAME`
An IP address or host name.

INTEGER

A positive integer value between 0 and the maximum supported integer value in the system address space. On 32-bit systems, 2147483647. On 64-bit systems, 9223372036854775807.

INTERVAL

An interval value of the form *hhmm*-*hhmm 0123456*, where the first element specifies the starting hour, the next element the finishing hour in 24-hour time format, from 0000-2359, and the second specifies days, starting with Sunday (0) to Saturday (6).

IP_RANGE

An IP address or range of address in one of the following formats:

- IP address in dotted decimal form.
- IP address and bits, in the form of network number/mask bits.
- IP address and quad, in the form of a pair of dotted decimal quads.
- All address. A catch-all for clients that are note placed into other, higher priority groups.
- 0.0.0.0. This address is for groups to which initial membership is not considered. For example, for groups that clients switch to after their initial bind.
- IP address of the local host.

LDAP_URL

A valid LDAP URL as specified by RFC 2255 (http://www.ietf.org/rfc/rfc2255.txt).

MEMORY_SIZE

A memory size specified in gigabytes (G), megabytes (M),kilobytes (k), or bytes (b). Unlike DURATION properties, MEMORY_SIZE properties cannot combine multiple specifiers. However, MEMORY_SIZE properties allow decimal values, for example, 1.5M.

NAME

A valid cn (common name).

OCTAL_MODE

A three-digit, octal file permissions specifier. The first digit specifies permissions for the server user ID, the second for the server group ID, the last for other users. Each digit consists of a bitmask defining read (4), write (2), execute (1), or no access (0) permissions, thus 640 specifies read-write access for the server user, read-only access for other users of the server group, and no access for other users.

PASSWORD_FILE

The full path to the file from which the bind password should be read.

PATH

A valid, absolute file system path.

STRING

A DirectoryString value, as specified by RFC 2252 (http://www.ietf.org/rfc/rfc2252.txt).

`SUPPORTED_SSL_CIPHER`
An SSL cipher supported by the server. See the Reference for a list of supported ciphers.

`SUPPORTED_SSL_PROTOCOL`
An SSL protocol supported by the server. See the Reference for a list of supported protocols.

`TIME`
A time of the form *hhmm* in 24-hour format, where *hh* stands for hours and *mm* stands for minutes.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Evolving

See Also dsconf(1M), all-ids-threshold(5dsconf), desc(5dsconf)

Name log, buffering-enabled, level, max-age, max-disk-space-size, max-file-count, max-size, min-free-disk-space-size, path, perm, rotation-interval, rotation-min-file-size, rotation-time, verbose-enabled – DS logging configuration (LOG) properties

Description Directory Server writes to three main types of log files you can configure, the *INSTANCE_PATH*`/logs/access`, *INSTANCE_PATH*`/logs/audit`, and *INSTANCE_PATH*`/logs/errors` logs, where *INSTANCE_PATH* is the full path where the server instance is located, such as `/local/ds`.

When you specify one of these properties with `dsconf get-log-prop` or `dsconf set-log-prop`, you must specify which type of log configuration, `access`, `audit`, or `errors`, you want to examine. For example, to see whether audit logging is enabled for a server instance:

```
$ dsconf get-log-prop audit enabled
enabled  :  off
$
```

PROPERTY: `buffering-enabled`

Syntax	`on	off`
Default Value	`on` for `access`, not applicable to `audit` and `errors` logs
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property indicates whether Directory Server writes access log entries directly to disk, or use a buffer, by default.

PROPERTY: `enabled`

Syntax	`on	off`
Default Value	`on` for `access`, `off` for `audit`, `on` for `errors`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property indicates whether the specified log type is enabled.

PROPERTY: `level`

Syntax	See the description that follows.
Default Value	`default`

Is readable	Yes
Is modifiable	Yes
Is multi-valued	Yes

This property defines which kinds of messages get logged. This property is applicable only to `access`, and `errors` logs.

`access` log levels The following settings are supported:

`acc-internal`	Log access information for internal operations.
`default`	Log client access to entries.
`acc-default_plus_referrals`	As default, but also log access to referrals.
`acc-timing`	Use precise timing for microsecond resolution of elapsed times.

`errors` log levels The following settings are supported:

`default`	Log startup, shutdown, errors, and warnings.
`err-function-calls`	Log when server enters or exits a function.
`err-search-args`	Log search arguments.
`err-connection`	Connection management.
`err-packets`	Log packets sent and received.
`err-search-filter`	Log search filter information.
`err-config-file`	Log information for changes to the configuration file dse.ldif.
`err-acl`	Log access control processing information.
`err-ldbm`	Log information from the ldbm database plugin.
`err-entry-parsing`	Log LDIF parsing errors.
`err-housekeeping`	Log event queue information.
`err-replication`	Log information about replication operations.
`err-entry-cache`	Log entry cache information.
`err-plugins`	Log information from server plug-ins.
`err-dsml`	Log information from DSML front end.

`err-dsml-advanced`	Debugging information for DSML.

PROPERTY: `max-age`

Syntax	`DURATION	unlimited`
Default Value	`1M` (one month)
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property defines the age beyond which the specified type of log file is deleted.

PROPERTY: `max-disk-space-size`

Syntax	`MEMORY_SIZE	unlimited`
Default Value	`500M`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property defines the maximum disk space the specified type of log is allowed to consume. When the limit is reached, the server deletes the oldest log file to reclaim disk space.

PROPERTY: `max-file-count`

Syntax	`Integer`
Default Value	`10` for `access`, `2` for `errors`, `1` for `audit`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property defines the maximum number of log files, including rotated logs, of the specified type that the server allows to be created in the log file directory. When the limit is reached, the server deletes the oldest log file to reclaim disk space.

When you set this property to `1`, the specified log is not rotated.

PROPERTY: `max-size`

Syntax	`MEMORY_SIZE	unlimited`

Default Value	`100M`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property defines the maximum file size for the specified log. When the limit is reached, the server rotates the log file, unless `max-file-count` is set to `1`.

PROPERTY: `min-free-disk-space-size`

Syntax	`MEMORY_SIZE`
Default Value	`5M`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property defines the minimum free space allowed on the disk where the specified log is stored. When the limit is reached, the server deletes the oldest log files until enough space is available.

PROPERTY: `path`

Syntax	`PATH`
Default Value	*INSTANCE_PATH*`/logs/access`, *INSTANCE_PATH*`/logs/audit`, *INSTANCE_PATH*`/logs/errors`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property defines the full path to the specified log file type.

PROPERTY: `perm`

Syntax	`OCTAL_MODE`
Default Value	`600`
Is readable	Yes
Is modifiable	Yes

Is multi-valued	No

This property defines the read, write, and execute permissions on the specified log file.

PROPERTY: `rotation-interval`

Syntax	`DURATION	unlimited`
Default Value	`1d` (one day) for `access`, `1w` (one week) for `audit` and `errors`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property defines the duration between rotations of the specified log file.

PROPERTY: `rotation-min-file-size`

Syntax	`MEMORY_SIZE	undefined`
Default Value	`undefined`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property defines the minimum size the specified log file must have before the server rotates it.

PROPERTY: `rotation-time`

Syntax	`TIME	undefined`
Default Value	`undefined`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property defines the time of day when the server rotates the specified log file.

PROPERTY: `verbose-enabled`

Syntax	`on|off`
Default Value	`off`

Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property determines whether extra informational messages are written to the `errors` log.

Description

Syntax values shown in lower case or partly in lower case are literal values.

Those shown in upper case are syntax types, defined as follows:

`ATTR_NAME`
A valid attribute type name such as `cn` or `objectClass`.

`BOOLEAN`
`true` or `false`.

`DN`
A valid distinguished name such as `ou=People,dc=example,dc=com`.

`DURATION`
A duration specified in months (`M`), weeks (`w`), days (`d`), hours (`h`), minutes (`m`), seconds (`s`), and miliseconds (`ms`), or some combination with multiple specifiers. For example, you can specify one week as `1w`, `7d`, `168h`, `10080m`, or `604800s`. You can also specify one week as `1w0d0h0m0s`.

`DURATION` properties typically do not each support all duration specifiers (`Mwdhms`). Examine the output of `dsconf help-properties` for the property to determine which duration specifiers are supported.

`EMAIL_ADDRESS`
A valid e-mail address.

`HOST_NAME`
An IP address or host name.

`INTEGER`
A positive integer value between 0 and the maximum supported integer value in the system address space. On 32-bit systems, `2147483647`. On 64-bit systems, `9223372036854775807`.

`INTERVAL`
An interval value of the form *hhmm-hhmm 0123456*, where the first element specifies the starting hour, the next element the finishing hour in 24-hour time format, from `0000-2359`, and the second specifies days, starting with Sunday (`0`) to Saturday (`6`).

`IP_RANGE`
An IP address or range of address in one of the following formats:

- IP address in dotted decimal form.
- IP address and bits, in the form of network number/mask bits.

- IP address and quad, in the form of a pair of dotted decimal quads.
- All address. A catch-all for clients that are note placed into other, higher priority groups.
- 0.0.0.0. This address is for groups to which initial membership is not considered. For example, for groups that clients switch to after their initial bind.
- IP address of the local host.

LDAP_URL
A valid LDAP URL as specified by RFC 2255 (http://www.ietf.org/rfc/rfc2255.txt).

MEMORY_SIZE
A memory size specified in gigabytes (G), megabytes (M),kilobytes (k), or bytes (b). Unlike DURATION properties, MEMORY_SIZE properties cannot combine multiple specifiers. However, MEMORY_SIZE properties allow decimal values, for example, 1.5M.

NAME
A valid cn (common name).

OCTAL_MODE
A three-digit, octal file permissions specifier. The first digit specifies permissions for the server user ID, the second for the server group ID, the last for other users. Each digit consists of a bitmask defining read (4), write (2), execute (1), or no access (0) permissions, thus 640 specifies read-write access for the server user, read-only access for other users of the server group, and no access for other users.

PASSWORD_FILE
The full path to the file from which the bind password should be read.

PATH
A valid, absolute file system path.

STRING
A DirectoryString value, as specified by RFC 2252 (http://www.ietf.org/rfc/rfc2252.txt).

SUPPORTED_SSL_CIPHER
An SSL cipher supported by the server. See the Reference for a list of supported ciphers.

SUPPORTED_SSL_PROTOCOL
An SSL protocol supported by the server. See the Reference for a list of supported protocols.

TIME
A time of the form *hhmm* in 24-hour format, where *hh* stands for hours and *mm* stands for minutes.

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Evolving

See Also dsconf(1M), enabled(5dsconf)

Name moddn-enabled – Whether the server accepts mod DN operations

Description

Syntax	`on	off`
Default Value	`off` (`inherited` at suffix level)
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether the server accepts requests to perform modify DN operations (to move entries).

All server instances in a replication topology must be recent enough to have support for modify DN operations before you set this property to `on`.

Description Syntax values shown in lower case or partly in lower case are literal values.

Those shown in upper case are syntax types, defined as follows:

`ATTR_NAME`
A valid attribute type name such as `cn` or `objectClass`.

`BOOLEAN`
`true` or `false`.

`DN`
A valid distinguished name such as `ou=People,dc=example,dc=com`.

`DURATION`
A duration specified in months (`M`), weeks (`w`), days (`d`), hours (`h`), minutes (`m`), seconds (`s`), and miliseconds (`ms`), or some combination with multiple specifiers. For example, you can specify one week as `1w`, `7d`, `168h`, `10080m`, or `604800s`. You can also specify one week as `1w0d0h0m0s`.

`DURATION` properties typically do not each support all duration specifiers (`Mwdhms`). Examine the output of `dsconf help-properties` for the property to determine which duration specifiers are supported.

`EMAIL_ADDRESS`
A valid e-mail address.

`HOST_NAME`
An IP address or host name.

`INTEGER`
A positive integer value between 0 and the maximum supported integer value in the system address space. On 32-bit systems, `2147483647`. On 64-bit systems, `9223372036854775807`.

`INTERVAL`
An interval value of the form *hhmm-hhmm 0123456*, where the first element specifies the starting hour, the next element the finishing hour in 24-hour time format, from `0000-2359`, and the second specifies days, starting with Sunday (`0`) to Saturday (`6`).

`IP_RANGE`
An IP address or range of address in one of the following formats:

- IP address in dotted decimal form.
- IP address and bits, in the form of network number/mask bits.
- IP address and quad, in the form of a pair of dotted decimal quads.
- All address. A catch-all for clients that are note placed into other, higher priority groups.
- `0.0.0.0`. This address is for groups to which initial membership is not considered. For example, for groups that clients switch to after their initial bind.
- IP address of the local host.

`LDAP_URL`
A valid LDAP URL as specified by RFC 2255 (`http://www.ietf.org/rfc/rfc2255.txt`).

`MEMORY_SIZE`
A memory size specified in gigabytes (`G`), megabytes (`M`),kilobytes (`k`), or bytes (`b`). Unlike `DURATION` properties, `MEMORY_SIZE` properties cannot combine multiple specifiers. However, `MEMORY_SIZE` properties allow decimal values, for example, 1.5M.

`NAME`
A valid `cn` (common name).

`OCTAL_MODE`
A three-digit, octal file permissions specifier. The first digit specifies permissions for the server user ID, the second for the server group ID, the last for other users. Each digit consists of a bitmask defining read (`4`), write (`2`), execute (`1`), or no access (`0`) permissions, thus `640` specifies read-write access for the server user, read-only access for other users of the server group, and no access for other users.

`PASSWORD_FILE`
The full path to the file from which the bind password should be read.

`PATH`
A valid, absolute file system path.

`STRING`
A DirectoryString value, as specified by RFC 2252 (`http://www.ietf.org/rfc/rfc2252.txt`).

`SUPPORTED_SSL_CIPHER`
An SSL cipher supported by the server. See the Reference for a list of supported ciphers.

`SUPPORTED_SSL_PROTOCOL`
An SSL protocol supported by the server. See the Reference for a list of supported protocols.

`TIME`
A time of the form *hhmm* in 24-hour format, where *hh* stands for hours and *mm* stands for minutes.

Attributes

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Evolving

See Also

dsconf(1M), server(5dsconf), suffix(5dsconf)

Name

nsAbandonedSearchCheckInterval – interval between checks for abandoned chaining operations

Description

The number of seconds that pass before the server checks for abandoned operations.

Entry DN	`cn=default instance config,cn=chaining database, cn=plugins,cn=config`
Valid Range	0 to 2147483647 seconds
Default Value	`2`
Syntax	Integer
Example	`nsabandonedsearchcheckinterval: 10`

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

dse.ldif(4)

Name nsActiveChainingComponents – components using chaining

Description Lists the components using chaining. A component is any functional unit in the server. The value of this attribute overrides the value in the global configuration attribute. To disable chaining on a particular database instance, use the value `None`.

This attribute also allows you to alter the components used to chain. By default, no components are allowed to chain. For this reason, this attribute does not appear in a list of `cn=config,cn=chaining database,cn=config` attributes, as LDAP considers empty attributes to be nonexistent.

Entry DN	`cn=config,cn=chaining database,cn=plugins,cn=config`
Valid Range	Any valid component entry.
Default Value	None
Syntax	DirectoryString
Example	`nsActiveChainingComponents: cn=uid uniqueness,cn=plugins,cn=config`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsBindConnectionsLimit – maximum TCP connections for chaining

Description Maximum number of TCP connections the chained suffix establishes with the remote server.

Entry DN	`cn=default instance config,cn=chaining database, cn=plugins,cn=config`
Valid Range	1 to 50 connections
Default Value	`3`
Syntax	Integer
Example	`nsbindconnectionslimit: 3`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsBindRetryLimit – maximum bind attemps for chaining suffix

Description Number of times a chained suffix attempts to bind with the remote server if the initial bind attempt is unsuccessful. A value of 0 here indicates that the chained suffix will only attempt to bind once only.

Entry DN	`cn=default instance config,cn=chaining database, cn=plugins,cn=config`
Valid Range	0 to 5
Default Value	`3`
Syntax	Integer
Example	`nsbindretrylimit: 3`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name

nsBindTimeout – timeout for chaining binds

Description

Period of time before the bind attempt times out. There is no real Valid Range for this attribute, except reasonable patience limits.

Entry DN	`cn=default instance config,cn=chaining database,`
	`cn=plugins,cn=config`

Valid Range	0 to 60 seconds
Default Value	`15`
Syntax	Integer
Example	`nsbindtimeout:15`

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

dse.ldif(4)

Name nsCheckLocalACI – evaluate access control on chained suffix

Description Reserved for advanced use only. Controls whether ACIs are evaluated on the chained suffix as well as the remote data server. Changes to this attribute only take effect once the server has been restarted.

Entry DN	`cn=default instance config,cn=chaining database,`
	`cn=plugins,cn=config`

Valid Range	on | off
Default Value	`off`
Syntax	DirectoryString
Example	`nschecklocalaci: on`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsConcurrentBindLimit – maximum concurrent binds on TCP connection for chaining

Description The maximum number of concurrent bind operations per TCP connection.

Entry DN	`cn=default instance config,cn=chaining database, cn=plugins,cn=config`
Valid Range	1 to 25 binds
Default Value	`10`
Syntax	Integer
Example	`nsconcurrentbindlimit:10`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsConcurrentOperationsLimit – maximum concurrent operations for chaining

Description The maximum number of concurrent operations allowed.

Entry DN	`cn=default instance config,cn=chaining database, cn=plugins,cn=config`
Valid Range	1 to 50 operations
Default Value	`50`
Syntax	Integer
Example	`nsconcurrentoperationslimit: 50`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsConnectionLife – connection lifetime for chaining

Description Specifies the connection lifetime. You can keep connections between the chained suffix and the remote server open for an unspecified time, or you can close them after a specific period of time. Keeping the connections open is faster, but uses more resources. When the value is 0 and a list of failover servers is provided in the `nsFarmServerURL` attribute, the “main” server is never contacted after failover to the alternate server.

Entry DN `cn=default instance config,cn=chaining database, cn=plugins,cn=config`

Valid Range 0 to limitless seconds (where 0 means forever)

Default Value `0`

Syntax Integer

Example `nsconnectionlife: 0`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name

nsds50ruv – Replication update vector, version 5

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	N/A
Default Value	N/A
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

This attribute is responsible for managing the internal state of the replica via the replication update vector. It is always present and must not be changed.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

nsds5BeginReplicaRefresh – Force initialization of the consumer

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	`start` | `stop`
Default Value	N/A
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

When set to `start`, this attribute causes the supplier to perform at total update of the consumer.

Examples

```
nsds5BeginReplicaRefresh: start
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

nsDS5Flags – Change logging and referral flags

Description

PROPERTY	VALUE
Entry DN	`cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	`0 | 1 | 4 | 5`
Default Value	`1` (master), `0` (consumer)
Syntax	Integer

This attribute is part of replica configuration for `nsDS5Replica` entries.

This attribute takes one of the following values:

`0` No changes are logged. Automatic referrals are not overwritten.

`1` Changes are logged. Automatic referrals are not overwritten.

`4` No changes are logged. Automatic referrals are overwritten.

`5` Changes are logged. Automatic referrals are overwritten.

Examples

```
nsDS5Flags: 1
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

nsDS5ReplicaAutoReferral – Reserved for internal use

Description

PROPERTY	VALUE
Entry DN	`cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	N/A
Default Value	None
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5Replica` entries.

This attribute is reserved for internal use.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

nsDS5ReplicaBindDN – Bind DN for replication operations

Description

PROPERTY	VALUE
Entry DN	`cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	Any valid account DN
Default Value	`cn=replication manager,cn=replication,cn=config`
Syntax	DN

This attribute is part of replica configuration for `nsDS5Replica`, and `nsDS5ReplicationAgreement` entries.

When this multi-valued attribute is set, the accounts with the specified DNs can be used by the server to bind before performing replication operations. The DNs specified in this attribute can be used in replication agreements on the supplier side, and to bind on the consumer side. The DN can either be a local entry on the consumer server or, in the case of an SSL connection, the certificate identity associated with the same DN.

Examples

```
nsDS5ReplicaBindDN: cn=replication manager,cn=replication,
 cn=config
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

nsDS5ReplicaBindMethod – Bind protocol used for replication

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	`SIMPLE` | `SSLCLIENTAUTH`
Default Value	`SIMPLE`
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

Specifies the bind protocol used for replication. When this attribute is set to SIMPLE, simple authentication is used. When this attribute is set to SSLLCIENTAUTH, SSL client authentication is used.

Examples

```
nsDS5ReplicaBindMethod: SIMPLE
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name
nsDS5ReplicaChangeCount – Number of entries in the change log

Description

PROPERTY	VALUE
Entry DN	`cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	N/A
Default Value	N/A
Syntax	Integer

This attribute is part of replica configuration for `nsDS5Replica` entries.

This read-only attribute shows the number of entries remaining in the change log. The change log is purged according to how `nsslapd-changelogmaxage` and `nsslapd-changelogmaxentries` are set.

Examples

```
nsDS5ReplicaChangeCount: 10
```

Attributes
See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also
replication(5dsconf)

Name

nsds5replicaChangesSentSinceStartup – Number of updates since startup

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	`0` maximum integer
Default Value	None
Syntax	Integer

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

This read-only attribute shows the number of changes sent to this replica since the server started.

Examples

```
nsds5replicaChangesSentSinceStartup: 161803399
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

nsDS5ReplicaCredentials – Credentials for replication operations

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	N/A
Default Value	N/A
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

Specifies the credentials on the consumer for the account with DN nsDS5ReplicaBindDN(5dsconf) used for replication configured to use simple authentication.

Examples

```
nsDS5ReplicaCredentials:: e0RFU31JakduS3VZSWhEcThEcExDQlU2
 VlN2QTdjcUw4emhDdXl3Sldmc3NTZ2t3eS9mWmR4VmpUZlVYRE1NLzR2T
 UVBDQpyZVdYU3A3U1ZwYz0=
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

nsDS5ReplicaHost – Host name of consumer

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	Any valid host name
Default Value	N/A
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

Specifies the hostname for the host where the consumer replica is located. Do not modify this attribute after it has been set.

Examples

```
nsDS5ReplicaHost: ds.example.com
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

nsDS5ReplicaId – Replica identification number

Description

PROPERTY	VALUE
Entry DN	`cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	1-65534 (master), 65535 (consumer or hub)
Default Value	None
Syntax	Integer

This attribute is part of replica configuration for `nsDS5Replica` entries.

This attribute specifies a unique ID for a master replica in a particular topology, or the ID `65535` for a consumer, or for a hub.

Examples

```
nsDS5ReplicaId: 1
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

nsds5replicaLastInitEnd – Time of last initialization

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	A valid timestamp
Default Value	None
Syntax	GeneralizedTime

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

This read-only attribute shows when the most recent initialization of the replica finished.

Examples

```
nsds5replicaLastInitEnd: 20051223113229
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

nsds5replicaLastInitStart – Time of last initialization

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	A valid timestamp
Default Value	None
Syntax	GeneralizedTime

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

This read-only attribute shows when the most recent initialization of the replica started.

Examples

```
nsds5replicaLastInitStart: 20051223113214
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

nsds5replicaLastInitStatus – Replica initialization status

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	A message concerning initialization
Default Value	None
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

This read-only attribute shows the status of the most recent replication initialization.

Examples

```
nsds5replicaLastInitStatus: 0 Consumer Initialization Succeeded
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

nsds5replicaLastUpdateEnd – Time of last update

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	A valid timestamp
Default Value	None
Syntax	GeneralizedTime

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

This read-only attribute shows when the most recent update of the replica finished.

Examples

```
nsds5replicaLastUpdateEnd: 20051223113229
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

nsds5replicaLastUpdateStart – Time of last update

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	A valid timestamp
Default Value	None
Syntax	GeneralizedTime

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

This read-only attribute shows when the most recent update of the replica started.

Examples

```
nsds5replicaLastUpdateStart: 20051223113214
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name nsds5replicaLastUpdateStatus – Replica update status

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	A message concerning the last update
Default Value	None
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

This read-only attribute shows the status of the latest update of the replica.

Messages include the following.

`-1 Could not bind to replica`
Solution: The credentials are wrong for the replication manager bind DN. Fix the replication agreement.

`-1 Incremental update has failed and requires a total update`
Solution: Reinitialize the replica. See the `errors` log to determine what led to the problem.

`-1 Incremental update has failed and requires administrator action`
Solution: Reinitialize the replica. See the `errors` log to determine what led to the problem.

`-1 Internal error: Could not get access to the replica RUV`
Solution: Replication stopped and requires a total update. Reinitialize the replica.

`-1 Partial replication configuration error`
Solution: Replication stopped and requires configuration fix. Fix the configuration.

`-1 Partial replication configuration has changed`
Solution: Replication stopped and requires a total update. Reinitialize the replica.

`-1 Total update required`
Solution: Reinitialize the replica. See the `errors` log to determine what led to the problem.

`0 Incremental update session interrupted`
Solution: A directory administrator has stopped or disabled replication.

`0 Incremental update session started`
Solution: Replication is proceeding normally.

`0 Incremental update session stopped: nothing to replicate`

Solution: Replication is proceeding normally.

`0 Incremental update session succeeded`

Solution: Replication is proceeding normally.

`0 Incremental update started`

Solution: Replication is proceeding normally

`0 Incremental update stopped : Nothing acquired`

Solution: Replication is proceeding normally. Another supplier is replicating to the consumer, and replication from this supplier will resume after that operation finishes.

`0 Incremental update succeeded`

Solution: Replication is proceeding normally.

`0 No replication sessions started since server startup`

Solution: No replication operation has been attempted yet and therefore no status is available. This is typically the case when restarting replication.

`0 Replica acquired successfully`

Solution: Replication is proceeding normally.

`0 Replication session successful`

Solution: Replication is proceeding normally.

`1 Replication error acquiring replica: replica busy`

Solution: Another supplier is replicating to the consumer, and replication from this supplier will resume after that operation finishes.

`11 Replication error acquiring replica: duplicate replica ID detected`

Solution: More than one master is using the same replica ID. Fix the configuration.

`12 Replication session aborted`

Solution: The consumer is disabled. Error `8194` should also appear in the `errors` log. Investigate the cause of the problem, fix it, and restart replication.

`2 Replication error acquiring replica: excessive clock skew.`

Solution: The time difference for clocks on different replicas is too big for replication to handle. Synchronize the system clocks.

`202 Incremental update session aborted: Timeout while waiting for change acknowledgement [`*hostname*`:`*port-number*`]`

Solution: Replication is proceeding normally. A timeout temporarily prevented replication from continuing.

`3 Replication error acquiring replica: permission denied`

Solution: The credentials are wrong for the replication manager bind DN. Fix the replication agreement.

`4 Replication error acquiring replica: decoding error`

Solution: A protocol error occurred.

`401 Incremental update session stopped: Could not parse update vector`

Solution: Replication is proceeding normally. A parse error temporarily prevented replication from continuing.

`401 Replication session failed, consumer replica needs to be initialized`

Solution: The database on the consumer has not been initialized. Either perform a total update on the consumer, or initialize the consumer with the same data as the supplier.

`402 Replication session failed, consumer replica has a different data version`

Solution: The database on the consumer has been initialized with different data from that of the supplier. Either perform a total update on the consumer, or initialize the consumer with the same data as the supplier.

`5 Replication error acquiring replica: unknown update protocol`

Solution: The consumer does not support the same replication protocol as the supplier.

`6 Replication error acquiring replica: no such replica`

Solution: The consumer is not configured for replication for the suffix to be replicated.

`7 Replication error acquiring replica: csn below purge point`

Solution: The replication change log has been purged and therefore no longer contains the changes necessary to update the consumer.

`8 Replication error acquiring replica: internal error`

Solution: The consumer failed to replay an replication operation. See the `errors` log on the consumer to determine what led to the problem.

`801 Incremental update session aborted : Unable to adjust the time between replicas`

Solution: The time difference for clocks on different replicas is too big for replication to handle. Synchronize the system clocks, perhaps using the network time protocol (NTP).

`810 Replication error acquiring replica: Supplier and consumer use the same replica ID`
Solution: More than one replica is using the same replica ID. Fix the configuration.

`820 Incremental update session stopped: unable to replicate schema`
Solution: Replication is proceeding normally, but schema could not be replicated. If this message is observed repeatedly, replicate schema to the consumer manually.

`829 Replication error acquiring replica: not able to use partial replication to read-write replica`
Solution: Only consumers can be partial replica. Fix the configuration.

`9 Replication error acquiring replica: replica released`
Solution: Replication is proceeding normally.

Examples

```
nsds5replicaLastUpdateStatus: 0 replica acquired successfully
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduledfor removal after this release

See Also replication(5dsconf)

Name

nsDS5ReplicaName – Unique replica identifier

Description

PROPERTY	VALUE
Entry DN	`cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	N/A
Default Value	N/A
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5Replica` entries.

This attribute value is allocated by the server when the replica is created. Reserved for internal use.

Examples

```
nsDS5ReplicaName: d2e14d02-600311da-80ace5db-c83e55ac
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name nsDS5ReplicaPort – Port number on which consumer listens

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	Any valid port number
Default Value	N/A
Syntax	Integer

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

Specifies the hostname for the port number on which the consumer replica listens. Do not modify this attribute after it has been set.

Examples

```
nsDS5ReplicaPort: 389
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also replication(5dsconf)

Name

nsDS5ReplicaPurgeDelay – Maximum time to keep tombstones

Description

PROPERTY	VALUE
Entry DN	`cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	`0` (keep forever) to max. integer seconds
Default Value	`604800` (one week)
Syntax	Integer

This attribute is part of replica configuration for `nsDS5Replica` entries.

When this attribute is set, the server deletes tombstone entries older than the number of seconds you specify. Tombstone entries are those entries that have been marked for deletion but not yet removed, and the associated replication state information. When setting this attribute, ensure that the purge delay is longer than the longest replication cycle in your replication policy to avoid incurring conflict resolution problems and divergence between replica.

Examples

```
nsDS5ReplicaPurgeDelay: 604800
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

nsDS5ReplicaReferral – Referrals for a replica

Description

PROPERTY	VALUE
Entry DN	`cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	Any valid LDAP URL
Default Value	N/A
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5Replica` entries.

This attribute should be set on a consumer only. When this multi-valued attribute is set, the server returns these referrals when a client attempts to update a read-only consumer.

If this attribute is not set, the read-only consumer refers clients to supplier servers on update.

Examples

```
nsDS5ReplicaReferral: ldap://master.example.com
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

nsDS5ReplicaRoot – Base DN for replication

Description

PROPERTY	VALUE
Entry DN	`cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	DN of replicated suffix
Default Value	None
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5Replica` entries.

The server replicates entries below the entry with this DN. This DN must correspond to the root DN of a replicated suffix. Once set, this attribute must not be modified.

Examples

```
nsDS5ReplicaRoot: dc=example,dc=com
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

nsDS5ReplicatedAttributeList – Attributes not to replicate

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	Any valid attribute types
Default Value	N/A
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

When this multi-valued attribute is set, the supplier does not replicate updates to the specified attribute types to the consumer.

Examples

```
nsDS5ReplicatedAttributeList: userPassword
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name nsds5ReplicaTimeout – Timeout for replication operations

Description

PROPERTY	VALUE
Entry DN	cn=*agreement name*,cn=replica,cn=*suffix name*,cn=mapping tree,cn=config
Valid Range	0 maximum integer (seconds)
Default Value	600 (seconds)
Syntax	Integer

This attribute is part of replica configuration for nsDS5ReplicationAgreement entries.

When set, this attribute causes the supplier to wait at most the specified number of seconds for a response from the consumer concerning a replication operation.

If you see Warning: timed out messages in the errors log file, then you should increase the value of this attribute. You can find out the amount of time the operation actually lasted by examining the access log on the consumer. You can then tune this attribute to optimize performance.

Examples

```
nsds5ReplicaTimeout: 1200
```

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also replication(5dsconf)

Name

nsDS5ReplicaTombstonePurgeInterval – Time interval between tombstone purge operations

Description

PROPERTY	VALUE
Entry DN	`cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	`0` to maximum integer (seconds)
Default Value	`300` (five minutes)
Syntax	Integer

This attribute is part of replica configuration for `nsDS5Replica` entries.

When this attribute is set, the server waits the number of seconds you specify after each operation to purge tombstone entries. Bear in mind that purge operations can be time consuming.

Examples

```
nsDS5ReplicaTombstonePurgeInterval: 300
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

nsDS5ReplicaTransportInfo – Transport used for replication

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	`LDAP` | `SSL`
Default Value	N/A
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

Specifies the type of transport used for replication. When this attribute is set to LDAP, standard LDAP connections are used. When this attribute is set to SSL, LDAPS connections are used. Do not modify this attribute after it has been set.

Examples

```
nsDS5ReplicaTransportInfo: SSL
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

nsDS5ReplicaType – Role of the replica

Description

PROPERTY	VALUE
Entry DN	`cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	`0	1	2	3`
Default Value	3 (master), 2 (consumer)
Syntax	Integer

This attribute is part of replica configuration for `nsDS5Replica` entries.

When this attribute is set, the server starts accepting client updates after waiting the number of seconds you specify.

Examples

```
nsDS5ReplicaType: 3
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

nsds5replicaUpdateInProgress – Indicate whether or not a replication schedule update is in progress

Description

PROPERTY	VALUE
Entry DN	`cn=`*Replication Agreement Name*`,cn=replica,cn=`*Suffix Name*`, cn=mapping tree,cn=config`
Valid Range	`true` | `false`
Default Value	None
Syntax	Boolean

This read-only attribute states whether or not a replication schedule update is in progress.

Examples

```
nsds5replicaUpdateInProgress: true
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Unstable

Name

nsDS5ReplicaUpdateSchedule – When replication is scheduled

Description

PROPERTY	VALUE
Entry DN	`cn=`*agreement name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	*hhmm-hhmm 0123456* | *
Default Value	*
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5ReplicationAgreement` entries.

When set to `*`, this attribute causes the supplier to replicate as necessary. This attribute can alternatively take multiple values of the form *hhmm*`-`*hhmm 0123456*, where the first element specifies the starting hour, then finishing hour in 24-hour time format, from `0000-2359`, and the second specifies which days, starting with Sunday (`0`) to Saturday (`6`).

Examples

The following setting specifies the supplier should replicate to the consumer between midnight and 4 am on Sundays, Tuesdays, Thursdays, and Saturdays:

```
nsDS5ReplicaUpdateSchedule: 0000-0400 0246
```

See Also

replication(5dsconf)

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name

nsDS5Task – Internal replication tasks

Description

PROPERTY	VALUE
Entry DN	`cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	N/A
Default Value	None
Syntax	DirectoryString

This attribute is part of replica configuration for `nsDS5Replica` entries.

This attribute is reserved for internal use.

Examples

```
nsDS5Task: CL2LDIF
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name nsFarmServerURL – LDAP URL for chaining farm server

Description The LDAP URL of the remote server. A *farm server* is contains data in one or more databases. This attribute can contain optional servers for failover, separated by spaces. For cascading chaining, this URL can point to another chained suffix.

Entry DN	`cn=`*chainedSuffix*`,cn=chaining database, cn=plugins,cn=config`
Valid Range	Any valid remote server LDAP URL.
Default Value	N/A
Syntax	DirectoryString

Example	`nsFarmServerURL:`
	`ldap://epdiote.example.com:alternate_server:3333`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nshoplimit – maximum hops for chaining

Description Specifies the maximum number of times a suffix is allowed to chain, that is, the number of times a request can be forwarded from one chained suffix to another.

Entry DN	`cn=`*chainedSuffix*`,cn=chaining database, cn=plugins,cn=config`
Valid Range	1 to an appropriate upper limit for your deployment.
Default Value	10
Syntax	Integer
Example	`nsHopLimit: 3`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name

nsIndexType – type of index

Description

This optional, multivalued attribute specifies the types of index used in Directory Server operations and the values of the attributes to be indexed. Each index type must be entered on a separate line.

Entry DN `cn=default indexes,cn=config,cn=ldbm database, cn=plugins,cn=config`

Valid Range

`approx`	Approximate, sounds alike, index
`browse`	Virtual list view index
`eq`	Equality index
`matching rule`	Matching rule index
`pres`	Presence index
`sub`	Substring index

Default Value Not applicable

Syntax DirectoryString

Example `nsindextype: eq`

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

dse.ldif(4)

Name nsLookthroughLimit – maximum number of entries checked during search

Description This performance-related attribute specifies the maximum number of entries that Directory Server checks when examining candidate entries in response to a search request. If you bind as Directory Manager, `unlimited` is set by default and overrides any other settings you may specify here.

Binder based resource limits work for this limit, which means that if a value for the operational attribute `nsLookThroughlimit` is present in the entry used to bind, the default limit is overridden. If you attempt to set a value that is not a number or is too big for a 64-bit signed integer, you receive an `LDAP_UNWILLING_TO_PERFORM` error message with additional error information explaining the problem.

Entry DN	`cn=config,cn=ldbm database,cn=plugins,cn=config`
Valid Range	-1 to the maximum number of entries, where -1 is unlimited
Default Value	`5000`
Syntax	Integer
Example	`nsLookthroughLimit: 5000`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsMatchingRule – collation order OID for international index

Description This optional, multivalued attribute specifies the collation order object identifier, OID, required for Directory Server to operate international indexing.

Entry DN	`cn=default indexes,cn=monitor,cn=ldbm database,`
	`cn=plugins,cn=config`

Valid Range	Any valid collation order object identifier (OID)
Default Value	None
Syntax	DirectoryString
Example	`nsMatchingRule: 1.3.6.1.4.1.42.2.27.9.4.23.1` (For Bulgarian)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsMaxResponseDelay – maximum delay for chained response

Description This error detection, performance related attribute specifies the maximum period of time it can take a remote server to respond to an LDAP operation request made by a chained suffix before an error is suspected. Once this delay period has been met, the chained suffix tests the connection with the remote server.

Entry DN	`cn=config,cn=chaining database,cn=plugins,cn=config`
Valid Range	Any valid delay period in seconds.
Default Value	60 seconds
Syntax	Integer
Example	`nsMaxResponseDelay: 60`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsMaxTestResponseDelay – maximum delay to test chaining response

Description This error detection, performance related attribute specifies the duration of the test issued by the chained suffix to check whether the remote server is responding. If a response from the remote server is not returned within this period, the chained suffix assumes the remote server is down and the connection is not used for subsequent operations.

Entry DN	`cn=config,cn=chaining database,cn=plugins,cn=config`
Valid Range	Any valid delay period in seconds.
Default Value	15 seconds
Syntax	Integer
Example	`nsMaxTestResponseDelay: 15`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsMultiplexorBindDN – bind DN for chaining multiplexor

Description DN of the administrative entry used to communicate with the remote server. The *multiplexor* is the server that contains the chained suffix and communicates with the farm server. This bind DN cannot be the Directory Manager. If this attribute is not specified, the chained suffix binds as anonymous.

Entry DN	`cn=`*chainedSuffix*`,cn=chaining database, cn=plugins,cn=config`
Valid Range	Not applicable
Default Value	DN of the multiplexor.
Syntax	DirectoryString
Example	`nsMultiplexorBindDN: cn=proxy manager`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name

nsMultiplexorCredentials – bind password for chaining multiplexor

Description

Password for the administrative user, in plain text. If no password is provided, users can bind as anonymous. The password is encrypted in the configuration file. Please note that the example below is what you *view* , *not* what you type.

Entry DN	`cn=`*chainedSuffix*`,cn=chaining database, cn=plugins,cn=config`
Valid Range	Any valid password (that is encrypted using the DES reversible password encryption schema.)
Default Value	Not applicable
Syntax	DirectoryString
Example	`nsMultiplexorCredentials: {DES} 9Eko69APCJfF`

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

dse.ldif(4)

Name nsOperationConnectionsLimit – maximum number of LDAP connections for chaining

Description Maximum number of LDAP connections the chained suffix establishes with the remote server.

Entry DN	`cn=default instance config,cn=chaining database,`
	`cn=plugins,cn=config`

Valid Range	1 to 20 connections
Default Value	`10`
Syntax	Integer
Example	`nsoperationconnectionslimit:10`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsProxiedAuthorization – disable proxy authorization for chaining

Description Reserved for advanced use only, this attribute permits you to disable proxied authorization. A value of `off` means that proxied authorization is disabled, and that all binds for chained operations are executed as the user specified in `nsMultiplexorBindDN`.

Entry DN	`cn=default instance config,cn=chaining database,`
	`cn=plugins,cn=config`

Valid Range	on | off
Default Value	`on`
Syntax	DirectoryString
Example	`nsproxiedauthorization: on`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsReferralOnScopedSearch – referrals for chained searches

Description Controls whether referrals are returned for searches with scope of one level or subtree. When `nsReferralOnScopedSearch` is set to `on`, Directory Server returning referrals for such searches, instead of chaining the searches, allowing clients that can handle referrals to access the appropriate directory directly.

Entry DN	`cn=default instance config,cn=chaining database,`
	`cn=plugins,cn=config`

Valid Range	on | off
Default Value	`off`
Syntax	DirectoryString
Example	`nsreferralonscopedsearch: off`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name

nsslapd-accesscontrol – Turn access control on and off

Description

PROPERTY	VALUE
Entry DN	`cn=config`
Valid Range	`on|off`
Default Value	`on`
Syntax	DirectoryString

This attribute turns access control on and off. If this attribute has a value `off`, any valid bind attempt including an anonymous bind results in full access to all information stored in Directory Server.

Note – Do not set this attribute to `off` unless you are told to do so by technical support personnel.

Examples

```
nsslapd-accesscontrol: on
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Unstable

Name

nsslapd-accesslog, nsslapd-auditlog, nsslapd-errorlog – Specify the path and filename of the access|audit|error log

Description

PROPERTY	VALUE
Entry DN	`cn=config`
Valid Range	Any valid file name
Default Value	See description that follows.
Syntax	DirectoryString

These attributes specify the path and filename of the log used to record each database access. Default values are:

install-path`/logs/access`
install-path`/logs/audit`
install-path`/logs/errors`

The following information is recorded in the access log file by default:

- IP address of the client machine that accessed the database
- Operations performed such as search, add, modify
- Result of the access such as the number of entries returned for a search

The information recorded in the error log depends on the error log level but typically contains at least the server startup and shutdown times, and the server port number.

For logging to be enabled, both of the following conditions must be met:

- The `nsslapd-*log` attribute must be set to a valid file name.
- The `nsslapd-*log-logging-enabled` attribute must be set to `on`.

Other configurations result in logging being disabled.

Examples

```
nsslapd-accesslog: /local/ds/logs/access
nsslapd-auditlog: /local/ds/logs/audit
nsslapd-errorlog: /local/ds/logs/errors
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Stable

Name

nsslapd-accesslog-level, nsslapd-auditlog-level, nsslapd-errorlog-level – Control what is logged to the access, audit, or error log

Description

PROPERTY	VALUE
Entry DN	`cn=config`
Valid Range	See the description that follows.
Default Value	`256`
Syntax	Integer

This attribute specifies the access log levels. Log levels can be added together to provide you with the exact type of logging you require. For example, 516 (4 + 512) obtains the internal access operation, entry access, and referral logging. The following levels are supported:

0	No access logging
4	Logging for internal access operations
256	Logging for access to an entry
512	Logging for access to an entry and referrals
131072	Precise timing of operation duration. This gives microsecond resolution for the elapsed time item in the access log.

Examples

```
nsslapd-accesslog-level: 256
```

Attributes

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Unstable

Name

nsslapd-accesslog-list, nsslapd-auditlog-list, nsslapd-errorlog-list – Provide a list of log files used in access, audit, or error log rotation.

Description

PROPERTY	VALUE
Entry DN	`cn=config`
Valid Range	Not applicable
Default Value	None
Syntax	DirectoryString

This attribute provides a list of log files used in access, audit, or error log rotation. This attribute is read only and cannot be set.

Examples

```
nsslapd-accesslog-list:accesslog2,accesslog3
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Unstable

Name

nsslapd-accesslog-logbuffering, nsslapd-auditlog-logbuffering, nsslapd-errorlog-logbuffering – Determines whether the server writes access log entries directly to disk

Description

PROPERTY	VALUE
Entry DN	`cn=config`
Valid Range	`on`|`off`
Default Value	`on`
Syntax	DirectoryString

When this attribute is set to `off`, the server writes all access log entries directly to disk.

Examples

```
nsslapd-accesslog-logbuffering: off
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Unstable

Name

nsslapd-accesslog-logexpirationtime, nsslapd-auditlog-logexpirationtime, nsslapd-errorlog-logexpirationtime – Specify the maximum age that an access, audit, or error log file is allowed to reach before it is deleted

Description

PROPERTY	VALUE
Entry DN	`cn=config`
Valid Range	1 to the maximum 32–bit integer value (`2147483647`)
Default Value	`1`
Syntax	Integer

Specifies the maximum age that an access, audit, or error log file is allowed to reach before it is deleted. This attribute supplies only the number of units. The units are provided by the `nsslapd-accesslog-logexpirationtimeunit`, `nsslapd-auditlog-logexpirationtimeunit`, or `nsslapd-errorlog-logexpirationtimeunit` attribute.

Examples

```
nsslapd-accesslog-logexpirationtime: 2
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Unstable

Name

nsslapd-accesslog-logexpirationtimeunit, nsslapd-auditlog-logexpirationtimeunit, nsslapd-errorlog-logexpirationtimeunit – Determines the unit of the log expiration time

Description

Property	Value
Entry DN	`cn=config`
Valid Range	`month` | `week` | `day`
Default Value	`month`
Syntax	DirectoryString

Specifies the unit for the `nsslapd-accesslog-logexpirationtime`, `nsslapd-auditlog-logexpirationtime`, or `nsslapd-errorlog-logexpirationtime` attribute. If the unit is unknown by the server, the log will never expire.

Examples

```
nsslapd-accesslog-logexpirationtimeunit: day
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Unstable

Name

nsslapd-accesslog-logging-enabled, nsslapd-auditlog-logging-enabled, nsslapd-errorlog-logging-enabled – Enable or disable access, audit, or error logging

Description

PROPERTY	VALUE
Entry DN	`cn=config`
Valid Range	`on|off`
Default Value	Access logging `on`, Audit logging `off`
Syntax	DirectoryString

This attribute is used in conjunction with the `nsslapd-accesslog`, `nsslapd-auditlog`, or `nsslapd-errorlog` attribute to disable and enable access, audit, or error logging. The `nsslapd-accesslog`, `nsslapd-auditlog`, or `nsslapd-errorlog` attributes specify the path and filename of the access, audit, and error logs, respectively.

For logging to be enabled, the `nsslapd-accesslog-logging-enabled`, `nsslapd-auditlog-logging-enabled`, or `nsslapd-errorlog-logging-enabled` attribute must be switched to `on`, and the respective `nsslapd-accesslog`, `nsslapd-auditlog`, or `nsslapd-errorlog` attribute must have a valid path and filename.

For information about the combinations of values for these attributes, see the `nsslapd-accesslog`, `nsslapd-auditlog`, or `nsslapd-errorlog` man page.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Unstable

Name

nsslapd-accesslog-logmaxdiskspace, nsslapd-auditlog-logmaxdiskspace, nsslapd-errorlog-logmaxdiskspace – Specify the maximum amount of disk space in megabytes that the access, audit or error logs are allowed to consume

Description

PROPERTY	VALUE
Entry DN	`cn=config`
Valid Range	`-1` | `1` to the maximum 32–bit integer value (`2147483647`). A value of `-1` means that the disk space allowed for the log is unlimited in size.
Default Value	Access logs `500`, audit logs and error logs `100`
Syntax	Integer

Specifies the maximum amount of disk space in megabytes that the access, audit or error logs are allowed to consume. If this value is exceeded, the oldest access, audit or error log is deleted.

When setting the maximum disk space, consider the total number of log files that can be created due to log file rotation. As there are three different log files – access log, audit log, and error log – maintained by Directory Server, each of which consume disk space. Compare these considerations to the total amount of disk space that you want to be used by the each log.

Examples

```
nsslapd-accesslog-logmaxdiskspace: 200
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Unstable

Name

nsslapd-accesslog-logminfreediskspace, nsslapd-auditlog-logminfreediskspace, nsslapd-errorlog-logminfreediskspace – Specify the minimum amount of free disk space in megabytes that is allowed before an access, audit, or error log is deleted

Description

PROPERTY	VALUE
Entry DN	`cn=config`
Valid Range	1 to the maximum 32–bit integer value (2147483647)
Default Value	5
Syntax	Integer

Specifies the minimum amount of free disk space in megabytes that is allowed before an access, audit, or error log is deleted. When the amount of free disk space falls below the value specified by this attribute, the oldest access, audit, or error log is deleted until enough disk space is freed to satisfy this attribute.

Examples

```
nsslapd-accesslog-logminfreediskspace: 4
```

Attributes

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Unstable

Name

nsslapd-accesslog-logrotationtime, nsslapd-auditlog-logrotationtime, nsslapd-errorlog-logrotationtime – Specify the time interval, the time of the day, and the minimum file size for rotation of the access log, audit log, or error log

Synopsis

```
nsslapd-accesslog-logrotationtime:
 {time-interval}  [time-of-day | *]  [min-file-size | *]
```

Description

PROPERTY	VALUE
Entry DN	`cn=config`
Valid Range	See description that follows.
Default Value	`1 * *`
Syntax	DirectoryString

This attribute specifies the time interval, the time of the day, and the minimum file size for access, audit, or error rotation. The unit of the time interval is specified by the `nsslapd-accesslog-logrotationtimeunit`, `nsslapd-auditlog-logrotationtimeunit`, or `nsslapd-errorlog-logrotationtimeunit` attribute.

The valid range for this attribute is as follows.

- *time-interval*

  Time interval at which the log is rotated. The unit of the time interval is given by the `nsslapd-auditlog-logrotationtimeunit` attribute.
- *time-of-day*

  Time of the day, on a 24-hour clock, at which the log is rotated.

  The value * means that no time of day is specified.
- *min-file-size*

  Minimum file size in kilobytes at which the log file is rotated. The log file is rotated if the file size is greater than the specified number of kilobytes.

  The value * means that no minimum file size is specified.

This attribute must be used in conjunction with the `nsslapd-accesslog-logrotationtimeunit`, `nsslapd-auditlog-logrotationtimeunit`, or `nsslapd-errorlog-logrotationtimeunit` attribute.

Examples

EXAMPLE 1 To Rotate the Access Log at 11:30 pm Every Day Regardless of the Size of the Log File

To specify when the access log is rotated, the `nsslapd-accesslog-logrotationtime` and the `nsslapd-accesslog-logrotationtimeunit` attributes must be set.

To rotate the log daily, the `nsslapd-accesslog-logrotationtimeunit` must be set as follows:

EXAMPLE 1 To Rotate the Access Log at 11:30 pm Every Day Regardless of the Size of the Log File *(Continued)*

```
nsslapd-accesslog-logrotationtimeunit: day
```

To rotate the log at 11:30 pm daily, the `nsslapd-accesslog-logrotationtime` must be set as follows:

```
nsslapd-accesslog-logrotationtime: 1 2330
```

The *min-file-size* is not specified, therefore,the log is rotated irrespective of the file size.

EXAMPLE 2 To Rotate the Error Log at 11:30 pm Every Day if the File Size is Greater Than 10 KB

To specify when the error log is rotated, the `nsslapd-errorlog-logrotationtime` and the `nsslapd-errorlog-logrotationtimeunit` attributes must be set.

To rotate the log daily, the `nsslapd-accesslog-logrotationtimeunit` must be set as follows:

```
nsslapd-accesslog-logrotationtimeunit: day
```

To rotate the log at 11:30 pm daily if the file size is greater than 10 KB, the `nsslapd-accesslog-logrotationtime` must be set as follows:

```
nsslapd-accesslog-logrotationtime: 1 2330 10
```

EXAMPLE 3 To Rotate the Error Log at Any Time if the File Size is Greater Than 10 KB

To specify when the error log is rotated, the `nsslapd-errorlog-logrotationtime` and the `nsslapd-errorlog-logrotationtimeunit` attributes must be set.

To rotate the log every minute if the file size is greater the 10 KB, the `nsslapd-errorlog-logrotationtimeunit` attribute must be set as follows:

```
nsslapd-accesslog-logrotationtimeunit: Minute
```

To rotate the log when the file size is greater than 10 KB , the `nsslapd-errorlog-logrotationtime` must be set as follows:

```
nsslapd-errorlog-logrotationtime: 1 * 10
```

The *time-of-day* is specified as `*`, therefore, no specific time of day is specified.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Unstable

Name

nsslapd-accesslog-logrotationtimeunit, nsslapd-auditlog-logrotationtimeunit, nsslapd-errorlog-logrotationtimeunit – Specify the unit for the `time-interval` part of the `nsslapd-access|audit|errorlog-logrotationtime` attribute

Description

PROPERTY	VALUE
Entry DN	`cn=config`
Valid Range	`Month` | `Week` | `Day` | `Hour` | `Minute`
Default Value	`Day`
Syntax	DirectoryString

Specifies the unit for the `time-interval` part of the `nsslapd-`*access*`|`*audit*`|`*error*`log-logrotationtime` attribute.

This attribute must be used in conjunction with the `nsslapd-`*access*`|`*audit*`|`*error*`log-logrotationtime` attribute.

Attributes

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Unstable

Name

nsslapd-accesslog-maxlogsize, nsslapd-auditlog-maxlogsize, nsslapd-errorlog-maxlogsize – Specify the maximum size of the access, audit, or error log in megabytes

Description

PROPERTY	VALUE
Entry DN	`cn=config`
Valid Range	`-1` | 1 to the maximum 32–bit integer value (`2147483647`). A value of `-1` means the log file is unlimited in size.
Default Value	`100`
Syntax	Integer

This attribute specifies the maximum size of the access, audit, or error log in megabytes. When this value is reached, the log is rotated.

If the `nsslapd-accesslog-maxlogsperdir`, `nsslapd-auditlog-maxlogsperdir`, or `nsslapd-errorlog-maxlogsperdir` attribute is set to `1`, the server ignores the respective `nsslapd-accesslog-maxlogsize`, `nsslapd-auditlog-maxlogsize`, or `nsslapd-errorlog-maxlogsize` attribute.

When you set a maximum log size, consider the total number of log files that can be created due to log file rotation. Also, remember that there are three different log files (access log, audit log, and error log) maintained by Directory Server, each of which will consume disk space. Compare these considerations to the total amount of disk space that you want to be used by the log.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Unstable

Name

nsslapd-accesslog-maxlogsperdir, nsslapd-auditlog-maxlogsperdir, nsslapd-errorlog-maxlogsperdir – Specify the total number of access, audit, or error logs that can be contained in the access, audit, or error logs directory

Description

PROPERTY	VALUE
Entry DN	`cn=config`
Valid Range	1 to the maximum 32–bit integer value (2147483647)
Default Value	`10`
Syntax	Integer

This attribute specifies the total number of access, audit, or error logs that can be contained in the respective directory.

Each time the access log is rotated, a new log file is created. When the number of files contained in the access log directory exceeds the value stored on this attribute, the oldest version of the log file is deleted. The same scenario is true for audit logs and error logs.

If you set this value to 1, the server will not rotate the log and it will grow indefinitely.

If the value for this attribute is higher than 1, check the `nsslapd-accesslog-logrotationtime`,`nsslapd-auditlog-logrotationtime`, or `nsslapd-errorlog-logrotationtime` attribute to establish whether or not log rotation is specified.

If the `nsslapd-accesslog-logrotationtime`, `nsslapd-auditlog-logrotationtime`, or `nsslapd-errorlog-logrotationtime` attribute has a value of `-1`, there is no rotation of the respective log.

For more information, refer to the `nsslapd-accesslog-logrotationtime`, `nsslapd-auditlog-logrotationtime`, or `nsslapd-errorlog-logrotationtime` man page.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Unstable

Name

nsslapd-accesslog-permissions, nsslapd-auditlog-permissions, nsslapd-errorlog-permissions – Specify the permissions for the log files

Synopsis

`nsslapd-accesslog-permissions:` *permissions*

Description

PROPERTY	VALUE
Entry DN	`cn=config`
Valid Range	See description that follows.
Default Value	`600`
Syntax	UNIX style octal permissions. Not supported on Windows platforms.

This attribute specifies the permissions for the log file.

Valid range is `001` to `777`.

The permission `000` is not allowed because the server might have trouble starting for non-root users.

On Windows platforms, this attribute is ignored and a warning message is logged.

Examples

EXAMPLE 1 To Set the Access Log Permssions to 644

```
nsslapd-accesslog-permissions:  644
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Unstable

Name nsslapd-allidsthreshold – Maximum number of values per index key in an index list

Description This attribute defines a threshold to limit the length of an index list. The threshold is called the *index list threshold*. If the number of entries in the list for a particular key exceeds the index list threshold, an unindexed search is performed.

The value of the `nsslapd-allidsthreshold` attribute can be configured globally for a Directory Server instance, or can be configured for a suffix, or can be configured for an index type. If the value of the `nsslapd-allidsthreshold` attribute is configured globally for a suffix, it can then be changed for a specific index.

You must rebuild all indexes after you change the `nsslapd-allidsthreshold` attribute.

Entry DN `cn=config,cn=ldbm database,cn=plugins,cn=config`

Valid Range

0	The `nsslapd-allidsthreshold` attribute is not used. The global configured value is used.
<2000	Values less than 2000 are rounded up to 2000.
>2000	The value is the minimum guaranteed value. Because of internal mechanisms, the real value can be slightly more than the specified value.
-1	No limit

Default Value `4000`

Syntax Integer

Error Messages `LDAP_UNWILLING_TO_PERFORM` means ou have set a value that is not a number or the value is too big for a 64-bit signed integer. Additional error information is provided to explain the problem.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-attribute-name-exceptions – allow non-standard characters in attribute names

Description Allows non-standard characters in attribute names to be used for backward compatibility with older servers.

Entry DN	`cn=config`
Valid Range	on | off
Default Value	`off`
Syntax	DirectoryString
Example	`nsslapd-attribute-name-exceptions: on`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-backend – suffix used to process requests

Description Gives the name of the suffix or chained suffix used to process requests. This attribute can be multivalued if you are using a custom distribution plug-in, with one suffix name per value. In this case, you must also specify the `nsslapd-distribution-plugin` and `nsslapd-distribution-funct` attributes.

Note – Use Directory Proxy Server, rather than a Directory Server plug-in, for distribution.

This attribute is required when the value of the `nsslapd-state` attribute is set to `backend`, or to `referral on update`.

Entry DN	`cn="`*suffixName*`",cn=mapping tree,cn=config`
Valid Range	Any valid partition name.
Default Value	None
Syntax	DirectoryString
Example	`nsslapd-backend: example`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-berbufsize – initial BER buffer size

Description This attribute defines the initial size in bytes of the buffer used to handle BER values.

In some cases where searches retrieving very large static groups, the default buffer size may cause slow performance due to the number of memory reallocation requests.

Entry DN `cn=config`

Valid Range 0 - `nsslapd-maxbersize`, in bytes

A value of 0 indicates that the default value should be used.

Default Value `1024`

Syntax Integer

Example `nsslapd-berbufsize: 16384`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-cachememsize – cache memory size

Description Specifies the entry cache size in terms of the available memory space. Limiting cache size in terms of memory occupied is the simplest method. If you attempt to set a value that is not an integer or is too big for a 64-bit unsigned integer, or a 32-bit unsigned integer for 32-bit installations, you receive an `LDAP_UNWILLING_TO_PERFORM` error message with additional error information explaining the problem.

Entry DN	`cn=`*dbName*`,cn=ldbm database,cn=plugins,cn=config`
Valid Range	200 KB to 2^{64}-1 Bytes (200 KB to 2^{32}-1 Bytes for 32-bit installations)
Default Value	`10 485 760` (10Mb)
Syntax	Integer
Example	`nsslapd-cachememsize:10`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-cachesize – cache size

Description Specifies the entry cache size in terms of the number of entries it can hold. Note that it is simpler to limit the cache by memory size only using the `nsslapd-cachememsize` attribute. If you attempt to set a value that is not an integer or is too big for a 64-bit unsigned integer, or a 32-bit unsigned integer for 32-bit installations, you receive an `LDAP_UNWILLING_TO_PERFORM` error message with additional error information explaining the problem.

Entry DN	`cn=`*dbName*`,cn=ldbm database,cn=plugins,cn=config`
Valid Range	1 to 2,147,483,647 (or -1 which means unlimited) entries
Default Value	`-1`
Syntax	Integer
Example	`nsslapd-cachesize: -1`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-certmap-basedn – certificate map search base

Description This attribute can be used when client authentication is performed using SSL certificates in order to avoid limitation of the security subsystem certificate mapping, configured in `certmap.conf`.

Depending on the `certmap.conf` configuration, the certificate mapping may be done using a directory subtree search based at the root DN. Note that if the search is based at the root DN, then the `nsslapd-certmap-basedn` attribute may force the search to be based at some entry other than the root.

Entry DN	`cn=config`
Valid Range	The DN of an entry in the directory
Default Value	`Not applicable`
Syntax	DN
Example	`nsslapd-certmap-basedn: ou=people,dc=example,dc=com`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-changelogdir – changelog path

Description This required attribute specifies the name of the directory in which the change log is created. Whenever a change log configuration entry is created it must contain a valid directory or the operation will be rejected.

Note – For performance reasons, it is recommended that you store this database on a different physical disk from other databases.

If you change this value after enabling replication, the old changelog is deleted and a new changelog is created. Therefore, you should not change the value of this attribute after replication has been enabled and consumers initialized.

Entry DNs	`cn=Retro Changelog Plugin,cn=plugins,cn=config`
Valid Range	Any valid path to the directory storing the change log
Default Value	None
Syntax	DirectoryString
Example	`nsslapd-changelogdir: /local/fastdisk/changelog`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-changelogmaxage – maximum changelog age

Description Specifies the maximum age of any entry in the change log. The change log contains a record for each directory modification and is used when synchronizing consumer servers. Each record contains a timestamp. Any record with a timestamp that is older than the value specified in this attribute will be removed. If this attribute is absent, there is no age limit on change log records.

Entry DNs	`cn=Retro Changelog Plugin,cn=plugins,cn=config`
Valid Range	0 (meaning that entries are not removed according to their age) to maximum integer (2147483647)
Default Value	`0`
Syntax	DirectoryString *IntegerTimeunit* Here, *Timeunit* is `s` for seconds, `m` for minutes, `h` for hours, `d` for days, `w` for weeks.
Example	`nsslapd-changelogmaxage: 30d`

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-changelogmaxentries – maximum number of changelog records

Description Specifies the maximum number of records the change log may contain. If this attribute is absent, there is no maximum number of records the change log can contain.

Entry DNs	`cn=Retro Changelog Plugin,cn=plugins,cn=config`
Valid Range	0 (no limit to the number of entries) to the maximum 32 bit integer value (2147483647).
Default Value	0
Syntax	Integer
Example	`nsslapd-changelogmaxentries: 0`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-config – DN of the server configuration

Description This read-only attribute is the configuration DN.

Entry DN	`cn=config`
Valid Range	Any valid config DN.
Default Value	`cn=config`
Syntax	DirectoryString
Example	`nsslapd-config: cn=config`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-dbcachesize – database cache size

Description This performance tuning related attribute specifies database cache size. Note that this is neither the index cache nor the entry cache. If you activate automatic cache resizing, you override this attribute, by replacing these values with its own guessed values at a later stage of the server startup.

If you attempt to set a value that is not a number or is too big for a 32-bit or 64-bit signed integer, you receive an `LDAP_UNWILLING_TO_PERFORM` error message with additional error information explaining the problem.

Note – The actual cache used may be significantly higher than what is specified in the `nsslapd-cachememsize` and `nsslapd-dbcachesize` attributes. It is therefore recommended that you do not specify a total cache size of more than 2 GB for 32-bit servers.

Changes to database cache size take effect after the server has been restarted.

Entry DN	`cn=config,cn=ldbm database,cn=plugins,cn=config`
Valid Range	500 KB to 4 GB for 32-bit platforms and 500 KB to 2^64-1 for 64-bit platforms
Default Value	`32 MB`
Syntax	Integer
Example	`nsslapd-dbcachesize: 100 MB`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-db-checkpoint-interval – database checkpoint interval

Description The amount of time in seconds after which Directory Server sends a checkpoint record to the database transaction log. The database transaction log contains a sequential listing of all recent database operations and is used for database recovery only. A checkpoint record indicates which database operations have been physically written to the directory database. The checkpoint records are used to determine where in the database transaction log to begin recovery after a system failure. The `nsslapd-db-checkpoint-interval` attribute is absent from `dse.ldif`. To change the checkpoint interval, you add the attribute to `dse.ldif`. This attribute can be dynamically modified using `ldapmodify`.

This attribute is provided only for system modification and diagnostics. It should be changed only with the guidance of Sun engineering staff and Sun Professional Services. Inconsistent settings of this attribute and other configuration attributes may cause Directory Server to be unstable.

Entry DN	`cn=config,cn=ldbm database,cn=plugins,cn=config`
Valid Range	10 to 300 seconds
Default Value	`60`
Syntax	Integer
Example	`nsslapd-db-checkpoint-interval: 120`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-db-circular-logging – circulate through transaction logs

Description Specifies circular logging for the transaction log files. If this attribute is switched off, old transaction log files are not removed, and are kept renamed as old log transaction files. Turning circular logging off can severely degrade server performance. It should therefore only be modified with the guidance of Sun Professional Services.

Entry DN	`cn=config,cn=ldbm database,cn=plugins,cn=config`
Valid Range	on | off
Default Value	`on`
Syntax	DirectoryString
Example	`nsslapd-db-circular-logging: on`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-db-durable-transactions – when to write transactions to disk

Description Indicates whether database transaction log entries are immediately written to the disk. The database transaction log contains a sequential listing of all recent database operations and is used for database recovery only.

With durable transactions enabled, every directory change is physically recorded in the log file and is therefore able to be recovered in the event of a system failure. However, the durable transactions feature may also slow down the performance of Directory Server. With durable transactions disabled, all transactions are logically written to the database transaction log but may not be physically written to disk immediately. If there is a system failure before a directory change is physically written to disk, that change is not recoverable.

Entry DN	`cn=config,cn=ldbm database,cn=plugins,cn=config`
Valid Range	on | off
Default Value	`on`
Syntax	DirectoryString
Example	`nsslapd-db-durable-transactions: on`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-db-home-directory – database file location

Description Used to fix a situation where the operating system endlessly flushes pages. This flushing can be so excessive that performance of the entire system is severely degraded.

This situation will occur only for certain combinations of the database cache size, the size of physical memory, and kernel tuning attributes. In particular, this situation should not occur if the database cache size is less than 100 MB.

For example, if your Solaris host seems excessively slow and your database cache size is around 100 MB or more, then you can use the `iostat` utility to diagnose the problem. Use `iostat` to monitor the activity of the disk where the Directory Server database files are stored. If all of the following conditions are true, then you can use the `nsslapd-db-home-directory` attribute to specify a subdirectory of a `tempfs` type file system.

- The disk is heavily used, more than 1 MB per second of data transfer.
- Service time is long, more than 100 ms.
- There is mostly write activity.

Note – The directory referenced by the `nsslapd-db-home-directory` attribute must be a subdirectory of a file system of type `tempfs`, such as `/tmp`.

If you have multiple Directory Server instances on the same machine, their `nsslapd-db-home-directory` attributes must be configured with different directories. Failure to do so will result in the databases for both directories becoming corrupted.

Finally, use of this attribute causes internal Directory Server database files to be moved to the directory referenced by the attribute. It is possible, but unlikely, that the server will no longer start after the files have been moved because not enough memory can be committed. This is a symptom of an overly large database cache size being configured for your server. If this happens, reduce the size of your database cache size to a value where the server will start again.

Entry DN	`cn=config,cn=ldbm database,cn=plugins,cn=config`
Valid Range	Any valid directory name in a `tempfs` file system, such as `/tmp`.
Default Value	Not applicable
Syntax	DirectoryString
Example	`nsslapd-db-home-directory: /tmp/ds1-db`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-db-idl-divisor – number of blocks per database page

Description Specifies the index block size in terms of the number of blocks per database page. The block size is calculated by dividing the database page size by the value of this attribute. A value of 1 makes the block size exactly equal to the page size. The default value of 0 sets the block size to the page size minus an estimated allowance for internal database overhead. Before modifying the value of this attribute export all databases to LDIF. Once the modification has been made, reload the databases from LDIF.

Entry DN	`cn=config,cn=ldbm database,cn=plugins,cn=config`
Valid Range	0 to 8
Default Value	`0`
Syntax	Integer
Example	`nsslapd-db-idl-divisor: 2`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-db-locks – number of database locks used

Description Specifies the number of locks that can be used by the database. Increase the value of this attribute if you observe the following error:

```
libdb: Lock table is out of available locks
```

The current number of locks being used, the number of locks configured, and the maximum number of locks reached during the life of the process can be checked using the attributes `nsslapd-db-current-locks`, `nsslapd-db-configured-locks`, and `nsslapd-db-max-locks` respectively, under the entry `cn=database,cn=monitor,cn=ldbm dababase,cn=plugins,cn=config`.

Entry DN	`cn=config,cn=ldbm database,cn=plugins,cn=config`
Valid Range	1 to maximum integer
Default Value	20000
Syntax	Integer
Example	`nsslapd-db-locks: 20000`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-db-logbuf-size – database log buffer size

Description Specifies the log information buffer size. Log information is stored in memory until the buffer fills up or the transaction commit forces the buffer to be written to disk. Larger buffer sizes can significantly increase throughput in the presence of highly concurrent applications, or transactions producing large amounts of data. The `nsslapd-db-logbuf-size` attribute is only valid if the `nsslapd-db-durable-transaction` attribute is set to `on`.

Note – You must be prepared to export all databases to LDIF, remove existing databases, and import all databases from LDIF when modifying this attribute.

Entry DN	`cn=config,cn=ldbm database,cn=plugins,cn=config`
Valid Range	0, 32768 to 2097152 bytes (limited by the transaction log file size, which is 10 MB by default) 0 is equivalent to 32768 bytes
Default Value	524288 for new instances
Syntax	Integer
Example	`nsslapd-db-logbuf-size: 524288`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-db-logdirectory – database transaction log directory

Description The path to the directory containing the database transaction log. The database transaction log contains a sequential listing of all recent database operations and is used for database recovery only. By default, the database transaction log is stored in the same directory as the directory entries themselves, *instance-path*`/db`.

For fault-tolerance and performance reasons, you can move this log file to another physical disk. The `nsslapd-db-logdirectory` attribute is absent from `dse.ldif`. To change the location of the database transaction log, add the attribute to `dse.ldif`.

Note – You must be prepared to export all databases to LDIF, remove existing databases, and re-import all databases from LDIF when modifying this attribute.

Entry DN	`cn=config,cn=ldbm database,cn=plugins,cn=config`
Valid Range	Any valid path and directory name.
Default Value	Not applicable
Syntax	DirectoryString
Example	`nsslapd-db-logdirectory: /logs/txnlog`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-db-logfile-size – maximum size of single database log file

Description Specifies the maximum size of a single file in the log in bytes. By default, or if the value is set to 0, a maximum size of 10 MB is used. The maximum size is an unsigned four byte value. The value of this attribute can have significant impact on performance, as it can be tuned to avoid extensive log switching in the event of heavy entries.

Entry DN	`cn=config,cn=ldbm database,cn=plugins,cn=config`
Valid Range	0 to unsigned four byte integer
Default Value	`10 (MB)`
Syntax	Integer
Example	`nsslapd-db-logfile-size: 10`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-dbncache – split database cache

Description This attribute allows you to split the `ldbm` cache into equally sized separate pieces of memory. It is possible to specify caches that are large enough so that they cannot be allocated contiguously on some architectures. For example, some releases of Solaris limit the amount of memory that may be allocated contiguously by a process. If `nsslapd-dbncache` is 0 or 1, the cache will be allocated contiguously in memory. If it is greater than 1, the cache will be broken up into `ncache` equally sized separate pieces of memory.

This attribute is provided only for system modification, and diagnostics. It should be changed only with the guidance of Sun Professional Services. Inconsistent settings of this attribute and other configuration attributes may cause Directory Server to be unstable.

Entry DN	`cn=config,cn=ldbm database,cn=plugins,cn=config`
Valid Range	Positive integer or 0
Default Value	`0`
Syntax	Integer
Example	`nsslapd-dbncache: 0`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-db-page-size – size of database pages in bytes

Description Specifies the size of the pages used to hold items in the database, in bytes. Valid page sizes are 512, 1024, 2048, 4096, 8192, 16384, 32768, and 65536 bytes. If the page size is not explicitly set, Directory Server defaults to a page size of 8192 bytes. Ideally, page size is set such that entries fit in database pages without wasting space. Whether it is possible to achieve an ideal database page size depends on your entries, and on whether they are subject to change in size.

Note – You must be prepared to export all databases to LDIF, remove existing databases, and import all databases from LDIF when modifying this attribute.

For ZFS file systems, you may find performance is best when the file system block size is set to equal the database page size. For example, try setting both to 32K or 64K.

To change the value of `nsslapd-db-page-size`, perform the following steps.

1. Shut down the Directory Server instance using the `dsadm` command.
2. Back up the Directory Server databases to LDIF using the `dsadm` command.
3. Remove all existing database files.

   These files include subdirectories of the database directory, database transaction logs under the `nsslapd-db-logdirectory` directory, and everything under the `nsslapd-db-home-directory` directory.
4. Change the value of the attribute in `dse.ldif`.
5. Restore the Directory Server databases from LDIF using the `dsadm` command.
6. Restart the Directory Server instance using the `dsadm` command.

Changing this default value can have significant performance impact. If the page size is too small, it results in extensive page splitting and copying, whereas if the page size is too large, it can waste disk space.

The following list describes the characteristics of this configuration attribute.

Entry DN	`cn=config,cn=ldbm database,cn=plugins,cn=config`
Valid Range	512 bytes to 64 KB
Default Value	`8192` (bytes)
Syntax	Integer
Example	`nsslapd-db-page-size: 8192`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-db-transaction-batch-val – number of database transactions to batch before commit

Description Specifies how many transactions are batched before being committed. You can use this attribute to improve update performance when full transaction durability is not required. This attribute can be dynamically modified using `ldapmodify`.

If you do not define this attribute or set it to a value of 0, transaction batching will be turned off and it will be impossible to make remote modifications to this attribute via LDAP. However, setting this attribute to a value greater than 0 causes the server to delay committing transactions until the number of queued transactions is equal to the attribute value. A value greater than 0 also allows you to modify this attribute remotely via LDAP. A value of 1 for this attribute allows you to modify the attribute setting remotely via LDAP, but results in no batching behavior. A value of 1 at server startup is therefore useful for maintaining normal durability, while also allowing transaction batching to be turned on and off remotely when desired. Bear in mind that the value you choose for this attribute may require you to modify the `nsslapd-db-logbuf-size` attribute to ensure sufficient log buffer size for accommodating your batched transactions. In practice, values should be positive numbers. Values larger than 100 bring few benefits.

Note – The `nsslapd-db-transaction-batch-val` attribute is only valid if the `nsslapd-db-durable-transaction` attribute is set to on.

Entry DN	`cn=config,cn=ldbm database,cn=plugins,cn=config`
Valid Range	-2147483647 to 2147483648
Default Value	`0` (meaning turned off)
Syntax	Integer
Example	`nsslapd-db-transaction-batch-val: 5`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name

nsslapd-db-tx-max – maximum concurrent database transactions

Description

Specifies the maximum number of concurrent transactions that can be handled by the database. Increase the value of this attribute if you observe the following error.

```
Serious Error---Failed in dblayer_txn_begin, err=12 (Not enough space)
```

Entry DN	`cn=config,cn=ldbm database,cn=plugins,cn=config`
Valid Range	1 to maximum integer
Default Value	200
Syntax	Integer
Example	`nsslapd-db-tx-max: 200`

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

dse.ldif(4)

Name nsslapd-directory – absolute path to database instance

Description Specifies the absolute path to the database instance. If the database instance is created manually, this attribute must be included. Once the database instance has been created, do not modify this path as any changes risk preventing the server from accessing data.

Entry DN	`cn=`*dbName*`,cn=ldbm database,cn=plugins,cn=config`
Valid Range	Any valid absolute path to the database instance.
Default Value	None
Syntax	DirectoryString
Example	`nsslapd-directory: /local/ds/db`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-disk-full-threshold – full disk threshold to limit database updates

Description When the minimum free space on the disk in MB. When the available free space on any one of the disks used by a database instance falls below the value specified by this attribute, no updates are permitted and the server returns an `LDAP_UNWILLING_TO_PERFORM` error. Updates are allowed again as soon as free space rises above the threshold.

Entry DN	`cn=config,cn=ldbm database,cn=plugins,cn=config`
Valid Range	0 to unsigned 4-byte integer
Default Value	`10`
Syntax	Integer
Example	`nsslapd-disk-full-threshold: 10`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-disk-low-threshold – low disk threshold to limit database updates

Description Specifies the “low” free space on the disk (in MB). When the available free space on any one of the disks used by a database instance falls below the value specified by this attribute, protocol updates on that instance are permitted only by Directory Manager.

Entry DN	`cn=config,cn=ldbm database,cn=plugins,cn=config`
Valid Range	0 to unsigned 4-byte integer
Default Value	`100`
Syntax	Integer
Example	`nsslapd-disk-low-threshold: 100`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-distribution-funct – distribution plug-in function

Description Specifies the name of your distribution function within the library named by `nsslapd-distribution-plugin`. This attribute is required along with `nsslapd-distribution-plugin` when you have specified more than one database in the `nsslapd-backend` attribute.

Note – Use Directory Proxy Server, rather than a Directory Server plug-in, for distribution.

Entry DN	`cn="`*suffixName*`",cn=mapping tree,cn=config`
Valid Range	The name of the distribution function.
Default Value	None
Syntax	DirectoryString
Example	`nsslapd-distribution-funct: alphaNumDistrib`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name

nsslapd-distribution-plugin – distribution plug-in library

Description

Specifies the full path and filename of the shared library for the custom distribution plug-in. This attribute is required along with `nsslapd-distribution-funct` when you have specified more than one suffix in the `nsslapd-backend` attribute.

Note – Use Directory Proxy Server, rather than a Directory Server plug-in, for distribution.

Entry DN	`cn="`*suffixName*`",cn=mapping tree,cn=config`
Valid Range	The full path and filename of the plug-in library.
Default Value	None
Syntax	DirectoryString
Example	`nsslapd-distribution-plugin: /custom/plugins/myDistrib.so`

Once you have distributed entries, you cannot redistribute them. The following restrictions apply.

- You cannot change your distribution function once you have deployed entry distribution.
- You cannot use the `ldapmodify` command to change an entry if that would cause them to be distributed into a different database.
- You cannot replicate databases that are distributed over multiple databases.

  Violating these restrictions prevents Directory Server from correctly locating and returning entries.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

dse.ldif(4)

Name nsslapd-dn-cachememsize – DN cache memory size

Description Specifies the DN cache size in terms of the available memory space.

Entry DN	`cn=config`
Valid Range	Integer >= 1048576 Bytes
Default Value	`10 485 760` (10Mb)
Syntax	Integer
Multi-valued	No
Read-write access	RW
Restart needed	Yes
Example	`nsslapd-dn-cachememsize:2097152` (2Mb)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-dn-cachesize – DN cache size

Description Specifies the DN cache size in terms of the number of entries it can hold. The value that can be assigned to `nsslapd-dn-cachesize` has the following impact on its behavior.

- `0` — means disabled
- `-1` — means not limited by number of DNs but limited by the size specified in `nsslapd-dn-cachememsize`
- `>=1` — cache is limited to the size that you specify here

Entry DN	`cn=config`
Valid Range	Integer >= 1, 0, or -1
Default Value	`-1`
Syntax	Integer
Multi-valued	No
Read Write access	RW
Restart Needed	Yes
Example	`nsslapd-dn-cachesize: -1`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-ds4-compatible-schema – allow 4.x style schema definitions

Description Makes the schema in `cn=schema` compatible with 4.x versions of Directory Server.

Note – When this attribute is set to `on`, Directory Server can read schema from 4.x configuration files, which use syntax for attribute types and object classes that differs from the standard syntax defined by RFC 2252. As a result, when this attribute is set to `on`, schema cannot be modified using administrative tools, but must instead be modified manually.

Entry DN	`cn=config`
Valid Range	on | off
Default Value	`off`
Syntax	DirectoryString
Example	`nsslapd-ds4-compatible-schema: off`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-enquote-sup-oc – enable superior object class enquoting

Description Controls whether the quoting in the `objectclasses` attributes contained in the `cn=schema` entry conforms to the quoting specified by internet draft RFC 2252. By default, Directory Server does not place single quotes around the superior object class identified on the `objectclasses` attributes contained in `cn=schema`. RFC 2252 indicates that this value should not be quoted.

That is, Directory Server publishes `objectclasses` attributes in the `cn=schema` entry as follows:

```
objectclasses: ( 2.5.6.6
 NAME ’person’
 DESC ’Standard ObjectClass’
 SUP ’top’
 MUST ( objectclass $ sn $ cn )
 MAY ( aci $ description $ seealso $ telephonenumber $ userpassword ) )
```

However, RFC 2252 indicates that this attribute should be published as follows:

```
objectclasses: ( 2.5.6.6
 NAME ’person’
 DESC ’Standard ObjectClass’
 SUP top
 MUST ( objectclass $ sn $ cn )
 MAY ( aci $ description $ seealso $ telephonenumber $ userpassword ) )
```

Notice the absence of single quotes around the word `top`.

Turning this attribute on means that some LDAP clients will no longer function, as they require the schema as defined in RFC 2252.

Turning this attribute off causes Directory Server to conform to RFC 2252, but doing so may interfere with some earlier LDAP clients.

Entry DN	`cn=config`
Valid Range	on | off
Default Value	`off`
Syntax	DirectoryString
Example	`nsslapd-enquote-sup-oc: off`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

dse.ldif(4)

Name nsslapd-exclude-from-export – attributes excluded during database export

Description Specifies a list of attributes that are excluded when the database is exported.

Entry DN	`cn=config,cn=ldbm database,cn=plugins,cn=config`
Valid Range	N/A
Default Value	`entrydn entryid dncomp parentid numSubordinates`
Syntax	DirectoryString
Example	`nsslapd-exclude-from-export: entrydn entryid`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-groupevalnestlevel – level of access control nesting for group evaluations

Description Specifies the number of levels of nesting that the access control system will perform for group evaluation.

Entry DN	`cn=config`
Valid Range	0 to the maximum 64-bit integer value
Default Value	0
Syntax	Integer
Example	`nsslapd-groupevalnestlevel: 5`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-idletimeout – idle timeout

Description Specifies the amount of time in seconds after which an idle LDAP client connection is closed by the server. A value of 0 indicates that the server will never close idle connections.

Entry DN	`cn=config`
Valid Range	0 to the maximum 32-bit integer value (2147483647)
Default Value	`0`
Syntax	Integer
Example	`nsslapd-IdleTimeout: 0`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-import-cachesize – database cache size for import

Description This performance tuning related attribute determines the size of the database cache used in the bulk import process. By setting this attribute value so that the maximum available system physical memory is used for the database cache during bulk importing, you can optimize bulk import speed. If you attempt to set a value that is not a number or is too big for a 32-bit signed integer, you receive an `LDAP_UNWILLING_TO_PERFORM` error message with additional error information explaining the problem.

Note – A cache is created for each load that occurs. For example, if the user sets the `nsslapd-import-cachesize` attribute to 1 GB, then 1 GB is used when loading one database, 2 GB is used when loading 2 databases, and so forth.

Ensure that you have sufficient physical memory to prevent swapping from occurring, as this results in performance degradation.

Entry DN	`cn=config,cn=ldbm database,cn=plugins,cn=config`
Valid Range	20 MB to 4 GB for 32-bit platforms and 20 MB to 2^64-1 for 64-bit platforms
Default Value	64 MB
Syntax	Integer
Example	`nsslapd-import-cachesize: 209715200`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-infolog-area – Specify the component for which logging information should be provided.

Description

PROPERTY	VALUE
Entry DN	`cn=config`
Valid Range	See description that follows.
Default Value	`0`
Syntax	Integer

Specifies the component for which logging information should be provided. Each component is identified as an area, whose value is a decimal translation of the hex values in `slapi-plugin.h`. The valid range includes the following values:

0 Default logging area, used for critical errors and other messages that are always written to the error log, for example server startup messages. Messages at this level are always included in the error log regardless of the `nsslapd-infolog-level` setting.

1 Trace function calls. Logs a message when the server enters and exits a function.

4 Search arguments processing.

8 Connection management

16 Print out packets sent/received

32 Search filter processing

64 Config file processing

128 Access control list processing

512 LDBM processing.

2048 Log LDIF entry parsing debugging

4096 Housekeeping thread debugging

8192 Replication debugging

32768 Database cache debugging.

65536 Server plug-in debugging. An entry is written to the log file when a server plug-in calls `slapi_log_info_ex()`.

The log area is additive. For example, to enable logging on search filter processing (32) and Config file processing (64), you would set this attribute to 96 (32+64).

Examples `nsslapd-infolog-area: 4096`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Unstable

Name

nsslapd-infolog-level – Specify the level of logging information that should be returned for the server component defined by the `nsslapd-infolog-area` attribute

Description

PROPERTY	VALUE
Entry DN	`cn=config`
Valid Range	`0` | `1`
Default Value	`0`
Syntax	Integer

Specifies the level of logging information that should be returned for the server component defined by the `nsslapd-infolog-area` attribute. A value of `0` means that only default logging information is returned for the selected area. Setting this attribute to `1` enables additional logging information to be returned for the selected area.

Examples

```
nsslapd-infolog-level: 1
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Unstable

Name nsslapd-instancedir – instance path

Description Specifies the full path to the directory where this server instance is installed, by default the *instance-path* given at installation time. Do *not* change this value after installation.

Entry DN	`cn=config`
Valid Range	Any valid file path.
Default Value	*instance-path*
Syntax	DirectoryString
Example	`nsslapd-instancedir: /local/ds`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-ioblocktimeout – IO block time out

Description Specifies the amount of time in milliseconds after which the connection to a stalled LDAP client is closed. An LDAP client is considered to be stalled when it has not made any I/O progress for read or write operations.

Entry DN	`cn=config`
Valid Range	0 to the maximum 32-bit integer value (2147483647)
Default Value	`1800000`
Syntax	Integer
Example	`nsslapd-ioblocktimeout: 1800000`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-lastmod – track modification time

Description Specifies whether Directory Server maintains the modification attributes for Directory Server entries. These attributes include the following.

- `modifiersname`, which is the distinguished name of the person who last modified the entry.
- `modifytimestamp`, which is the timestamp, in GMT format, for when the entry was last modified.
- `creatorsname`, which is the distinguished name of the person who initially created the entry.
- `createtimestamp`, which is the timestamp for when the entry was created in GMT format.

Entry DN	`cn=config`
Valid Range	on | off
Default Value	`on`
Syntax	DirectoryString
Example	`nsslapd-lastmod: off`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-listenBacklog – maximum number of pending connections

Description Allows you to configure the maximum number of pending connections on a socket used by Directory Server. This configuration value is passed as the *backlog* parameter to the `listen()` call on Solaris systems for example.

Entry DN	`cn=config`
Valid Range	`0` to the maximum `int` for the system
Default Value	`128`
Syntax	Integer
Example	`nsslapd-listen-backlog: 1024`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-listenhost – listen to IP address

Description Allows multiple Directory Server instances to run on a multi-homed machine, and makes it possible to limit listening to one or more interfaces of a multi-homed machine. Provide the host name or host names corresponding to the IP interface or interfaces you want to specify as values for this attribute. Directory Server responds only to requests sent to the interface or interfaces corresponding to the host name or host names specified. This prevents other programs from using the same port as Directory Server on the specified interfaces.

Entry DN	`cn=config`
Valid Range	Any host name or host names
Default Value	Not applicable
Syntax	DirectoryString
Example	`nsslapd-listenhost:` *host_name*

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name

nsslapd-localhost – local host system name

Description

This read-only attribute specifies the host system on which Directory Server runs.

Entry DN	`cn=config`
Valid Range	Any fully qualified hostname.
Default Value	Hostname of installed machine.
Syntax	DirectoryString
Example	`nsslapd-localhost: myServer.example.com`

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

dse.ldif(4)

Name nsslapd-localuser – local user name

Description Specifies the user under which Directory Server runs. The group under which the user runs is derived from this attribute, by examining the groups that the user is a member of. Should the user change, all the files in the installation directory must be owned by this user.

Entry DN	`cn=config`
Valid Range	Any valid user on the local system.
Default Value	To run as the same user who started Directory Server.
Syntax	DirectoryString
Example	`nsslapd-localuser: nobody`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-maxbersize – maximum message size

Description Defines the maximum size in bytes allowed for an incoming message. This limits the size of LDAP requests that can be handled by Directory Server. Limiting the size of requests prevents some kinds of denial of service attacks.

The limit applies to the total size of the LDAP request. For example, if the request is to add an entry, and the entry in the request is larger than two megabytes, then the add request is denied. Care should be taken when changing this attribute.

Entry DN	`cn=config`
Valid Range	0 - 2 GB (2,147,483,647 bytes), where a value of 0 indicates that the default value should be used.
Default Value	`2097152`
Syntax	Integer
Example	`nsslapd-maxbersize: 2097152`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-maxconnections – maximum number of connections

Description This attribute limits the number of simultaneous connections the server can manage. The value of this attribute is not set by default. If it is not set manually, its implicit value is the maximum number of file descriptors a process can open on the system.

You can use this attribute to limit the amount of memory used by Directory Server. Directory Server allocates `n*512` bytes of data, where n is equal to the value of `nsslapd-maxconnections`, if set, or to the maximum number of file descriptors a process can open on the system.

For example, on Solaris 9 systems, the maximum number of file descriptors is 64000. If `nsslapd-maxconnections` is not set, Directory Server allocates 35 MB of data, which may cause problems for some deployments. Setting `nsslapd-maxconnections` to a suitable value can help to alleviate this problem.

Entry DN `cn=config`

Valid Range `nsslapd-reservedescriptors` +1 to `maxdescriptors`.

If the `maxdescriptors` attribute is not set, the maximum value of `nsslapd-maxconnections` is the maximum number of file descriptors a process can open on the system.

Default Value `N/A`

Syntax Integer

Example `nsslapd-maxconnections: 4096`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name

nsslapd-maxdescriptors – maximum file descriptors

Description

This attribute sets the maximum, platform-dependent number of file descriptors that Directory Server will try to use. A file descriptor is used whenever a client connects to the server. It is also used for some server activities such as index maintenance. The number of available file descriptors for TCP/IP connections is the total for the `nsslapd-maxdescriptors` attribute minus the number of file descriptors used by the server for non-client connections, such as index management and managing replication, as specified in the `nsslapd-reservedescriptors` attribute.

The number that you specify here should not be greater than the total number of file descriptors that your operating system allows the `ns-slapd` process to use. This number will differ depending on your operating system. Some operating systems allow you to configure the number of file descriptors available to a process. Refer to your operating system documentation for details on file descriptor limits and configuration. It is worth noting that the included `idsktune` program can be used to suggest changes to the system kernel or TCP/IP tuning attributes, including increasing the number of file descriptors if necessary. You should consider increasing the value on this attribute if Directory Server is refusing connections because it is out of file descriptors. When this occurs, the following message is written to the Directory Server errors log file:

```
Not listening for new connections -- too many fds open
```

Note – UNIX shells usually have configurable limits on the number of file descriptors. Refer to your operating system documentation for further information regarding `limit` and `ulimit` as these limits can often cause problems.

This parameter is not applicable on Windows.

Entry DN	`cn=config`
Valid Range	1 to 65535
Default Value	Maximum number of file descriptors allowed for a process
Syntax	Integer
Example	`nsslapd-maxdescriptors: 8192`

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

dse.ldif(4)

Name nsslapd-maxpsearch – maximum number of persistent searches

Description Defines the maximum number of persistent searches that can be performed on Directory Server. The persistent search mechanism provides an active channel through which entries that change, and information about the changes that occur, can be communicated. Because each persistent search operation uses one thread, limiting the number of simultaneous persistent searches prevents certain kinds of denial of service attacks.

Entry DN	`cn=config`
Valid Range	1 to maximum thread number
Default Value	`30`
Syntax	Integer
Example	`nsslapd-maxpsearch: 30`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-maxthreadsperconn – maximum threads per connection

Description Defines the maximum number of threads that a connection should use. For normal operations where a client binds and performs only one or two operations before unbinding, you should use the default value. For situations where a client binds and simultaneously issues many requests, you should increase this value to allow each connection enough resources to perform all the operations.

Entry DN	`cn=config`
Valid Range	1 to maximum thread number
Default Value	`5`
Syntax	Integer
Example	`nsslapd-maxthreadsperconn: 5`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name

nsslapd-mode – database index file permissions

Description

Specifies the permissions used for newly created index files.

Entry DN `cn=config,cn=ldbm database,cn=plugins,cn=config`

Valid Range Any four-digit octal number. However, mode 0600 is recommended. This allows read and write access for the owner of the index files, which is the server user, and no access for other users.

Default Value `0600`

Syntax Integer

Example `nsslapd-mode: 0600`

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

dse.ldif(4)

Name nsslapd-nagle – delay sending responses

Description When the value of this attribute is `off`, the `TCP_NODELAY` option is set so that LDAP responses, such as entries or result messages, are sent back to a client immediately. When the attribute is turned on, default TCP behavior applies. That is, the sending of data is delayed, in the hope that this will enable additional data to be grouped into one packet of the underlying network MTU size, which is typically 1500 bytes for Ethernet.

Entry DN	`cn=config`
Valid range	on | off
Default value	`off`
Syntax	DirectoryString
Example	`nsslapd-nagle: off`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name

nsslapd-plugin, nsslapd-plugin-depends-on-named, nsslapd-plugin-depends-on-type, nsslapd-pluginDescription, nsslapd-pluginEnabled, nsslapd-pluginId, nsslapd-pluginInitfunc, nsslapd-pluginPath, nsslapd-pluginType, nsslapd-pluginVendor, nsslapd-pluginVersion – Directory Server plug-in legacy configuration

Description

The `nsslapd-plugin` attribute on `cn=config` is multi-valued, read-only attribute lists the syntaxes and matching rules loaded by the server. This manual page covers server plug-in configuration, rather than the `nsslapd-plugin` attribute.

This manual page provides an overview of legacy configuration information for server plug-ins. This manual page covers the individual plug-in configuration entry attributes. Also, this manual page covers the plug-ins provided with Directory Server, including configurable options, configurable arguments, default setting, dependencies, general performance related information, and further reading.

Note – In most circumstances, you configure plug-in functionality using the dsconf(1M) command. See plugin(5dsconf) for a list of configurable properties.

ATTRIBUTES FOR PLUG-IN CONFIGURATION ENTRIES

The following list covers each plug-in configuration entry attribute.

nsslapd-plugin-depends-on-named

This is a multivalued attribute, used to ensure that plug-ins are called by the server in the correct order. It takes a value that corresponds to the `cn` value of a plug-in. The plug-in whose `cn` value matches one of the values below it is started by the server prior to this plug-in. If the plug-in does not exist, the server fails to start.

Entry DN	`cn=`*pluginName*`,cn=plugins, cn=config`
Valid Range	Plug-in name
Default Value	None
Syntax	DirectoryString
Example	`nsslapd-plugin-depends-on-named: Class of Service`

nsslapd-plugin-depends-on-type

This is a multivalued attribute, used to ensure that plug-ins are called by the server in the correct order. It takes a value that corresponds to the type of a plug-in, contained in the attribute `nsslapd-pluginType`, and requires that plug-ins of that type are started before the present plug-in.

Entry DN	`cn=`*pluginName*`,cn=plugins, cn=config`

Valid Range	Plug-in type
Default Value	None
Syntax	DirectoryString
Example	`nsslapd-plugin-depends-on-type: database`

nsslapd-pluginDescription

Provides a description of the plug-in.

Entry DN	`cn=`*pluginName*`,cn=plugins,cn=config`
Valid Range	Any DirectoryString
Default Value	None
Syntax	DirectoryString
Example	`nsslapd-pluginDescription: acl access check plug-in`

`nsslapd-pluginEnabled`

Specifies whether or not the plug-in is enabled. This attribute can be changed over protocol, but will only take effect when the server is next restarted.

Entry DN	`cn=`*pluginName*`,cn=plugins,cn=config`
Valid Range	`on | off`
Default Value	`on`
Syntax	DirectoryString
Example	`nsslapd-pluginEnabled: on`

`nsslapd-pluginId`

Specifies the plug-in ID.

Entry DN	`cn=`*pluginName*`,cn=plugins,cn=config`
Valid Range	Any valid plug-in ID.
Default Value	None
Syntax	DirectoryString
Example	`nsslapd-pluginId: chaining database`

`nsslapd-pluginInitfunc`

Specifies the plug-in function to be initiated.

Entry DN	`cn=`*pluginName*`,cn=plugins,cn=config`
Valid Range	Any valid plug-in function.
Default Value	None

Syntax	DirectoryString
Example	`nsslapd-pluginInitfunc: NS7bitAttr_Init`

`nsslapd-pluginPath`

Specifies the full path to the plug-in.

Entry DN	`cn=`*pluginName*`,cn=plugins,cn=config`
Valid Range	Any valid path
Default Value	None
Syntax	DirectoryString
Example	`nsslapd-pluginPath: /opt/SUNWdsee/ds6/lib/uid-plugin.so`

`nsslapd-pluginType`

Specifies the plug-in type.

Entry DN	`cn=`*pluginName*`,cn=plugins,cn=config`
Valid Range	Any valid plug-in type.
Default Value	None
Syntax	DirectoryString
Example	`nsslapd-pluginType: preoperation`

nsslapd-pluginVendor

Specifies the vendor of the plug-in.

Entry DN	`cn=`*pluginName*`,cn=plugins,cn=config`
Valid Range	Any approved plug-in vendor.
Default Value	Sun Microsystems, Inc.
Syntax	DirectoryString
Example	`nsslapd-pluginVendor: Sun Microsystems, Inc.`

nsslapd-pluginVersion

Specifies the plug-in version.

Entry DN	`cn=`*pluginName*`,cn=plugins,cn=config`
Valid Range	Any valid plug-in version.
Default Value	Product version
Syntax	DirectoryString
Example	`nsslapd-pluginVersion: 6.0`

7-BIT CHECK PLUG-IN Consider the following aspects of this plug-in.

Plug-In Name	7-Bit Check (NS7bitAttr)
DN of Configuration Entry	`cn=7-bit check,cn=plugins,cn=config`
Description	Checks certain attributes are seven-bit clean.
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	List of attributes, `uid mail userpassword`, followed by a comma, and then by the suffix or suffixes on which the check is to occur.
Dependencies	None
Performance Related Information	None
Further Information	If your Directory Server uses non-ASCII characters such as Japanese and other languages for some attributes, remove those attributes from the list of attributes checked by this plug-in. When adding or modifying an attribute value checked by this plug-in, and the new value violates the seven-bit check, the client receives a `LDAP_CONSTRAINT_VIOLATION` (19) return code, and a message such as the following: `Value of attribute` *attr* `contains extended (8-bit) characters:` *value*

ACL PLUG-IN Consider the following aspects of this plug-in.

Plug-In Name	ACL Plugin
DN of Configuration Entry	`cn=ACL Plugin,cn=plugins,cn=config`
Description	ACL access check plug-in
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	None
Performance Related Information	Leave this plug-in running at all times.

ACL PREOPERATION PLUG-IN Consider the following aspects of this plug-in.

Plug-In Name	ACL preoperation

DN of Configuration Entry	`cn=ACL preoperation,cn=plugins,cn=config`
Description	ACL access check plug-in.
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	Database
Performance Related Information	Leave this plug-in running at all times.

BINARY SYNTAX PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	Binary Syntax
DN of Configuration Entry	`cn=Binary Syntax,cn=plugins,cn=config`
Description	Syntax for handling binary data.
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	None
Performance Related Information	Do not modify the configuration of this plug-in. Leave this plug-in running at all times.

BOOLEAN SYNTAX PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	Boolean Syntax
DN of Configuration Entry	`cn=Boolean Syntax,cn=plugins,cn=config`
Description	Syntax for handling booleans.
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	None
Performance Related Information	Do not modify the configuration of this plug-in. Leave this plug-in running at all times.

CASE EXACT STRING SYNTAX PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	Case Exact String Syntax

DN of Configuration Entry	`cn=Case Exact String Syntax,cn=plugins,cn=config`
Description	Syntax for handling case-sensitive strings.
Configurable Options	`on	off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	None
Performance Related Information	Do not modify the configuration of this plug-in. Leave this plug-in running at all times.

CASE IGNORE STRING SYNTAX PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	Case Ignore String Syntax
DN of Configuration Entry	`cn=Case Ignore String Syntax,cn=plugins,cn=config`
Description	Syntax for handling case-insensitive strings.
Configurable Options	`on	off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	None
Performance Related Information	Do not modify the configuration of this plug-in. Leave this plug-in running at all times.

CHAINING DATABASE PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	Chaining Database
DN of Configuration Entry	`cn=Chaining database,cn=plugins,cn=config`
Description	Syntax for handling DNs.
Configurable Options	`on	off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	None
Performance Related Information	Do not modify the configuration of this plug-in. Leave this plug-in running at all times.

CLASS OF SERVICE PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	Class of Service
DN of Configuration Entry	`cn=Class of Service,cn=plugins,cn=config`
Description	Allows for sharing of attributes between entries.
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	Set the `nsslapd-pluginarg0` attribute to: ■ `0` (default) to enable fast lookup of classic CoS templates ■ `1` to disable fast lookup for classic CoS template selection ■ `2` to disable checks for ambiguous pointer and classic CoS definitions Ambiguous definitions result when more than one value could be returned for the same attribute of the same entry. When checking remains enabled, Directory Server logs an informational message upon encountering such an ambiguity, provided you have set the log level to allow plug-ins to log informational messages. ■ `3` to disable both Restart Directory Server for modifications to take effect.
Dependencies	None
Performance Related Information	Leave this plug-in running at all times.

COUNTRY STRING SYNTAX PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	Country String Syntax
DN of Configuration Entry	`cn=Country String Syntax,cn=plugins,cn=config`
Description	Syntax for handling countries.
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	None

Dependencies	None
Performance Related Information	Do not modify the configuration of this plug-in. Leave this plug-in running at all times.

DISTINGUISHED NAME SYNTAX PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	Distinguished Name Syntax
DN of Configuration Entry	`cn=Distinguished Name Syntax,cn=plugins,cn=config`
Description	Syntax for handling DNs.
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	None
Performance Related Information	Do not modify the configuration of this plug-in. Leave this plug-in running at all times.

DSML FRONTEND SYNTAX PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	Frontend
DN of Configuration Entry	`cn=DSMLv2-SOAP-HTTP,cn=frontends,cn=plugins,cn=config`
Description	Enables you to access the directory using DSML v2 over SOAP/HTTP.
Configurable Options	`on | off`
Default Setting	`off`
Configurable Arguments	`ds-hdsml-soapschemalocation` `ds-hdsml-dsmlschemalocation`
Dependencies	None
Performance Related Information	None

GENERALIZED TIME SYNTAX PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	Generalized Time Syntax
DN of Configuration Entry	`cn=Generalized Time Syntax,cn=plugins,cn=config`

Description	Syntax for dealing with dates, times, and time zones.
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	None
Performance Related Information	Do not modify the configuration of this plug-in. Leave this plug-in running at all times.
Further Information	The Generalized Time String consists of the four digit year, two digit month (for example, 01 for January), two digit day, two digit hour, two digit minute, two digit second, an optional decimal part of a second and a time zone indication. We strongly recommend that you use the Z time zone indication (Greenwich Mean Time).

INTEGER SYNTAX PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	Integer Syntax
DN of Configuration Entry	`cn=Integer Syntax,cn=plugins,cn=config`
Description	Syntax for handling integers.
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	None
Performance Related Information	Do not modify the configuration of this plug-in. Leave this plug-in running at all times.

INTERNATIONALIZATION PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	Internationalization Plugin
DN of Configuration Entry	`cn=Internationalization Plugin,cn=plugins,cn=config`
Description	Syntax for handling DNs.
Configurable Options	`on | off`
Default Setting	`on`

Configurable Arguments	None. In contrast to previous versions of Directory Server, the collation orders and locales used by the internationalization plug-in are now stored in the configuration.
Dependencies	None
Performance Related Information	Do not modify the configuration of this plug-in. Leave this plug-in running at all times.

LDBM DATABASE PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	ldbm database plug-in
DN of Configuration Entry	`cn=ldbm database plug-in,cn=plugins,cn=config`
Description	Implements local databases.
Configurable Options	None
Default Setting	`on`
Configurable Arguments	None
Dependencies	None
Performance Related Information	Leave this plug-in running at all times.

MULTIMASTER REPLICATION PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	Multimaster Replication Plugin
DN of Configuration Entry	`cn=Multimaster Replication plugin,cn=plugins, cn=config`
Description	Enables replication between two Directory Server suffixes.
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	database
Performance Related Information	None
Further Information	You can turn this plug-in off if you have only one server, which will never replicate.

OCTET STRING SYNTAX PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	Octet String Syntax

DN of Configuration Entry	`cn=Octet String Syntax,cn=plugins,cn=config`
Description	Syntax for handling octet strings.
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	None
Performance Related Information	Do not modify the configuration of this plug-in. Leave this plug-in running at all times.

CLEAR PASSWORD STORAGE PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	CLEAR
DN of Configuration Entry	`cn=CLEAR,cn=Password Storage Schemes,cn=plugins, cn=config`
Description	CLEAR password storage scheme used for password encryption.
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	None
Performance Related Information	Do not modify the configuration of this plug-in. Leave this plug-in running at all times.

CRYPT PASSWORD STORAGE PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	CRYPT
DN of Configuration Entry	`cn=CRYPT,cn=Password Storage Schemes,cn=plugins, cn=config`
Description	CRYPT password storage scheme used for password encryption.
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	None

Performance Related Information	Do not modify the configuration of this plug-in. Leave this plug-in running at all times.

NS-MTA-MD5 PASSWORD STORAGE SCHEME PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	NS-MTA-MD5
DN of Configuration Entry	`cn=NS-MTA-MD5,cn=Password Storage Schemes, cn=plugins,cn=config`
Description	NS-MTA-MD5 password storage scheme for password encryption.
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	None
Performance Related Information	Do not modify the configuration of this plug-in. Leave this plug-in running at all times.
Further Information	You can no longer choose to encrypt passwords using the NS-MTA-MD5 password storage scheme. The storage scheme is still present, but for backward compatibility only. The data in your directory still contains passwords encrypted with the NS-MTA-MD5 password storage scheme.

RMCE PASSWORD STORAGE SCHEME PLUG-IN

This password storage scheme plug-in is used for example by the administration framework and is reserved for internal use.

SHA PASSWORD STORAGE SCHEME PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	SHA
DN of Configuration Entry	`cn=SHA,cn=Password Storage Schemes,cn=plugins, cn=config`
Description	SHA password storage scheme for password encryption.
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	None

Performance Related Information	If there are no passwords encrypted using the SHA password storage scheme, you may turn this plug-in off. If you want to encrypt your password with the SHA password storage scheme, choose SSHA instead. SSHA is a far more secure option.

SSHA PASSWORD STORAGE SCHEME PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	SSHA
DN of Configuration Entry	`cn=SSHA,cn=Password Storage Schemes,cn=plugins, cn=config`
Description	SSHA password storage scheme for password encryption.
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	None
Performance Related Information	Do not modify the configuration of this plug-in. Leave this plug-in running at all times.

STRONG PASSWORD CHECK PLUG-IN

When Directory Server is configured to check password quality, and this plug-in is enabled, the plug-in checks the following each time a password is added or modified.

- Clear text password values contain the classes of characters specified by the configuration.
- Clear text password values do not contain any sequence of four characters present in the dictionary file specified by the configuration.

Hashed password values such as `{SSHA}0Ri1g2yqlH3GTZcuRQ4uS22syCQLBKAU2ypLSw==` are not checked.

Consider the following aspects of this plug-in.

Plug-In Name	Strong Password Checking plug-in
DN of Configuration Entry	`cn=Strong Password Check,cn=plugins,cn=config`
Configurable options and arguments	`on | off` `nsslapd-pluginarg0`, which takes an integer representing a mask of values representing the character classes that must be present in a valid

password. Set `nsslapd-pluginarg0` to one of or a sum of the following values, not counting the special values `16` and `17`.

- `1` means the password must contain special characters.
- `2` means the password must contain numeric characters.
- `4` means the password must contain upper case characters.
- `8` means the password must contain lower case characters.
- `16` is a special value meaning at least three of the four character classes.
- `17` is a special value meaning at least two of the four character classes.

The default setting is `15`.

`nsslapd-pluginarg1`, which takes the absolute file system path to an ASCII dictionary file. If the argument is missing, the dictionary check is skipped. The plug-in does not initialize and Directory Server does not start if the value of this attribute is invalid or refers to an inaccessible file.

Default settings	`off`
Dependencies	Default password file, *install-path*`/ds6/plugins/words-english-big.txt`

POSTAL ADDRESS STRING SYNTAX PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	Postal Address Syntax
DN of Configuration Entry	`cn=Postal Address Syntax,cn=plugins,cn=config`
Description	Syntax used for handling postal addresses.
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	None
Performance Related Information	Do not modify the configuration of this plug-in. Leave this plug-in running at all times.

PTA PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	Pass Through Authentication

DN of Configuration Entry	`cn=Pass Through Authentication,cn=plugins,`
	`cn=config`

Description	Enables pass-through authentication, the mechanism that allows one directory to consult another to authenticate bind requests.
Configurable Options	`on | off`
Default Setting	`off`

Configurable Arguments	The LDAP URL to the configuration directory.
	`nsslapd-pluginarg0:`
	`ldap://config.example.com/o=example`

Dependencies	None

REFERENTIAL INTEGRITY POSTOPERATION PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	Referential Integrity Postoperation

DN of Configuration Entry	`cn=Referential Integrity Postoperation,`
	`cn=plugins,cn=config`

Description	Enables the server to ensure referential integrity. All attributes in all databases that are used by the referential integrity plug-in must be indexed. The indexes need to be created in the configuration of all the databases. When the retro change log is enabled, the `cn=changelog` suffix must be indexed.
Configurable Options	All configuration and `on | off`
Default Setting	`off`
Configurable Arguments	When enabled, the post operation Referential Integrity plug-in performs integrity updates on the `member`, `uniquemember`, `owner`, and `seeAlso` attributes immediately after a delete or rename operation. You can reconfigure the plug-in to perform integrity checks on all other attributes. The following arguments are configurable: 1. (`nsslapd-pluginarg0`) Check for referential integrity `-1` = no check for referential integrity

`0` = check for referential integrity is performed immediately

`positive integer` = request for referential integrity is queued and processed at a later stage. This positive integer serves as a wake-up call for the thread to process the request, at intervals corresponding to the integer specified.

2. (`nsslapd-pluginarg1`) Log file for storing the change, for example `/local/ds/logs/referint`
3. (`nsslapd-pluginarg2`) Reserved for future use.
4. (Other `nsslapd-pluginarg*` attributes) Attribute names to be checked for referential integrity.

Dependencies

`database` type

Tuning Recommendations

Do the following when you use the referential integrity plug-in in a multi-master replication environment:

- Enable the referential integrity plug-in on all servers containing master replicas
- Enable the referential integrity plug-in with the same configuration on every master

Set the first argument to a positive value, such as `10`, meaning ten seconds, to ensure that work performed by this plug-in happens asynchronously, rather than synchronously.

When enabling the plug-in, also create equality indexes for all attributes configured for use with the plug-in. The plug-in uses such indexes when searching for entries to update. Without equality indexes for the attributes it uses, the plug-in must perform costly unindexed searches that have negative impact on performance.

RETRO CHANGE LOG PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name

Retro Changelog Plugin

DN of Configuration Entry

`cn=Retro Changelog Plugin,cn=plugins,cn=config`

Description

Used by LDAP clients for maintaining application compatibility with Directory Server 4.x versions.

Maintains a log of all changes occurring in Directory Server. The retro change log offers the same functionality as the changelog in the 4.x versions of Directory Server.

Configurable Options `on | off`

Default Setting `off`

Configurable Arguments

The following arguments can be configured for the retro change log plug-in:

- `nsslapd-pluginarg0: -ignore_attributes`configures the retro change log plug-in to ignore attributes specified by the following `nsslapd-pluginarg`. This argument is configured by default.
- `nsslapd-pluginarg1: copyingFrom`specifies a list of attributes to be ignored by the preceding `nsslapd-pluginarg`. This argument is configured by default.
- `nsslapd-pluginarg2: suffixes="suffix1","suffix2"` configures the retro change log to record updates to specified suffixes only
- `nsslapd-pluginarg3: deletedEntryAttributes=attribute1,attribute2` configures the retro change log to record specified attributes of an entry when that entry is deleted

Dependencies None

Performance Related Information May slow down Directory Server performance.

Example Configuration

```
dn: cn=Retro Changelog Plugin,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: ds-signedPlugin
objectClass: extensibleObject
cn: Retro Changelog Plugin
nsslapd-pluginPath: /opt/SUNWdsee/ds6/lib/retrocl-plugin.so
nsslapd-pluginInitfunc: retrocl_plugin_init
nsslapd-pluginType: object
nsslapd-plugin-depends-on-type: database
nsslapd-pluginarg0: -ignore_attributes
nsslapd-pluginarg1: copyingFrom
nsslapd-pluginarg2: suffixes="ou=people","dc=example","dc=com"
nsslapd-pluginarg3: deletedEntryAtrributes="objectclass","employeenumber"
nsslapd-changelogdir: /local/ds/db/changelog
nsslapd-pluginEnabled: on
nsslapd-pluginId: retrocl
```

```
nsslapd-pluginVersion: 6.0
nsslapd-pluginVendor: Sun Microsystems, Inc.
nsslapd-pluginDescription: Retrocl Plugin
ds-pluginSignatureState: valid signature
```

ROLES PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	Roles Plugin
DN of Configuration Entry	`cn=Roles Plugin,cn=plugins,cn=config`
Description	Enables the use of roles in Directory Server.
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	State Change Plugin
Performance Related Information	Do not modify the configuration of this plug-in. Leave this plug-in running at all times.

STATE CHANGE PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	State Change Plugin
DN of Configuration Entry	`cn=State Change Plugin,cn=plugins,cn=config`
Description	State change notification service plug-in for detecting updates, such as configuration changes, and triggering callbacks when updates happen. This plug-in is used internally by the roles plug-in.
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	None

SUBTREE ENTRY COUNTER PLUG-INS

Consider the following aspects of this plug-in.

Plug-In Name	Subtree Entry Counter For *ObjectClass*
DN of Configuration Entry	`cn=Subtree Entry Counter for` *ObjectClass*`,cn=plugins, cn=config`
Description	Maintain a count of entries with a particular object class. The following plug-ins are provided.

- Subtree entry counter for departments in domains
- Subtree entry counter for domains within a domain
- Subtree entry counter for mail lists
- Subtree entry counter for nested departments
- Subtree entry counter for total domains
- Subtree entry counter for users

Configurable Options	`on | off`
Default Setting	`off`
Configurable Arguments	None
Dependencies	None
Performance Related Information	These plug-ins are provided for use with Messaging Server only, and are disabled by default. Leave these plug-ins disabled unless your Messaging Server requires them.

Counter Attributes Maintained

`nsNumDepts`	Either the number of departments within a domain, or the number of departments within a department (nested departments), depending on the DN of the entry.
`nsNumDomains`	Either the number of total domains, or the number of domains within a domain or nested domain, depending on the DN of the entry.
`nsNumMailLists`	Number of mail lists.

TELEPHONE SYNTAX PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	Telephone Syntax
DN of Configuration Entry	`cn=Telephone Syntax,cn=plugins,cn=config`
Description	Syntax for handling telephone numbers.
Configurable Options	`on | off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	None

Performance Related Information	Do not modify the configuration of this plug-in. Leave this plug-in running at all times.

UID UNIQUENESS PLUG-IN

Consider the following aspects of this plug-in.

Plug-In Name	UID Uniqueness
DN of Configuration Entry	`cn=UID Uniqueness,cn=plugins,cn=config`
Description	Checks that the values of specified attributes are unique each time a modification occurs on an entry.
Configurable Options	`on | off`
Default Setting	`off`
Configurable Arguments	You may configure this plug-in in either of two different ways.

1. Specify attributes that must be unique for a series of one or more subtrees identified by DNs. For example, to specify that `employeeNumber` and `uid` attribute values must be unique across both `o=org1,dc=example,dc=com` and `o=org2,dc=example,dc=com` , configure the arguments in the configuration entry as follows:

```
nsslapd-pluginarg0: employeeNumber
nsslapd-pluginarg1: uid
nsslapd-pluginarg2: o=org1,dc=example,dc=com
nsslapd-pluginarg3: o=org2,dc=example,dc=com
```

2. You specify attributes that must be unique inside congruent subtrees, optionally only on entries of a specified object class. For example, to specify that `employeeNumber` and `uid` attribute values must be unique in either `o=org1,dc=example,dc=com` or `o=org2,dc=example,dc=com`, but only on entries of the `inetOrgPerson` object class, configure the arguments in the configuration entry as follows:

```
nsslapd-pluginarg0: employeeNumber
nsslapd-pluginarg1: uid
nsslapd-pluginarg2: MarkerObjectClass="organization"
 RequiredObjectClass="inetOrgPerson"
```

Dependencies	`database` type

Performance Related Information	Directory Server provides the UID Uniqueness plug-in by default. To ensure unique values for other attributes, you can create instances of the UID Uniqueness plug-in for those attributes. The UID Uniqueness plug-in may slow down Directory Server performance.

URI PLUG-IN Consider the following aspects of this plug-in.

Plug-In Name	URI Syntax
DN of Configuration Entry	`cn=URI Syntax,cn=plugins,cn=config`
Description	Syntax for handling URIs (Unique Resource Identifiers) including URLs (Unique Resource Locators.)
Configurable Options	`on	off`
Default Setting	`on`
Configurable Arguments	None
Dependencies	None
Performance Related Information	Do not modify the configuration of this plug-in. Leave this plug-in running at all times.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

dse.ldif(4)

Name nsslapd-port – LDAP port number

Description TCP/IP port number used for LDAP communications. If you want to run SSL/TLS over this port, you can do so through the Start TLS extended operation. This selected port must be unique on the host system; make sure no other application is attempting to use the same port number. Specifying a port number of less than 1024 requires Directory Server to run as super user.

Note – Be aware when changing this port number of other applications whose configurations you may have to modify to reflect the change.

You must restart the server for the port number change to be taken into account.

Entry DN	`cn=config`
Valid Range	1 to 65535
Default Value	`389`
Syntax	Integer
Example	`nsslapd-port: 389`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-privatenamespaces – private naming contexts

Description Contains the list of the private naming contexts `cn=config`, `cn=schema`, and `cn=monitor`.

Entry DN	`cn=config`
Valid Range	`cn=config`, `cn=schema`, and `cn=monitor`
Default Value	`N/A`
Syntax	DirectoryString
Example	`nsslapd-privatenamespaces: cn=config`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name

nsslapd-pwdgeneratorpwdlen – Generated password length

Description

PROPERTY	VALUE
Entry DN	cn=config
Valid Range	6 to 512
Default Value	6
Syntax	Integer

This attribute specifies the length of the password generated by Directory Server when a password is reset using the LDAP Password Modify Extended Operation defined in RFC 3062 and no new password value is specified.

Examples

```
nsslapd-pwdgeneratorpwdlen: 8
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name nsslapd-readonly – read only mode

Description Specifies whether the whole server, or an individual database, is in read-only mode, meaning that neither data in a database nor configuration information can be modified. Any attempt to modify a database in read-only mode returns an error indicating that the server is unwilling to perform the operation.

Entry DN	`cn=config`
	`cn=`*dbName*`,cn=ldbm database,cn=plugins,cn=config`

Valid Range	on | off
Default Value	`off`
Syntax	DirectoryString
Example	`nsslapd-readonly: off`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-referral – referral

Description This multivalued attribute specifies the LDAP URL or URLs to be returned by the suffix, when the server receives a request for an entry not belonging to the local tree, that is, an entry whose suffix does not match the value specified on any of the suffix attributes. For example, suppose the database contains only the entries under the following DN.

```
ou=People, dc=example,dc=com
```

Yet, the request is for an entry under the following DN.

```
ou=Groups, dc=example,dc=com
```

In this case, the referral is returned so the client may contact the corresponding directory for the requested entry. Although only one referral is allowed per Directory Server instance, this referral can have multiple values.

Note – If you want to use SSL and TLS communications, the referral attribute should be of the following form.

`ldaps://`*hostname*

Start TLS does not support referrals.

For suffix configuration entries, this attribute is required when the value of the `nsslapd-state` attribute is set to `referral`.

Entry DNs	`cn=config`
	`cn="`*suffixName*`",cn=mapping tree,cn=config`

Valid Range	Valid LDAP URL in the following format: `ldap://` *hostname*
Default Value	Not applicable
Syntax	DirectoryString
Example	`nsslapd-referral: ldap://alternate.example.com`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-referralmode – referral mode

Description When set, this attribute causes the server send back the referral for *any* request on *any* suffix.

Entry DN	`cn=config`
Valid Range	Valid LDAP URL in the following format: `ldap://` *serverHost*
Default Value	Not applicable
Syntax	DirectoryString
Example	`nsslapd-referralmode: ldap://backup.example.com`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name

nsslapd-require-index – allow only indexed searches

Description

When switched to `on`, this attribute allows you to refuse unindexed searches. This performance related attribute avoids saturating the server with erroneous searches.

Entry DN	`cn=`*dbname*`,cn=ldbm database,cn=plugins,cn=config`
Valid Range	on | off
Default Value	`off`
Syntax	DirectoryString
Example	`nsslapd-require-index: off`

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

dse.ldif(4)

Name

nsslapd-reservedescriptors – reserve file descriptors

Description

This read-only attribute specifies the number of file descriptors that Directory Server reserves for managing non-client connections, such as index management and managing replication. The number of file descriptors that the server reserves for this purpose subtracts from the total number of file descriptors available for servicing LDAP client connections.

Most installations of Directory Server should never need to change this attribute. However, consider increasing the value on this attribute if all of the following are true:

- The server is replicating to a large number of consumer servers (more than 10), or the server is maintaining a large number of index files (more than 30).
- The server is servicing a large number of LDAP connections.
- You get error messages reporting that the server is unable to open file descriptors (the actual error message will differ depending on the operation that the server is attempting to perform), but these error messages are NOT related to managing client LDAP connections.

  Increasing the value on this attribute may result in more LDAP clients being unable to access your directory. Therefore, when you increase the value on this attribute, increase the value on the `nsslapd-maxdescriptors` attribute also. Note that you may not be able to increase the `nsslapd-maxdescriptors` value if your server is already using the maximum number of file descriptors that your operating system allows a process to use. Refer to your operating system documentation for details. If this is the case, then reduce the load on your server by causing LDAP clients to search alternative directory replicas.

  To assist you in computing the number of file descriptors you set for this attribute, use the following formula:

```
nsslapd-reservedescriptor =
20 +
(NumBackends * 4) +
NumGlobalIndexes  +
ReplicationDescriptors +
ChainingBackendDescriptors +
PTADescriptors +
SSLDescriptors
```

  The terms in the formula are as follows.

NumldbmBackends	Number of LDBM databases.
NumGlobalIndexes	Total number of configured indexes for all databases including system indexes. By default, there are 8 system indexes and 17 additional indexes per database.
ReplicationDescriptors	*NumSupplierReplicas* + 8

Where *NumSupplierReplicas* is number of replicas in the server that can act as a supplier (hub or master).

ChainingBackendDescriptors

NumChainingBackends * `nsOperationConnectionsLimit`

Where `nsOperationConnectionsLimit` is defined in the chained suffix configuration and 10 by default.

PTADescriptors

3 if PTA is configured, 0 if PTA is not configured.

SSLDescriptors

5 (4 files + 1 listen socket) if SSL is configured, 0 if SSL is not configured.

Entry DN	`cn=config`
Valid Range	1 to 65535
Default Value	`64`
Syntax	Integer
Example	`nsslapd-reservedescriptors: 64`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-return-exact-case – return exact case

Description Returns the exact case of attribute names, as defined in the schema.

Attribute names are case-insensitive by default. However, when an attribute is returned by Directory Server. as the result of a search operation, some client applications require attribute names to match the case of the attribute as it is listed in the schema. Other client applications require attribute names to be returned in lower case.

`nsslapd-return-exact-case` is enabled by default. You should disable this attribute if you have legacy clients that expect attribute names to be returned in lower case for backward compatibility with Directory Server 4.x. You must stop and restart the server for changes to this attribute to be taken into account.

Note that if the attribute name is specified in the search, it is returned in the case in which it is specified, regardless of the value of `nsslapd-return-exact-case`.

For example, the following search command:

```
ldapsearch -b "cn=config" -s base objectclass=* "PassWordMinAGe"
```

Returns the attribute as `PassWordMinAGe=0`, whether `nsslapd-return-exact-case` is set to `on` or `off`.

If `nsslapd-return-exact-case` is set to `on`, the following search command:

```
ldapsearch -b "cn=config" -s base objectclass=*
```

Returns the attribute as `passwordMinAge=0`, which is how this attribute is defined in the schema.

If `nsslapd-return-exact-case` is set to `off`, the same search command:

```
ldapsearch -b "cn=config" -s base objectclass=*
```

Returns the attribute as `passwordminage=0`, in lower case.

Entry DN	`cn=config`
Valid Range	on | off
Default Value	`on`
Syntax	DirectoryString
Example	`nsslapd-return-exact-case: on`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

dse.ldif(4)

Name nsslapd-rootdn – Directory Manager DN

Description Specifies the distinguished name of an entry that is not subject to access control restrictions, administrative limit restrictions for operations on the directory or resource limits in general. The attributes `nsslapd-sizelimit`, `nsslapd-timelimit`, and `nsslapd-schemacheck` do not apply to this DN either. `nsslapd-idletimeout` does however apply to connections opened by this DN.

Entry DN	`cn=config`
Valid Range	Any valid distinguished name
Default Value	`cn=Directory Manager`
Syntax	DN
Example	`nsslapd-rootdn: cn=Directory Manager`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-rootpw – Directory Manager password

Description Allows you to specify the password associated with the Directory Manager DN. When you provide the root password, it will be encrypted according to the encryption method you selected for `nsslapd-rootpwstoragescheme`.

When viewed from the `dse.ldif` file, this attribute shows the encryption method followed by the encrypted string of the password.

Note – If you configure a root DN at server installation time, you must also provide a root password. However, it is possible for the root password to be deleted from `dse.ldif` by direct editing of the file. In this situation, the root DN can only obtain the same access to your directory as you allow for anonymous access. Always make sure that a root password is defined in `dse.ldif` when a root DN is configured for your database.

Entry DN	`cn=config`
Valid Range	Any valid password encrypted by any one of the encryption methods described in passwordStorageScheme(5dsat).
Default Value	Not applicable
Syntax	DirectoryString: `{`*encryption_method*`}`*encrypted_password*
Example	`nsslapd-rootpw: {SSHA}fp+C/eJCYVV0ZlXDE52Pd9uOzjvJVk4B10biAg==`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-rootpwstoragescheme – root password storage scheme

Description This attribute indicates the encryption method used for the root password.

Entry DN	`cn=config`
Valid Range	Any encryption method described in passwordStorageScheme(5dsat)
Default Value	`SSHA`
Syntax	DirectoryString
Example	`nsslapd-rootpwstoragescheme: SSHA`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-schemacheck – schema checking

Description Specifies whether the database schema will be enforced during entry insertion or modification. When this attribute has a value of `on`, Directory Server will not check the schema of existing entries until they are modified. The database schema defines the type of information allowed in the database. You can extend the default schema using the `objectclasses` and attribute types.

Note – Schema checking works by default when database modifications are made using an LDAP client, such as `ldapmodify`, the Directory Server console, or when importing a database from LDIF.

If you turn schema checking off, you will have to verify manually that your entries conform to the schema. If schema checking is turned on, the server sends an error message to inform you of the entries that do not match the schema. Make sure that the attributes and object classes you create in your LDIF statements are both spelled correctly and identified in `dse.ldif.` You will need to create a file in LDIF format in the schema directory or add the elements to `99user.ldif`.

Entry DN	`cn=config`
Valid Range	on | off
Default Value	`on`
Syntax	DirectoryString
Example	`nsslapd-schemacheck: on`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-schema-repl-useronly – replicate only user-defined schema elements

Description This attribute allows you to have greater control over the schema that is replicated. The attribute is `off` by default, implying that the entire schema is replicated. If the attribute is set to `on`, only schema with an X-ORIGIN of `user-defined` is replicated. This setting greatly improves the performance of schema replication.

If you are replicating from a current Directory Server to a 5.1 server, you *must* set this attribute to `on`. Otherwise the current schema will be pushed to the 5.1 server and the 5.1 server will be unable to restart, due to duplicate objects.

Entry DN	`cn=config`
Valid Range	on | off
Default Value	`off`
Syntax	DirectoryString
Example	`nsslapd-schema-repl-useronly: off`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name

nsslapd-search-tune – Skip check for modifications during search before returning entries

Description

PROPERTY	VALUE
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	See description that follows.
Default Value	Not defined
Syntax	Integer

This attribute specifies that Directory Server should skip the double-check it normally does to verify that search results returned include the most current version of the entry content, even if the entry has been modified during the search. This double-check verification involves testing the search filter against each entry to return in response to the search.

Allowing Directory Server to skip the filter test when the search involves complex filters and large static groups can result in significant performance improvement.

When `nsslapd-search-tune` is set, the access log identifies searches for which the filter test is skipped with the tag `notes=F`.

Set `nsslapd-search-tune` to a sum of the following values:

1 Enable the filter test to be skipped before the entry is returned.

2 Reserved, do not use.

4 Reserved, do not use.

8 Skip the filter test even if the attribute in the filter is in the list of requested attributes. This could potentially cause the server to return entries that no longer correspond to search criteria.

16 Skip the filter test when search filters are complex, using `&` and `|`.

32 Always finish building the list of candidate entries from the index.

Examples

```
nsslapd-search-tune: 49
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name

nsslapd-securelistenhost – listen to IP address for secure connections

Description

Allows multiple Directory Server instances to run on a multi-homed machine, using secure SSL/TLS connections, and makes it possible to limit listening to one or more interfaces of a multi-homed machine. Provide the hostname or host names corresponding to the IP interface or interfaces you want to specify as the values for this attribute. Directory Server responds only to requests sent to the interface or interfaces corresponding to the host name or host names specified. This prevents other programs from using the same port as Directory Server on the interfaces specified.

Entry DN	`cn=config`
Valid Range	Any secure host name or host names
Default Value	Not applicable
Syntax	DirectoryString
Example	`nsslapd-securelistenhost:` *secure-hostname*

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

dse.ldif(4)

Name nsslapd-securePort – encrypted LDAP port number

Description TCP/IP port number used for SSL/TLS communications. This selected port must be unique on the host system; make sure no other application is attempting to use the same port number. Specifying a port number of less than 1024 requires that Directory Server runs as super user.

Note – Be aware when changing this port number of other applications whose configurations you may have to modify to reflect the change.

The default value 636 is only used if the server has been configured with a private key and a certificate; otherwise it does not listen on this port.

You must restart the server for the port number change to be taken into account.

Entry DN	`cn=config`
Valid Range	1 to 65535
Default Value	`636`
Syntax	Integer
Example	`nsslapd-securePort: 636`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-security – security

Description Enables the use of security features, SSL/TLS and attribute encryption, in Directory Server. If you require secure connections, or the use of the attribute encryption feature, this attribute should be set to `on`.

Entry DN	`cn=config`
Valid Range	on | off
Default Value	`off`
Syntax	DirectoryString
Example	`nsslapd-security: off`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-sizelimit – size limit

Description Specifies the maximum number of entries to return from a search operation. If this limit is reached, the server returns any entries it has located that match the search request, as well as an exceeded size limit error.

When no limit is set, the server will return every matching entry to the client regardless of the number found. To set a no limit value whereby Directory Server will wait indefinitely for the search to complete, specify a value of `-1` for this attribute in the `dse.ldif` file.

This limit applies to everyone regardless of their organization.

Entry DN	`cn=config`
	`cn=default instance config,cn=chaining database, cn=plugins,cn=config`

Valid Range	-1 to the maximum 32 bit integer value (2147483647)
Default Value	`2000`
Syntax	Integer
Example	`nsslapd-sizelimit: 2000`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-state – how a suffix handles operations

Description Determines how the suffix handles operations.

Entry DN `cn="`*suffixName*`",cn=mapping tree,cn=config`

Valid Range

`backend`	The backend database is used to process all operations.
`disabled`	The database is not available for processing operations. The server returns a “No such search object” error in response to requests made by client applications.
`referral`	A referral is returned for requests made to this suffix.
`referral on update`	The database is used for all operations except update requests, which receive a referral.

Default Value `backend`

Syntax DirectoryString

Example `nsslapd-state: backend`

Note – You can manually change the value of the `nsslapd-state` attribute. For example, you can change the value to `referral` or `referral on update` if you want the server to be read-only for the duration of a backup.

However, if replication is enabled, replication manages the value of the `nsslapd-state` attribute, and overwrites the value you have manually set.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-suffix – chained suffix identifier

Description Specifies the chained suffix. This is a single-valued attribute as each database instance can have only one suffix. Previously, it was possible to have more than one suffix on a single database instance but this is no longer the case. Any changes made to this attribute after the entry has been created take effect only after you restart the server containing the chained suffix.

Entry DN	`cn=`*dbName*`,cn=ldbm database,cn=plugins,cn=config`
Valid Range	Any valid DN
Default Value	Not applicable
Syntax	DirectoryString
Example	`nsslapd-suffix: o=example`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-threadnumber – thread number

Description Defines the number of operation threads that Directory Server creates during startup. The `nsslapd-threadnumber` value should be increased if you have many directory clients performing time-consuming operations such as add or modify. This ensures that there are other threads available for servicing short-lived operations such as simple searches.

Entry DN	`cn=config`
Valid Range	1 to the number of threads supported by your system
Default Value	`30`
Syntax	Integer
Example	`nsslapd-threadnumber: 60`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsslapd-timelimit – time limit

Description Specifies the maximum number of seconds allocated for a search request. If this limit is reached, Directory Server returns any entries it has located that match the search request, as well as an exceeded time limit error.

When no limit is set, the server returns every matching entry to the client regardless of the time it takes. To set a no limit value whereby Directory Server waits indefinitely for the search to complete, specify a value of `-1` for this attribute in the `dse.ldif` file. A value of zero causes no time to be allowed for searches. The smallest time limit is 1 second.

Entry DN	`cn=config`
	`cn=default instance config,cn=chaining database, cn=plugins,cn=config`

Valid range	-1 to the maximum 32 bit integer value (2147483647) in seconds
Default value	`3600`
Syntax	Integer
Example	`nsslapd-timelimit: 3600`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name

nsslapd-versionstring – version string

Description

Specifies the server version number.

Entry DN	`cn=config`
Valid range	Any valid server version number.
Default value	Not applicable
Syntax	DirectoryString
Example	`nsslapd-versionstring: Sun-Java(tm)-System-Directory/6.0`

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

dse.ldif(4)

Name nsSSL2 – SSL v2 support

Description Supports SSL version 2.

Entry DN	`cn=encryption,cn=config`
Valid Range	`on | off`
Default Value	`off`
Syntax	DirectoryString
Example	`nsSSL2: on`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name

nsSSL3 – SSL v3 support

Description

Supports SSL version 3.

Entry DN	`cn=encryption,cn=config`
Valid Range	`on	off`
Default Value	`off`
Syntax	DirectoryString
Example	`nsSSL3: on`

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

dse.ldif(4)

Name

nsSSL3ciphers – SSL encryption ciphers

Description

This multivalued attribute specifies the set of encryption ciphers Directory Server uses during SSL communications. The default value, `all`, does not mean all the supported SSL ciphers, as supported ciphers with NULL key length are removed from the list.

Entry DN	`cn=encryption,cn=config`
Valid Range	Ciphers shown as `supportedSSLCiphers` on the root DSE, or `all`
Default Value	`all`
Syntax	DirectoryString

Use the + symbol to enable or - symbol to disable ciphers, followed by the cipher identifier. Blank spaces are not allowed in the list of ciphers.

To enable all ciphers, except `rsa_null_md5` which must be specifically called, you can specify `all`.

Example `nsSSL3ciphers: +RSA_NULL_MD5,+RC4_56_SHA,-RC4_56_SHA`

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

dse.ldif(4)

Name nsSSLClientAuth – use SSL client authentication

Description In an SSL connection, this attribute specifies whether a client certificate is `allowed`, `required`, or should not be sent, `off`, to the SSL server.

Entry DN	`cn=encryption,cn=config`
Valid Range	`off	allowed	required`
Default Value	`allowed`
Syntax	DirectoryString
Example	`nsSSLClientAuth: allowed`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsSSLServerAuth – use SSL server authentication

Description Specifies the action that the SSL client should take on the server certificate sent by the SSL server in an SSL connection.

Entry DN `cn=encryption,cn=config`

Valid Range

`cert` Verify whether the server certificate is from a trusted certificate authority.

`cncheck` Verify whether the server certificate is from a trusted certificate authority *and* verify the DN contained in the server certificate, to avoid man-in-the middle attacks on the server.

`weak` Make no attempt to verify whether the server certificate is from a trusted certificate authority.

Default Value `cert`

Syntax DirectoryString

Example `nsSSLServerAuth: cert`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsSSLSessionTimeout – SSL session time out

Description Specifies the lifetime duration of an SSL session for both SSLv2 and SSLv3. The minimum timeout value is 5 seconds and if you enter a value below this, it is automatically replaced by 5 seconds. Values outside the valid ranges are replaced by the default value of 100 seconds for SSLv2.

Entry DN	`cn=encryption,cn=config`
Valid Range	(SSLv2) 5 seconds to 100 seconds (SSLv3) 5 seconds to 24 hours
Default Value	`0`, which translates to 100 seconds if you are running SSLv2 and 24 hours if you are running SSLv3
Syntax	Integer
Example	`nsSSLSessionTimeout: 5`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name

nsState – Clock state for replication, unique ID generation

Description

PROPERTY	VALUE

Entry DN	`cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
	`cn=uniqueid generator,cn=config`

Valid Range	N/A
Default Value	N/A
Syntax	Binary

This attribute is part of replica configuration for `nsDS5Replica` entries.

This attribute is reserved for internal use for handling clock skew and detecting backward clock errors. Do not edit the value of this attribute.

Examples

```
nsState:: AAEAAAAAAAAAAAAAQ4rs0AAAAAAAAAAAAAAAAAAAAAAAAQAAAAAAAA==
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name nsSystemIndex – identify index as system index

Description This mandatory attribute specifies whether the index is a system index, that is, an index that is vital for Directory Server operations. If this attribute has a value of `true`, it is system essential. System indexes must not be removed as this seriously disrupts server functionality.

Entry DN `cn=default indexes,cn=config,cn=ldbm database,`
`cn=plugins,cn=config`

Valid Range `true | false`

Default Value Depends on the index

Syntax DirectoryString

Example `nssystemindex: true`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name nsTransmittedControls – chained LDAP controls

Description This attribute, which can be both a global, and thus dynamic, configuration, or a `cn=chained suffix instance,cn=chaining database, cn=plugins,cn=config` instance configuration attribute, allows you to alter the controls that the chained suffix forwards. The following controls are forwarded by default.

- Managed DSA, object identifier: 2.16.840.1.113730.3.4.2.
- Virtual list view (VLV), object identifier: 2.16.840.1.113730.3.4.9
- Server side sorting, object identifier: 1.2.840.113556.1.4.473

Entry DN `cn=config,cn=chaining database,cn=plugins,cn=config`

Valid Range Any valid OID or the above listed controls forwarded by the chained suffix.

Default Value None

Syntax Integer

Example `nsTransmittedControls: 1.2.840.113556.1.4.473`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also dse.ldif(4)

Name plugin, argument, depends-on-named, depends-on-type, feature, init-func, lib-path, type, vendor, version – DS plug-in configuration (PLG) properties

Description Directory Server implements some key functionality as *plug-ins*. Plug-ins take the form of libraries loaded when the server starts, and called at different points in the processing of client application requests. When you create custom plug-ins, you must configure the server to load and use them, then restart Directory Server to load the newly configured plug-ins.

PROPERTY: `argument`

Syntax	`STRING`
Default Value	`None`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	Yes

This property specifies arguments passed to the plug-in when it is loaded by Directory Server. Arguments are passed in the order you specify them. Updating the list of arguments replaces all the existing arguments previously specified when the plug-in is loaded again.

PROPERTY: `depends-on-named`

Syntax	`STRING`
Default Value	`None`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	Yes

This property specifies names of plug-ins that must be available and loaded before Directory Server loads the current plug-in.

PROPERTY: `depends-on-type`

Syntax	`STRING` (See the description that follows.)
Default Value	`None`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	Yes

This property specifies types of plug-ins that must be available and loaded before Directory Server loads the current plug-in. The value must be the value of a plug-in `type` property.

PROPERTY: `feature`

Syntax	`STRING`
Default Value	`None`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the plug-in identifier from the `Slapi_PluginDesc` structure.

PROPERTY: `init-func`

Syntax	`STRING`
Default Value	`None`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the name of function called by Directory Server to initialize the plug-in.

PROPERTY: `lib-path`

Syntax	`PATH`
Default Value	`None`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the absolute file system path to the library containing plug-in.

PROPERTY: `type`

Syntax	`STRING` (See the description that follows.)
Default Value	`None`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the plug-in type. The following plug-in types are supported.

`ldbmentryfetchstore`	Entry store and fetch plug-in
`extendedop`	Extended operation plug-in
`passwordcheck`	Password check plug-in
`postoperation`	Post-operation plug-in
`preoperation`	Pre-operation plug-in
`internalpostoperation`	Internal post-operation plug-in
`internalpreoperation`	Internal pre-operation plug-in
`matchingrule`	Matching rule plug-in for extensible match search filters
`object`	Generic plug-in type, sometimes used to register other plug-ins
`passwordcheck`	Strong password check plug-in
`pwdstoragescheme`	Password storage scheme plug-in

PROPERTY: `vendor`

Syntax	`STRING`
Default Value	Vendor name in plug-in configuration
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the plug-in vendor from the `Slapi_PluginDesc` structure.

PROPERTY: `version`

Syntax	`STRING`
Default Value	Version string in plug-in configuration
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the plug-in version from the `Slapi_PluginDesc` structure.

Description Syntax values shown in lower case or partly in lower case are literal values.

Those shown in upper case are syntax types, defined as follows:

`ATTR_NAME`
A valid attribute type name such as `cn` or `objectClass`.

`BOOLEAN`
`true` or `false`.

`DN`
A valid distinguished name such as `ou=People,dc=example,dc=com`.

`DURATION`
A duration specified in months (`M`), weeks (`w`), days (`d`), hours (`h`), minutes (`m`), seconds (`s`), and miliseconds (`ms`), or some combination with multiple specifiers. For example, you can specify one week as `1w`, `7d`, `168h`, `10080m`, or `604800s`. You can also specify one week as `1w0d0h0m0s`.

`DURATION` properties typically do not each support all duration specifiers (`Mwdhms`). Examine the output of `dsconf help-properties` for the property to determine which duration specifiers are supported.

`EMAIL_ADDRESS`
A valid e-mail address.

`HOST_NAME`
An IP address or host name.

`INTEGER`
A positive integer value between 0 and the maximum supported integer value in the system address space. On 32-bit systems, `2147483647`. On 64-bit systems, `9223372036854775807`.

`INTERVAL`
An interval value of the form *hhmm*-*hhmm 0123456*, where the first element specifies the starting hour, the next element the finishing hour in 24-hour time format, from `0000-2359`, and the second specifies days, starting with Sunday (`0`) to Saturday (`6`).

`IP_RANGE`
An IP address or range of address in one of the following formats:

- IP address in dotted decimal form.
- IP address and bits, in the form of network number/mask bits.
- IP address and quad, in the form of a pair of dotted decimal quads.
- All address. A catch-all for clients that are note placed into other, higher priority groups.
- `0.0.0.0`. This address is for groups to which initial membership is not considered. For example, for groups that clients switch to after their initial bind.
- IP address of the local host.

`LDAP_URL`
A valid LDAP URL as specified by RFC 2255 (`http://www.ietf.org/rfc/rfc2255.txt`).

MEMORY_SIZE
: A memory size specified in gigabytes (G), megabytes (M),kilobytes (k), or bytes (b). Unlike DURATION properties, MEMORY_SIZE properties cannot combine multiple specifiers. However, MEMORY_SIZE properties allow decimal values, for example, 1.5M.

NAME
: A valid cn (common name).

OCTAL_MODE
: A three-digit, octal file permissions specifier. The first digit specifies permissions for the server user ID, the second for the server group ID, the last for other users. Each digit consists of a bitmask defining read (4), write (2), execute (1), or no access (0) permissions, thus 640 specifies read-write access for the server user, read-only access for other users of the server group, and no access for other users.

PASSWORD_FILE
: The full path to the file from which the bind password should be read.

PATH
: A valid, absolute file system path.

STRING
: A DirectoryString value, as specified by RFC 2252 (http://www.ietf.org/rfc/rfc2252.txt).

SUPPORTED_SSL_CIPHER
: An SSL cipher supported by the server. See the Reference for a list of supported ciphers.

SUPPORTED_SSL_PROTOCOL
: An SSL protocol supported by the server. See the Reference for a list of supported protocols.

TIME
: A time of the form *hhmm* in 24-hour format, where *hh* stands for hours and *mm* stands for minutes.

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Stable

See Also

dsconf(1M), desc(5dsconf), enabled(5dsconf)

Name referral-url – Whether the server accepts mod DN operations

Description

Syntax	`LDAP_URL | undefined`
Default Value	`undefined`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	Yes

This property specifies the URLs returned as referrals when clients request an operation not supported by the server instance or suffix for the target entry.

Description Syntax values shown in lower case or partly in lower case are literal values.

Those shown in upper case are syntax types, defined as follows:

`ATTR_NAME`
A valid attribute type name such as `cn` or `objectClass`.

`BOOLEAN`
`true` or `false`.

`DN`
A valid distinguished name such as `ou=People,dc=example,dc=com`.

`DURATION`
A duration specified in months (`M`), weeks (`w`), days (`d`), hours (`h`), minutes (`m`), seconds (`s`), and miliseconds (`ms`), or some combination with multiple specifiers. For example, you can specify one week as `1w`, `7d`, `168h`, `10080m`, or `604800s`. You can also specify one week as `1w0d0h0m0s`.

`DURATION` properties typically do not each support all duration specifiers (`Mwdhms`). Examine the output of `dsconf help-properties` for the property to determine which duration specifiers are supported.

`EMAIL_ADDRESS`
A valid e-mail address.

`HOST_NAME`
An IP address or host name.

`INTEGER`
A positive integer value between 0 and the maximum supported integer value in the system address space. On 32-bit systems, `2147483647`. On 64-bit systems, `9223372036854775807`.

`INTERVAL`
An interval value of the form *hhmm*-*hhmm 0123456*, where the first element specifies the starting hour, the next element the finishing hour in 24-hour time format, from `0000-2359`, and the second specifies days, starting with Sunday (`0`) to Saturday (`6`).

`IP_RANGE`
An IP address or range of address in one of the following formats:

- IP address in dotted decimal form.
- IP address and bits, in the form of network number/mask bits.
- IP address and quad, in the form of a pair of dotted decimal quads.
- All address. A catch-all for clients that are note placed into other, higher priority groups.
- `0.0.0.0`. This address is for groups to which initial membership is not considered. For example, for groups that clients switch to after their initial bind.
- IP address of the local host.

`LDAP_URL`
A valid LDAP URL as specified by RFC 2255 (`http://www.ietf.org/rfc/rfc2255.txt`).

`MEMORY_SIZE`
A memory size specified in gigabytes (`G`), megabytes (`M`),kilobytes (`k`), or bytes (`b`). Unlike `DURATION` properties, `MEMORY_SIZE` properties cannot combine multiple specifiers. However, `MEMORY_SIZE` properties allow decimal values, for example, 1.5M.

`NAME`
A valid `cn` (common name).

`OCTAL_MODE`
A three-digit, octal file permissions specifier. The first digit specifies permissions for the server user ID, the second for the server group ID, the last for other users. Each digit consists of a bitmask defining read (`4`), write (`2`), execute (`1`), or no access (`0`) permissions, thus `640` specifies read-write access for the server user, read-only access for other users of the server group, and no access for other users.

`PASSWORD_FILE`
The full path to the file from which the bind password should be read.

`PATH`
A valid, absolute file system path.

`STRING`
A DirectoryString value, as specified by RFC 2252 (`http://www.ietf.org/rfc/rfc2252.txt`).

`SUPPORTED_SSL_CIPHER`
An SSL cipher supported by the server. See the Reference for a list of supported ciphers.

`SUPPORTED_SSL_PROTOCOL`
An SSL protocol supported by the server. See the Reference for a list of supported protocols.

`TIME`
A time of the form *hhmm* in 24-hour format, where *hh* stands for hours and *mm* stands for minutes.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Evolving

See Also

dsconf(1M), server(5dsconf), suffix(5dsconf)

Name repl-agmt, auth-bind-dn, auth-protocol, auth-pwd, repl-fractional-exclude-attr, repl-fractional-include-attr, repl-schedule, transport-compression, transport-group-size, transport-window-size – DS replication agreement configuration (RAG) properties

Description A *replication agreement* governs how a Directory Server supplier updates a Directory Server consumer. Although this configuration element is called an agreement, it concerns the configuration only of the supplier.

PROPERTY: `auth-bind-dn`

Syntax	`DN	undefined`
Default Value	`undefined`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the bind DN used by the supplier to bind to the consumer in order to perform replication-related updates. This bind DN must be present on the consumer.

PROPERTY: `auth-protocol`

Syntax	`clear|ssl-simple|ssl-client`
Default Value	`clear`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the protocol used by the supplier to bind to the consumer in order to perform replication-related updates. The default is to bind with simple authentication in clear text without securing the connection, as most replications connections are made on an internal network. You may however configure replication to use SSL and simple authentication to protect the connection from malicious snooping, or SSL with client authentication to further protect the connection.

PROPERTY: `auth-pwd`

Syntax	`STRING`
Default Value	`None`
Is readable	Yes
Is modifiable	No

Is multi-valued	No

This property specifies the password used by the supplier to bind to the consumer. You provide it using `auth-pwd-file`.

PROPERTY: `auth-pwd-file`

Syntax	`PATH`
Default Value	`""`
Is readable	No
Is modifiable	Yes
Is multi-valued	No

This property specifies the file from which the bind password for replication is read to create the replication agreement. The file is read once on replication agreement creation, and the password is stored for future use.

PROPERTY: `repl-fractional-exclude-attr`

Syntax	`ATTR_NAME | ""`
Default Value	`""`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	Yes

This property specifies the list of attributes not to replicate. This property is mutually exclusive with `repl-fractional-include-attr`.

PROPERTY: `repl-fractional-include-attr`

Syntax	`ATTR_NAME | ""`
Default Value	`""`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	Yes

This property specifies the list of attributes to replicate. This property is mutually exclusive with `repl-fractional-exclude-attr`.

PROPERTY: `repl-schedule`

Syntax	`INTERVAL	always`
Default Value	`always`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the times and days when replication can take place.

PROPERTY: `transport-compression`

Syntax	`best-compression	best-speed	default-compression	no-compression`
Default Value	`no-compression`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the level of `libz`(3) compression used on replication updates from the supplier to the consumer. Supported settings are as follows.

`no-compression`	No compression
`default-compression`	Default zlib compression (zlib numeric value = -1)
`best-speed`	Fastest zlib compression (zlib numeric value = 1)
`best-compression`	Strongest zlib compression (zlib numeric value = 9)

If the bottleneck for replication in your environment is network bandwidth, this property can potentially help you tune the replication protocol for better performance.

PROPERTY: `transport-group-size`

Syntax	`INTEGER`
Default Value	`1`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies how many replication messages are grouped on the supplier before being sent to the consumer. Valid range is 1 to 255.

If the bottleneck for replication in your environment is network bandwidth, this property can potentially help you tune the replication protocol for better performance.

PROPERTY: `transport-window-size`

Syntax	`INTEGER`
Default Value	`10`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the number of replication messages sent from the supplier to the consumer before waiting for a response from the consumer to continue. Valid range is 1 to 65535.

If the bottleneck for replication in your environment is network latency or network bandwidth, this property can potentially help you tune the replication protocol for better performance.

Description

Syntax values shown in lower case or partly in lower case are literal values.

Those shown in upper case are syntax types, defined as follows:

`ATTR_NAME`
A valid attribute type name such as `cn` or `objectClass`.

`BOOLEAN`
`true` or `false`.

`DN`
A valid distinguished name such as `ou=People,dc=example,dc=com`.

`DURATION`
A duration specified in months (`M`), weeks (`w`), days (`d`), hours (`h`), minutes (`m`), seconds (`s`), and miliseconds (`ms`), or some combination with multiple specifiers. For example, you can specify one week as `1w`, `7d`, `168h`, `10080m`, or `604800s`. You can also specify one week as `1w0d0h0m0s`.

`DURATION` properties typically do not each support all duration specifiers (`Mwdhms`). Examine the output of `dsconf help-properties` for the property to determine which duration specifiers are supported.

`EMAIL_ADDRESS`
A valid e-mail address.

`HOST_NAME`
An IP address or host name.

`INTEGER`
A positive integer value between 0 and the maximum supported integer value in the system address space. On 32-bit systems, `2147483647`. On 64-bit systems, `9223372036854775807`.

`INTERVAL`
An interval value of the form *hhmm*-*hhmm 0123456*, where the first element specifies the starting hour, the next element the finishing hour in 24-hour time format, from `0000-2359`, and the second specifies days, starting with Sunday (`0`) to Saturday (`6`).

`IP_RANGE`
An IP address or range of address in one of the following formats:

- IP address in dotted decimal form.
- IP address and bits, in the form of network number/mask bits.
- IP address and quad, in the form of a pair of dotted decimal quads.
- All address. A catch-all for clients that are note placed into other, higher priority groups.
- `0.0.0.0`. This address is for groups to which initial membership is not considered. For example, for groups that clients switch to after their initial bind.
- IP address of the local host.

`LDAP_URL`
A valid LDAP URL as specified by RFC 2255 (`http://www.ietf.org/rfc/rfc2255.txt`).

`MEMORY_SIZE`
A memory size specified in gigabytes (`G`), megabytes (`M`),kilobytes (`k`), or bytes (`b`). Unlike `DURATION` properties, `MEMORY_SIZE` properties cannot combine multiple specifiers. However, `MEMORY_SIZE` properties allow decimal values, for example, 1.5M.

`NAME`
A valid `cn` (common name).

`OCTAL_MODE`
A three-digit, octal file permissions specifier. The first digit specifies permissions for the server user ID, the second for the server group ID, the last for other users. Each digit consists of a bitmask defining read (`4`), write (`2`), execute (`1`), or no access (`0`) permissions, thus `640` specifies read-write access for the server user, read-only access for other users of the server group, and no access for other users.

`PASSWORD_FILE`
The full path to the file from which the bind password should be read.

`PATH`
A valid, absolute file system path.

`STRING`
A DirectoryString value, as specified by RFC 2252 (`http://www.ietf.org/rfc/rfc2252.txt`).

`SUPPORTED_SSL_CIPHER`
An SSL cipher supported by the server. See the Reference for a list of supported ciphers.

`SUPPORTED_SSL_PROTOCOL`
An SSL protocol supported by the server. See the Reference for a list of supported protocols.

`TIME`
A time of the form *hhmm* in 24-hour format, where *hh* stands for hours and *mm* stands for minutes.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Evolving

See Also

dsconf(1M), desc(5dsconf), enabled(5dsconf)

Name

replication, nsDS5Replica, nsDS5ReplicationAgreement, ReplPriorityRule – Directory Server replication configuration

Description

Note – In most cases you do not need to know how to manage the object classes and attributes mentioned here. Instead you handle replication configuration through Directory Service Control Center or the dsconf(1M) command. The information here is included primarily for those of you who are familiar with command-line configuration for replication in previous Directory Server versions.

Replication works in Directory Server using extended operations. Changes to a suffix on a supplier server are replayed on the consumer server. Each server stores configuration information defining its role in replication, and defining the user account that has access to perform replication operations. The supplier server also stores configuration information about the replication agreement it has with the consumer.

Replication configuration is reflected in object classes and attributes under `cn=config` of the Directory Server instance.

Replica Configuration

The configuration entry that indicates a suffix is replicated has a DN of the following form.

```
cn=replica,cn=suffix name,cn=mapping tree,cn=config
```

For example, the following configuration entry DN corresponds to the suffix `dc=example,dc=com`.

```
cn=replica,cn=dc\=example\,dc\=com,cn=mapping tree,cn=config
```

Such entries have the object class `nsDS5Replica`.

Replication Agreement Configuration

The configuration entries that describe replication agreements with other servers have DNs of the following form.

```
cn=agreement name,cn=replica,cn=suffix name,cn=mapping tree,
 cn=config
```

For example, the following configuration entry DN corresponds to the suffix `dc=example,dc=com`.

```
cn=ds.example.com:389,cn=replica,cn=dc\=example\,dc\=com,
 cn=mapping tree,cn=config
```

Replication agreement entries have the object class `nsDS5ReplicationAgreement`.

Replication Priority Configuration

The configuration entries that describe replication priority rules have DNs of the form:

```
cn=rule name,cn=replica,cn=suffix name,cn=mapping tree,
 cn=config
```

For example, the following configuration entry DN corresponds to the suffix `dc=example,dc=com`.

```
cn=pwdReplPrio,cn=replica,cn=dc\=example\,dc\=com,cn=mapping tree,
 cn=config
```

Replication priority rule entries have the object class `ReplPriorityRule`.

Replication Manager Configuration

The configuration entry for the account used to bind and perform replication has, by default, the DN:

```
cn=replication manager,cn=replication,cn=config
```

The account entry is a standard person(5dsoc) object class. It defines the DN and `userPassword` for replication. The password policy for the account used to bind and perform replication has, by default, the DN:

```
cn=Password Policy,cn=replication manager,cn=replication,
 cn=config
```

The password policy entry is governed by the schema for pwpolicy(5dssd). As both `person` and `pwpolicy` related object classes and attribute types are described elsewhere, they are not further described here.

Extended Description

This section examines replication configuration on a master supplier server, and on a dedicated consumer server. The sample configurations shown here were created using the `dsconf` to configure replication.

Supplier Side Configuration

On the supplier side, the key configuration entries are the replica configuration and the replication agreement.

Supplier Replica Configuration

For a master supplier, an `nsDS5Replica` configuration entry looks something like the following:

```
dn: cn=replica,cn=\dc=example\,dc\=com,cn=mapping tree,cn=config
nsDS5ReplicaRoot: dc=example,dc=com
nsDS5ReplicaBindDN: cn=replication manager,cn=replication,cn=config
objectClass: top
objectClass: nsDS5Replica
nsDS5ReplicaId: 1
nsDS5Flags: 1
nsDS5ReplicaType: 3
cn: replica
creatorsName: cn=directory manager
modifiersName: cn=directory manager
nsState:: AAEAAAAAAAAAAAAAQ4rs0AAAAAAAAAAAAAAAAAAAAAAAAQAAAAAAAA==
nsDS5ReplicaName: d2e14d02-600311da-80ace5db-c83e55ac
numSubordinates: 1
```

In this example, the key attributes are the following:

`nsDS5ReplicaRoot` Holds the DN of the root of the replicated suffix. Once set, it must not be modified.

`nsDS5ReplicaBindDN`
Holds the DN of the account used to bind for replication. If replication is performed over SSL, this attribute can hold the certificate identity associated with the DN.

`nsDS5ReplicaId`
Specifies the unique ID of this master supplier server, a value from `1-65534`, inclusive.

A hub uses the same replica ID as a consumer, `65535`.

`nsDS5Flags`
Governs change logging and automatic referrals. It takes one of the following values:

`0` No changes are logged. Automatic referrals are not overwritten.

`1` Changes are logged. Automatic referrals are not overwritten.

`4` No changes are logged. Automatic referrals are overwritten.

`5` Changes are logged. Automatic referrals are overwritten.

`nsDS5ReplicaType`
Defines the role this replica plays in replicating with other servers. It takes one of the following values:

`0` Reserved for internal use

`1` Dedicated supplier

`2` Dedicated consumer (read-only)

`3` Supplier/consumer (read-write)

`cn`
This attribute names the replica. Once set, it must not be modified.

`nsState`
Stores the state of the clock for handling synchronization. Reserved for internal use.

`nsDS5ReplicaName`
Read-only unique identifier for the replica.

The replica configuration entry can also hold the following attributes not shown here:

ds5BeginReplicaAcceptUpdates(5dsconf)
ds5ReplicaConsumerTimeout(5dsconf)
ds5LastInitTimeStamp(5dsconf)
ds5ReferralDelayAfterInit(5dsconf)
dsChangelogMaxAge(5dsconf)
dsChangelogMaxentries(5dsconf)
dsFilterSPConfigchecksum(5dsconf)
nsDS5ReplicaAutoReferral(5dsconf)
nsDS5ReplicaChangeCount(5dsconf)

nsDS5ReplicaPurgeDelay(5dsconf)
nsDS5ReplicaReferral(5dsconf)
nsDS5ReplicaTombstonePurgeInterval(5dsconf)
nsDS5Task(5dsconf)

Supplier Replication Agreement

For a master supplier, an `nsDS5ReplicationAgreement` configuration entry looks something like the following:

```
dn: cn=ds.example.com:389,cn=replica,cn=dc\=example\,dc\=com,
 cn=mapping tree,cn=config
nsDS5ReplicaHost: ds.example.com
nsDS5ReplicaUpdateSchedule: *
nsDS5ReplicaTransportInfo: LDAP
objectClass: top
objectClass: nsDS5ReplicationAgreement
nsDS5ReplicaPort: 389
nsDS5ReplicaBindMethod: SIMPLE
cn: ds.example.com:389
nsDS5ReplicaRoot: dc=example,dc=com
nsDS5ReplicaBindDN: cn=replication manager,cn=replication,
 cn=config
nsDS5ReplicaCredentials:: e0RFU31JakduS3VZSWhEcThEcExDQlU2
 VlN2QTdjcUw4emhDdXl3Sldmc3NTZ2t3eS9mWmR4VmpUZlVYRE1NLzR2T
 UVBDQpyZVdYU3A3U1ZwYz0=
```

In this example, the key attributes are the following:

`nsDS5ReplicaHost`
Holds the host name where the consumer runs.

`nsDS5ReplicaUpdateSchedule`
Specifies when replication happens. If you must restrict the time when replication can happen, set this attribute. This attribute can take multiple values of the form *hhmm-hhmm 0123456*, where the first element specifies the time span, and the second specifies which days, starting with Sunday (`0`) to Saturday (`6`).

`nsDS5ReplicaTransportInfo`
Specifies the transport used for replication, `LDAP` or `SSL`.

`nsDS5ReplicaPort`
Holds port number on which the consumer listens.

`cn`
This attribute names the replication agreement. Once set, it must not be modified.

`nsDS5ReplicaBindMethod`
Specifies bind protocol, `SIMPLE`, `SSLCLIENTAUTH`.

`nsDS5ReplicaRoot`
Holds the DN of the root of the replicated suffix. Once set, it must not be modified.

`nsDS5ReplicaBindDN`	Holds the DN of the account on the consumer used for replication.
`nsDS5ReplicaCredentials`	Holds the bind credentials of the account on the consumer used for replication.

The replication agreement entry can also hold the following attributes not shown here:

description(5dsat)
ds5AgreementEnable(5dsconf)
ds5ReplicaAutomaticInit(5dsconf)
ds5ReplicaForce51Protocol(5dsconf)
ds5ReplicaTransportCompressionLevel(5dsconf)
ds5ReplicaTransportConcurrencyLevel(5dsconf)
ds5ReplicaTransportGroupSize(5dsconf)
ds5ReplicaTransportGrpPktSize(5dsconf)
ds5ReplicaTransportWindowSize(5dsconf)
ds6ruv(5dsconf)
dsReplFractionalExclude(5dsconf)
dsReplFractionalInclude(5dsconf)
nsDS5ReplicaUpdateSchedule(5dsconf)
nsds50ruv(5dsconf)
nsds5BeginReplicaRefresh(5dsconf)
nsds5ReplicaTimeout(5dsconf)
nsds5replicaChangesSentSinceStartup(5dsconf)
nsds5replicaLastInitEnd(5dsconf)
nsds5replicaLastInitStart(5dsconf)
nsds5replicaLastInitStatus(5dsconf)
nsds5replicaLastUpdateEnd(5dsconf)
nsds5replicaLastUpdateStart(5dsconf)
nsds5replicaLastUpdateStatus(5dsconf)
nsds5replicaUpdateInProgress(5dsconf)

Supplier Priority Rule

For a master supplier, a `ReplPriorityRule` configuration entry looks something like the following:

```
dn: cn=pwdReplPrio,cn=replica,cn=dc\=example\,dc\=com,cn=mapping tree,cn=config
objectClass: top
objectClass: ReplPriorityRule
replPriorityType: mod
replPriorityAttribute: userPassword
nsDS5ReplicaRoot: dc=example,dc=com
cn: pwdReplPrio
```

In this example, the key attributes are the following:

`replPriorityType`
Specifies the type of operation which are replicated with high priority.

`replPriorityAttribute`
Specifies the attribute whose changes are replicated with high priority.

`nsDS5ReplicaRoot`
Holds the DN of the root of the replicated suffix to which this priority rule applies.

`cn`
This attribute names the priority rule.

The replication priority rule entry can also hold the following attributes not shown here:

replPriorityBaseDN(5dsconf)
replPriorityBindDN(5dsconf)

Consumer Side Configuration

On the consumer side, the key configuration entry is the replica configuration. On a dedicated consumer, it is also useful to see how the mapping tree entry for the suffix is configured to refer updates to the supplier.

Consumer Replica Configuration

For a dedicated consumer, an `nsDS5Replica` configuration entry looks something like the following:

```
dn: cn=replica,cn=dc\=example\,dc\=com,cn=mapping tree,cn=config
nsDS5ReplicaRoot: dc=example,dc=com
nsDS5ReplicaBindDN: cn=replication manager,cn=replication,cn=config
objectClass: top
objectClass: nsDS5Replica
nsDS5Flags: 0
nsDS5ReplicaType: 2
cn: replica
nsDS5ReplicaId: 65535
nsState:: //8AAAAAAAAAAAAAQ4rtPAAAAAAAAAAAAAAAAAAAAAAAAQAAAAAAAAA==
nsDS5ReplicaName: 07ed9f82-600411da-8096d729-0ba76005
dsFilterSPConfigchecksum: 0
ds5LastInitTimeStamp: 0
```

Key attributes in this example are explained in *Supplier Replica Configuration*.

A hub uses the same replica ID as a consumer, `65535`.

Referrals On Consumer

For a dedicated consumer suffix, the mapping tree configuration entry refers client applications to the supplier for write operations:

```
dn: cn=dc\=example\,dc\=com,cn=mapping tree,cn=config
objectClass: top
objectClass: extensibleObject
objectClass: nsMappingTree
nsslapd-backend: example
cn: dc=example,dc=com
numSubordinates: 1
nsslapd-referral: ldap://master.example.com:389/dc%3Dexample,dc%3Dcom
nsslapd-state: referral on update
```

Notice that the `nsslapd-referral` attribute refers clients to the master on host `master.example.com` and port `389` when they request update operations that would write to the directory. These attributes are set by the server when replication is initialized. The dedicated consumer then accepts write operations only from the supplier replica.

Attributes

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

dsconf(1M), person(5dsoc), pwpolicy(5dssd)

Name repl-priority, attr, base-dn, bind-dn, op-type – DS prioritized replication configuration (RPR) properties

Description Prioritized replication lets you force a Directory Server supplier to assign higher priority to certain updates replicated on a Directory Server consumer. You prioritize replication operations by creating replication priority rules.

PROPERTY: `attr`

Syntax	`ATTR_NAME`
Default Value	`None`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the attribute type to which the replication priority rule applies.

PROPERTY: `base-dn`

Syntax	`DN`
Default Value	`None`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the DN under which the replication priority rule applies. For example, if you set `base-dn:ou=administrators,dc=example,dc=com`, then changes to `uid=myAdmin,ou=administrators,dc=example,dc=com` might be replicated with high priority, but changes to `uid=bjensen,ou=people,dc=example,dc=com` would not.

PROPERTY: `bind-dn`

Syntax	`DN`
Default Value	`None`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies a bind DN for an account whose updates might be replicated with high priority.

PROPERTY: `op-type`

Syntax	`add	mod	del`
Default Value	`None`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies a type of operation for which updates might be replicated with high priority.

Description

Syntax values shown in lower case or partly in lower case are literal values.

Those shown in upper case are syntax types, defined as follows:

`ATTR_NAME`
A valid attribute type name such as `cn` or `objectClass`.

`BOOLEAN`
`true` or `false`.

`DN`
A valid distinguished name such as `ou=People,dc=example,dc=com`.

`DURATION`
A duration specified in months (`M`), weeks (`w`), days (`d`), hours (`h`), minutes (`m`), seconds (`s`), and miliseconds (`ms`), or some combination with multiple specifiers. For example, you can specify one week as `1w`, `7d`, `168h`, `10080m`, or `604800s`. You can also specify one week as `1w0d0h0m0s`.

`DURATION` properties typically do not each support all duration specifiers (`Mwdhms`). Examine the output of `dsconf help-properties` for the property to determine which duration specifiers are supported.

`EMAIL_ADDRESS`
A valid e-mail address.

`HOST_NAME`
An IP address or host name.

`INTEGER`
A positive integer value between 0 and the maximum supported integer value in the system address space. On 32-bit systems, `2147483647`. On 64-bit systems, `9223372036854775807`.

`INTERVAL`
An interval value of the form *hhmm-hhmm 0123456*, where the first element specifies the starting hour, the next element the finishing hour in 24-hour time format, from `0000-2359`, and the second specifies days, starting with Sunday (`0`) to Saturday (`6`).

`IP_RANGE`
An IP address or range of address in one of the following formats:

- IP address in dotted decimal form.
- IP address and bits, in the form of network number/mask bits.
- IP address and quad, in the form of a pair of dotted decimal quads.
- All address. A catch-all for clients that are note placed into other, higher priority groups.
- `0.0.0.0`. This address is for groups to which initial membership is not considered. For example, for groups that clients switch to after their initial bind.
- IP address of the local host.

`LDAP_URL`
A valid LDAP URL as specified by RFC 2255 (`http://www.ietf.org/rfc/rfc2255.txt`).

`MEMORY_SIZE`
A memory size specified in gigabytes (`G`), megabytes (`M`),kilobytes (`k`), or bytes (`b`). Unlike `DURATION` properties, `MEMORY_SIZE` properties cannot combine multiple specifiers. However, `MEMORY_SIZE` properties allow decimal values, for example, 1.5M.

`NAME`
A valid `cn` (common name).

`OCTAL_MODE`
A three-digit, octal file permissions specifier. The first digit specifies permissions for the server user ID, the second for the server group ID, the last for other users. Each digit consists of a bitmask defining read (`4`), write (`2`), execute (`1`), or no access (`0`) permissions, thus `640` specifies read-write access for the server user, read-only access for other users of the server group, and no access for other users.

`PASSWORD_FILE`
The full path to the file from which the bind password should be read.

`PATH`
A valid, absolute file system path.

`STRING`
A DirectoryString value, as specified by RFC 2252 (`http://www.ietf.org/rfc/rfc2252.txt`).

`SUPPORTED_SSL_CIPHER`
An SSL cipher supported by the server. See the Reference for a list of supported ciphers.

`SUPPORTED_SSL_PROTOCOL`
An SSL protocol supported by the server. See the Reference for a list of supported protocols.

`TIME`
A time of the form *hhmm* in 24-hour format, where *hh* stands for hours and *mm* stands for minutes.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Evolving

See Also

dsconf(1M)

Name

replPriorityAttribute – Attribute to replicate with high priority

Description

PROPERTY	VALUE
Entry DN	`cn=`*rule name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	Any valid attribute type
Default Value	N/A
Syntax	DirectoryString

This attribute is part of replica configuration for `ReplPriorityRule` entries.

When this multi-valued attribute is set, the server replicates changes to the attribute you specify with high priority.

Examples

```
replPriorityAttribute: userPassword
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name

replPriorityBaseDN – Replicate changes under this base with high priority

Description

PROPERTY	VALUE
Entry DN	`cn=`*rule name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	Any valid DN inside the replicated suffix
Default Value	N/A
Syntax	DirectoryString

This attribute is part of replica configuration for `ReplPriorityRule` entries.

When this attribute is set, the server replicates changes to entries under the specified DN with high priority.

Examples

```
replPriorityBaseDN: ou=people,dc=example,dc=com
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name replPriorityBindDN – Replicate changes performed by this user with high priority

Description

PROPERTY	VALUE
Entry DN	`cn=`*rule name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	Any valid DN
Default Value	N/A
Syntax	DirectoryString

This attribute is part of replica configuration for `ReplPriorityRule` entries.

When this attribute is set, the server replicates changes made by the user having the specified bind DN with high priority.

Examples

```
replPriorityBindDN: uid=admin,ou=Administrators,dc=example,dc=com
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also replication(5dsconf)

Name

replPriorityType – Operation type to replicate with high priority

Description

PROPERTY	VALUE
Entry DN	`cn=`*rule name*`,cn=replica,cn=`*suffix name*`,cn=mapping tree,cn=config`
Valid Range	`add` | `del` | `mod`
Default Value	All update operations
Syntax	DirectoryString

This attribute is part of replica configuration for `ReplPriorityRule` entries.

When this attribute is set, the server replicates changes of the type you specify with high priority as follows:

`add` Additions and deletions are replicated with high priority.

`del` Deletions are replicated with high priority.

`mod` Modifications are replicated with high priority.

Examples

```
replPriorityType: mod
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

replication(5dsconf)

Name server, check-schema-enabled, check-syntax-enabled, config-magic-number, db-batched-transaction-count, db-cache-size, db-checkpoint-interval, db-env-path, db-lock-count, db-log-buf-size, db-log-path, def-repl-manager-pwd, def-repl-manager-pwd-file, dn-cache-count, dn-cache-size, dsml-answer-size, dsml-buffer-size, dsml-client-auth-mode, dsml-enabled, dsml-max-parser-count, dsml-min-parser-count, dsml-port, dsml-relative-root-url, dsml-request-max-size, dsml-secure-port, file-descriptor-count, heap-high-threshold-size, heap-low-threshold-size, host-access-dir-path, idle-timeout, import-cache-size, instance-path, ldap-port, ldap-secure-port, listen-address, look-through-limit, max-psearch-count, max-thread-count, max-thread-per-connection-count, mod-tracking-enabled, pwd-accept-hashed-pwd-enabled, pwd-check-enabled, pwd-compat-mode, pwd-expire-no-warning-enabled, pwd-expire-warning-delay, pwd-failure-count-interval, pwd-grace-login-limit, pwd-keep-last-auth-time-enabled, pwd-lockout-duration, pwd-lockout-enabled, pwd-lockout-repl-priority-enabled, pwd-max-age, pwd-max-failure-count, pwd-max-history-count, pwd-min-age, pwd-min-length, pwd-mod-gen-length, pwd-must-change-enabled, pwd-root-dn-bypass-enabled, pwd-safe-modify-enabled, pwd-storage-scheme, pwd-strong-check-dictionary-path, pwd-strong-check-enabled, pwd-strong-check-require-charset, pwd-supported-storage-scheme, pwd-user-change-enabled, read-write-mode, ref-integrity-attr, ref-integrity-check-delay, ref-integrity-enabled, repl-user-schema-enabled, require-bind-pwd-enabled, retro-cl-deleted-entry-attr, retro-cl-enabled, retro-cl-ignored-attr, retro-cl-max-age, retro-cl-max-entry-count, retro-cl-path, retro-cl-suffix-dn, root-dn, root-pwd, root-pwd-file, root-pwd-storage-scheme, search-size-limit, search-time-limit, secure-listen-address, ssl-cipher-family, ssl-client-auth-mode, ssl-enabled, ssl-rsa-cert-name, ssl-rsa-security-device, ssl-supported-ciphers, thread-count – DS server instance configuration (SER) properties

Description The behavior of a Directory Server instance is configured according to server properties documented here and in the documentation specified under the SEE ALSO section.

PROPERTY: `check-schema-enabled`

Syntax	`on | off`
Default Value	`on`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether the server checks that entries being updated still conform to the server schema.

PROPERTY: `check-syntax-enabled`

Syntax	on | off
Default Value	off
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether the server checks that attribute values being updated have valid syntax. The server logs an error message when encountering an invalid value and prevents the update. When this property is set to on, the server checks updates to attribute values defined as Boolean, DN, Directory String, Generalized Time, IA5 String, INTEGER, or Telephone Number syntax. This behavior holds both for offline import and for normal write operations.

By default, syntax checking is off. When syntax checking is on, all import and update operations are checked. Directory Manager (directory super user) cannot bypass syntax checking.

Syntax is not checked on existing entries in the database. To clean up existing data, dump the database to LDIF, turn syntax checking on, and reload the database. Data that violates the syntax is visible in the errors log, and can be corrected and reloaded. You can also repair existing bad data by deleting or replacing the bad value using an LDAP client. If syntax checking is on, when a database is reloaded from LDIF, invalid syntax values are skipped and recorded in the errors log. Valid syntax values are reloaded.

PROPERTY: `config-magic-number`

Syntax	STRING
Default Value	D-A00
Is readable	Yes
Is modifiable	No
Is multi-valued	No

This property specifies a value used by the Directory Server administration framework and tools to determine the capabilities of a server instance.

PROPERTY: `db-batched-transaction-count`

Syntax	INTEGER
Default Value	0
Is readable	Yes

Is modifiable	Yes
Is multi-valued	No

This property specifies how many server transactions are gathered into a batch before being written to the transaction log. If writes to the transaction log are a bottleneck, you may potentially improve performance by increasing this value. Valid range is 0-30, 0 meaning that batching is turned off.

PROPERTY: `db-cache-size`

Syntax	`MEMORY_SIZE`
Default Value	`32M`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the amount of physical memory Directory Server requests from the operating system to cache indexes for all suffixes supported by the server instance. See *Directory Server Data Caching* in *Directory Server Enterprise Edition Reference* for suggestions on sizing cache.

PROPERTY: `db-checkpoint-interval`

Syntax	`DURATION`
Default Value	`60s`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the interval between checkpoints recorded in the database transaction log.

PROPERTY: `db-env-path`

Syntax	`PATH`
Default Value	*instance-path*`/db`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies a valid directory, unique to the server instance, on a `tmpfs` file system used to limit the time spent flushing pages for a server instance handling a high write load. There must be enough space available on the `tmpfs` file system to house at least the actual size of the database cache.

When changing this property, you must stop the server, delete the existing database, and reimport all suffixes from LDIF, before restarting the server.

PROPERTY: `db-lock-count`

Syntax	`INTEGER`
Default Value	`20000`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the number of locks available to the server instance database. Increase this value if you observe the following message in the `errors` log:

```
libdb: Lock table is out of available locks
```

PROPERTY: `db-log-buf-size`

Syntax	`MEMORY_SIZE`
Default Value	`512k`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the transaction log buffer size. Valid range is 0 to the size of the transaction log, which is 10M by default.

After changing this property, you must restart the server in order to take the change into account.

PROPERTY: `db-log-path`

Syntax	`PATH`
Default Value	*instance-path*`/db`
Is readable	Yes
Is modifiable	Yes

Is multi-valued	No

This property specifies the file system directory containing the database transaction log.

When changing this property, you must stop the server, delete the existing database, and reimport all suffixes from LDIF, before restarting the server.

PROPERTY: `def-repl-manager-pwd`

Syntax	`STRING`
Default Value	See the description that follows.
Is readable	Yes
Is modifiable	No
Is multi-valued	No

This property lets you read the password used for replication binds performed using simple authentication. Either you specify the password before setting up replication by setting `def-repl-manager-pwd-file` to specify the file containing the password you want to use, or you accept the password value generated by the `dsconf accord-replication` subcommand.

PROPERTY: `def-repl-manager-pwd-file`

Syntax	`PATH | ""`
Default Value	`""`
Is readable	No
Is modifiable	Yes
Is multi-valued	No

This property specifies the file from which the default replication password is read and stored for future use when setting up replication.

PROPERTY: `dn-cache-count`

Syntax	`INTEGER | unlimited | disabled`
Default Value	`unlimited`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the size of the DN cache in terms of number of entries. The value of dn-cache-count is unlimited by default. The value of dn-cache-count can be an integer, unlimited, and disabled and each of these has the following effect on dn-cache-size.

- unlimited — cache is limited to the cache size specified for dn-cache-size.
- disabled — caching is disabled and dn-cache-size is ignored.
- INTEGER — cache is limited to the number of DNs specified by the value that you provide and dn-cache-size is ignored. The value must be 1 or greater than 1.

Changing this property requires you to restart the server.

PROPERTY:
dn-cache-size

Syntax	MEMORY_SIZE
Default Value	10M
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the size of the DN cache in terms of memory space. This property is set by default. The cache size must be larger than 1M. The DN cache size specified for this property is taken into account only when dn-cache-count is set to unlimited.

Changing this property requires you to restart the server.

PROPERTY:
dsml-answer-size

Syntax	MEMORY_SIZE
Default Value	64k
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the maximum size of a server response to a DSML request. Larger responses are chunked.

PROPERTY:
dsml-buffer-size

Syntax	MEMORY_SIZE
Default Value	8k
Is readable	Yes

Is modifiable	Yes
Is multi-valued	No

This property specifies the size of the buffer used to store DSML requests. If the server receives many DSML requests larger than this limit, increase the buffer size.

PROPERTY: dsml-client-auth-mode

Syntax	clientCertOnly | httpBasicOnly | clientCertFirst
Default Value	httpBasicOnly
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies how the server identifies a client application. The following settings are supported.

clientCertOnly — Use credentials from the client certificate to identify the client.

httpBasicOnly — Use credentials from the HTTP authorization header to identify the client.

clientCertFirst — Attempt to use the client certificate credentials to identify the client. If there are no client certificate credentials, credentials from the HTTP authorization header are used.

PROPERTY: dsml-enabled

Syntax	on | off
Default Value	off
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether the server accepts DSML requests.

PROPERTY: dsml-max-parser-count

Syntax	INTEGER
Default Value	5
Is readable	Yes

Is modifiable	Yes
Is multi-valued	No

This property specifies the maximum number of DSML parsers allocated to handle client requests. Increase the value of this property if the server must handle sustained, high numbers of DSML client requests.

PROPERTY: `dsml-min-parser-count`

Syntax	`INTEGER`
Default Value	`10`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the minimum number of DSML parsers allocated to handle client requests. Increase the value of this property if the server must handle sustained, high numbers of DSML client requests.

PROPERTY: `dsml-port`

Syntax	`INTEGER	disabled`
Default Value	`disabled`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the port number on which the server listens for DSML requests. Changing the value requires that you restart the server.

PROPERTY: `dsml-relative-root-url`

Syntax	`STRING`
Default Value	`/dsml`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the root URL HTTP clients should specify in their POST requests.

PROPERTY: `dsml-request-max-size`

Syntax	`MEMORY_SIZE`
Default Value	`32k`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the maximum size for DSML client requests.

PROPERTY: `dsml-secure-port`

Syntax	`INTEGER | disabled`
Default Value	`disabled`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the port number on which the server listens for DSML requests over HTTPS. Changing the value requires that you restart the server.

PROPERTY: `file-descriptor-count`

Syntax	`INTEGER`
Default Value	`1024`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the maximum number of file descriptors the server instance attempts to use to handle client requests. Increase this value if you observe the following message in the `errors` log:

```
Not listening for new connections -- too many fds open
```

PROPERTY: `heap-high-threshold-size`

Syntax	`MEMORY_SIZE | undefined`
Default Value	`undefined`
Is readable	Yes

Is modifiable	Yes
Is multi-valued	No

This property specifies a threshold value for the dynamic memory footprint. When the threshold memory is reached, Directory Server attempts to free memory from the entry caches, and to limit memory use.

- When `heap-low-threshold-size` is reached, Directory Server attempts to free memory concurrently with other operations.
- When `heap-high-threshold-size` is reached, Directory Server prevents operations on the cache while memory is freed.

`heap-high-threshold-size` and `heap-low-threshold-size` must be configured in conjunction with each other, as follows.

- If `heap-high-threshold-size` is set to `undefined` or is not set, `heap-low-threshold-size` is ignored.
- If `heap-high-threshold-size` is set, its value must be at least one gigabyte.
- If `heap-high-threshold-size` is set, the value of `heap-low-threshold-size` must be less than that of `heap-high-threshold-size`. If not, `heap-low-threshold-size` is automatically set by default to 7/8 of the value of `heap-high-threshold-size`.
- If `heap-high-threshold-size` is set to a value other than `undefined`, `heap-low-threshold-size` is automatically set by default to 7/8 of the value of `heap-high-threshold-size`.
- If `heap-high-threshold-size` and `heap-low-threshold-size` are both set to a value other than `undefined`, `heap-low-threshold-size` must be greater than or equal to `(heap-high-threshold-size + minheap)/2`, where `minheap` is the amount of heap memory used by the server at startup. If this condition is not met, `heap-low-threshold-size` is automatically set by default to 7/8 of the value of `heap-high-threshold-size`.

The number of times the memory thresholds have been exceeded can be monitored by using the `heapmaxhighhits` and `heapmaxlowhits` attributes on `cn=monitor`.

PROPERTY: `heap-low-threshold-size`

Syntax	`MEMORY_SIZE | undefined`
Default Value	`undefined`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

See the description for `heap-high-threshold-size`.

PROPERTY: `host-access-dir-path`

Syntax	`PATH	""`
Default Value	`""`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the local directory path on the server host where `hosts.allow` and `hosts.deny` files are located. If this property is not set, or if the files are not found, Directory Server does not enable the additional connection-based access controls provided by these files.

PROPERTY: `idle-timeout`

Syntax	`INTEGER	none`
Default Value	`none`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies how many seconds the server waits for traffic on an idle LDAP client connection before closing the connection.

PROPERTY: `import-cache-size`

Syntax	`MEMORY_SIZE`
Default Value	`64M`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the amount of physical memory Directory Server requests from the operating system to cache data used when initializing a suffix from LDIF. See *Directory Server Data Caching* in *Directory Server Enterprise Edition Reference* for suggestions on sizing cache.

PROPERTY: `instance-path`

Syntax	`PATH`

Default Value	Path set at server creation
Is readable	Yes
Is modifiable	No
Is multi-valued	No

This property specifies the file system directory under which the server instance was created using the `dsadm create` command.

PROPERTY: `ldap-port`

Syntax	`INTEGER | disabled`
Default Value	`389 | 1389`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the port on which the server listens for LDAP client requests. The default port is 389 when the instance is created by the system super user, 1389 otherwise. Changing this property requires that you restart the server.

If you set both `ldap-port` and `ldap-secure-port` to `disabled`, you can no longer use `dsconf` to configure the server.

PROPERTY: `ldap-secure-port`

Syntax	`INTEGER | disabled`
Default Value	`636 | 1636`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the port on which the server listens for LDAPS client requests using TLS or SSL. The default port is 636 when the instance is created by the system super user, 1636 otherwise. Changing this property requires that you restart the server.

If you set both `ldap-port` and `ldap-secure-port` to `disabled`, you can no longer use `dsconf` to configure the server.

PROPERTY: `listen-address`

Syntax	`STRING`
Default Value	`0.0.0.0`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	Yes

This property specifies the IP address at which the server listens for LDAP client requests using the regular LDAP port. You can specify more than one listen address for the same port number. The default listen address is 0.0.0.0. Changing this property requires that you restart the server.

PROPERTY: `look-through-limit`

Syntax	`INTEGER | unlimited`
Default Value	`5000`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the maximum number of entries the server examines when checking candidates to respond to a search request.

PROPERTY: `max-psearch-count`

Syntax	`INTEGER`
Default Value	`30`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the maximum number persistent searches allowed. You can read the number of active persistent searches in the value of `currentpsearches` on `cn=monitor`.

PROPERTY: `max-thread-count`

Syntax	`INTEGER`
Default Value	`30`
Is readable	Yes

Is modifiable	Yes
Is multi-valued	No

This property specifies the number of threads created at startup to process operations. When tuning server performance, try setting this to twice the number of processors or 20 plus the number of simultaneous updates expected. You can read the number of active threads in the value of `threads` on `cn=monitor`.

PROPERTY: `max-thread-per-connection-count`

Syntax	`INTEGER`
Default Value	`5`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the maximum number of concurrent threads used to process operations on a single connection.

PROPERTY: `mod-tracking-enabled`

Syntax	`on | off`
Default Value	`on`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether the server maintains modification timestamps for updated entries.

PROPERTY: `pwd-accept-hashed-pwd-enabled`

Syntax	`on | off`
Default Value	N/A
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether the server accepts modifications with hashed password values without checking their content. This property takes effect only when `pwd-check-enabled` is `on`.

PROPERTY: `pwd-check-enabled`

Syntax	`on	off`
Default Value	`off`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether the server checks the quality of password values when they are modified.

PROPERTY: `pwd-compat-mode`

Syntax	`DS5-compatible-mode	DS6-migration-mode	DS6-mode`
Default Value	`DS5-compatible-mode`
Is readable	Yes
Is modifiable	No
Is multi-valued	No

This property specifies the password policy compatibility mode for the server. Change it using `dsconf pwd-compat`. See *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide* for details on password policy.

PROPERTY: `pwd-expire-no-warning-enabled`

Syntax	`on	off`
Default Value	`on`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether a password can expire without prior warning to a client application.

PROPERTY: `pwd-expire-warning-delay`

Syntax	`DURATION	disabled`

Default Value	1d
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the duration preceding password expiration during which the server returns warnings about the password expiring to client applications binding using the password.

PROPERTY: pwd-failure-count-interval

Syntax	DURATION | disabled
Default Value	10m
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the age beyond which password failures are purged from the failure count.

PROPERTY: pwd-grace-login-limit

Syntax	INTEGER | disabled
Default Value	disabled
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the number of times an expired password can be used to authenticate.

PROPERTY: pwd-keep-last-auth-time-enabled

Syntax	on | off
Default Value	off
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether to record authentication times in the pwdLastAuthTime operational attribute on user entries.

PROPERTY: pwd-lockout-duration

Syntax	DURATION | disabled
Default Value	1h
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the how long before the server unlocks an account that is locked.

PROPERTY: pwd-lockout-enabled

Syntax	on | off
Default Value	off
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether the server locks accounts after a specified number, pwd-max-failure-count, of consecutive failed attempts to bind.

PROPERTY: pwd-lockout-repl-priority-enabled

Syntax	on | off
Default Value	off
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether password lockout attributes are replicated with high priority.

PROPERTY: pwd-max-age

Syntax	DURATION | disabled
Default Value	disabled
Is readable	Yes

Is modifiable	Yes
Is multi-valued	No

This property specifies the age beyond which a password expires.

PROPERTY:
`pwd-max-failure-count`

Syntax	`INTEGER | disabled`
Default Value	`3`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the number of consecutive failed bind attempts after which the password may not be used to authenticate to the server.

PROPERTY:
`pwd-max-history-count`

Syntax	`INTEGER | disabled`
Default Value	`disabled`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the number of password values stored in the password history of the entry. These values cannot be used again until they are no longer present in the history.

PROPERTY:
`pwd-min-age`

Syntax	`DURATION | disabled`
Default Value	`disabled`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the minimum duration between password modifications.

PROPERTY: pwd-min-length

Syntax	INTEGER | disabled
Default Value	6
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the minimum number of characters allowed in a password value when quality checking has been enabled.

PROPERTY: pwd-mod-gen-length

Syntax	INTEGER | disabled
Default Value	6
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the length of the password generated by Directory Server when a password is reset using the LDAP Password Modify Extended Operation defined in RFC 3062 and no new password value is specified.

Although the syntax for this property is integer, its value must be between 6 and 512, inclusive.

PROPERTY: pwd-must-change-enabled

Syntax	on | off
Default Value	off
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether the password must be changed after the initial client bind after the password has been set or reset by another user.

PROPERTY: pwd-root-dn-bypass-enabled

Syntax	on | off
Default Value	off

Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether the directory super user is allowed to update passwords with values that violate password policy.

PROPERTY: `pwd-safe-modify-enabled`

Syntax	`on | off`
Default Value	`off`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether the current password must be provided with the request to modify the password.

PROPERTY: `pwd-storage-scheme`

Syntax	`STRING`
Default Value	`SSHA`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the algorithm used to encode password values.

PROPERTY: `pwd-strong-check-dictionary-path`

Syntax	`PATH | none`
Default Value	*install-path*`/ds6/plugins/words-english-big.txt`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the path to the dictionary file used for strong password checks.

PROPERTY: pwd-strong-check-enabled

Syntax	`on	off`
Default Value	`off`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether the server checks new password values to ensure they match with `pwd-strong-check-require-charset` settings, and do not match records in the dictionary file.

PROPERTY: pwd-strong-check-require-charset

Syntax	`lower	upper	digit	special	any-two	any-three`
Default Value	`lower && upper && digit && special`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	Yes

This property specifies the sets of characters that must be present in a password value modification.

`lower` The new password must include a lower case character.

`upper` The new password must include an upper case character.

`digit` The new password must include a digit.

`special` The new password must include a special character.

`any-two` The new password must include at least one character from each of at least two of the abovementioned character sets.

`any-three` The new password must include at least one character from each of at least three of the abovementioned character sets.

PROPERTY: pwd-supported-storage-scheme

Syntax	`STRING`
Default Value	See the following description.
Is readable	Yes
Is modifiable	No

Is multi-valued	Yes

This property specifies the set of encryption storage schemes supported for Directory Server user passwords. Supported storage schemes include `CRYPT`, `SHA`, `SSHA`, `NS-MTA-MD5`, and `CLEAR`.

PROPERTY: `pwd-user-change-enabled`

Syntax	`on | off`
Default Value	`on`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether users may change their own passwords.

PROPERTY: `read-write-mode`

Syntax	`read-only | read-write | frozen`
Default Value	`read-write`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether the suffixes and configuration data on the server can be modified. Use `frozen` when quiescing a server for online file system backup.

PROPERTY: `ref-integrity-attr`

Syntax	`ATTR_NAME | ""`
Default Value	`""`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies attributes for which referential integrity must be checked on update.

PROPERTY: `ref-integrity-check-delay`

Syntax	`DURATION | undefined`

Default Value	`undefined`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the delay between referential integrity checks. The default is no delay.

PROPERTY: `ref-integrity-enabled`

Syntax	`on | off`
Default Value	`off`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether referential integrity checks are performed by the server.

PROPERTY: `repl-user-schema-enabled`

Syntax	`on | off`
Default Value	`off`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether only schema elements with `X-ORIGIN` of `user-defined` are replicated. This can be useful when replicating between server versions with schema that are not fully compatible.

PROPERTY: `require-bind-pwd-enabled`

Syntax	`on | off`
Default Value	`on`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether the server rejects simple authentication attempts to bind that do not include a password.

PROPERTY: `retro-cl-deleted-entry-attr`

Syntax	`ATTR_NAME	""`
Default Value	`""`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	Yes

This property specifies the attributes to record in the retro change log when an entry is deleted.

PROPERTY: `retro-cl-enabled`

Syntax	`on	off`
Default Value	`off`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether the server maintains a retro changelog of all changes occurring on the server instance.

PROPERTY: `retro-cl-ignored-attr`

Syntax	`ATTR_NAME	""`
Default Value	`""`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the list of attributes not to record in the retro changelog when updates occur.

PROPERTY: `retro-cl-max-age`

Syntax	`DURATION	undefined`
Default Value	`undefined`
Is readable	Yes
Is modifiable	Yes

Is multi-valued	No

This property specifies the maximum age of records in the retro changelog. Older records are purged.

PROPERTY: `retro-cl-max-entry-count`

Syntax	`INTEGER`
Default Value	`0`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the maximum number of records in the retro changelog. Older records are purged. The value 0 corresponds to an unlimited number.

PROPERTY: `retro-cl-path`

Syntax	`PATH`
Default Value	*instance-path*`/db/changelog`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the file system directory in which the changelog is created.

PROPERTY: `retro-cl-suffix-dn`

Syntax	`DN | undefined`
Default Value	`undefined`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	Yes

This property specifies the suffixes for which retro changelog records are maintained.

PROPERTY: `root-dn`

Syntax	`DN`

Default Value	`cn=Directory Manager`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the Distinguished Name of the Directory Manager user, a user not subject to access controls.

PROPERTY: `root-pwd`

Syntax	`STRING`
Default Value	None
Is readable	Yes
Is modifiable	No
Is multi-valued	No

This property specifies the password for the Directory Manager user. It is show hashed according to the password storage scheme used.

PROPERTY: `root-pwd-file`

Syntax	`PATH | ""`
Default Value	`""`
Is readable	No
Is modifiable	Yes
Is multi-valued	No

This property specifies the file containing the password for the Directory Manager user. The file is read once, and the password is stored for future use.

PROPERTY: `root-pwd-storage-scheme`

Syntax	`STRING`
Default Value	SSHA
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the algorithm used to encrypt the password for the Directory Manager user. It must be one of the schemes specified by the `pwd-supported-storage-scheme` property.

PROPERTY: `search-size-limit`

Syntax	`INTEGER	unlimited`
Default Value	`2000`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the maximum number of entries the server returns for a search operation.

PROPERTY: `search-time-limit`

Syntax	`INTEGER	unlimited`
Default Value	`3600`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the maximum number of seconds allocated by the server to respond to a search request.

PROPERTY: `secure-listen-address`

Syntax	`STRING`
Default Value	`0.0.0.0`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	Yes

This property specifies the IP address at which the server listens for LDAP client requests using the secure LDAP port. You can specify more than one secure listen address for the same port number. The default secure listen address is 0.0.0.0. Changing this property requires that you restart the server.

PROPERTY: `ssl-cipher-family`

Syntax	`STRING	all`
Default Value	`all`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	Yes

This property specifies the SSL ciphers the server can use for SSL communications. The default value, `all`, does not mean all the supported SSL ciphers, as supported ciphers with NULL key length are removed from the list.

PROPERTY: `ssl-client-auth-mode`

Syntax	`allowed	required	disabled`
Default Value	`allowed`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether the server allows, requires, or does not allow SSL client authentication, in which the client application authenticates sending its SSL certificate to the server.

PROPERTY: `ssl-enabled`

Syntax	`on	off`
Default Value	`off`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether the server accepts SSL connnections.

PROPERTY: `ssl-rsa-cert-name`

Syntax	`STRING`
Default Value	defaultCert
Is readable	Yes
Is modifiable	Yes

Is multi-valued	No

This property specifies the name of the SSL certificate for the server.

PROPERTY: `ssl-rsa-security-device`

Syntax	`STRING`
Default Value	`internal (software)`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the name of the security device used by the server.

PROPERTY: `ssl-supported-ciphers`

Syntax	`STRING`
Default Value	Depends on underlying SSL library
Is readable	Yes
Is modifiable	No
Is multi-valued	No

This property specifies the full list of SSL ciphers the server can support.

Description

Syntax values shown in lower case or partly in lower case are literal values.

Those shown in upper case are syntax types, defined as follows:

`ATTR_NAME`
A valid attribute type name such as `cn` or `objectClass`.

`BOOLEAN`
`true` or `false`.

`DN`
A valid distinguished name such as `ou=People,dc=example,dc=com`.

`DURATION`
A duration specified in months (`M`), weeks (`w`), days (`d`), hours (`h`), minutes (`m`), seconds (`s`), and miliseconds (`ms`), or some combination with multiple specifiers. For example, you can specify one week as `1w`, `7d`, `168h`, `10080m`, or `604800s`. You can also specify one week as `1w0d0h0m0s`.

DURATION properties typically do not each support all duration specifiers (Mwdhms). Examine the output of dsconf help-properties for the property to determine which duration specifiers are supported.

EMAIL_ADDRESS
A valid e-mail address.

HOST_NAME
An IP address or host name.

INTEGER
A positive integer value between 0 and the maximum supported integer value in the system address space. On 32-bit systems, 2147483647. On 64-bit systems, 9223372036854775807.

INTERVAL
An interval value of the form *hhmm-hhmm 0123456*, where the first element specifies the starting hour, the next element the finishing hour in 24-hour time format, from 0000-2359, and the second specifies days, starting with Sunday (0) to Saturday (6).

IP_RANGE
An IP address or range of address in one of the following formats:

- IP address in dotted decimal form.
- IP address and bits, in the form of network number/mask bits.
- IP address and quad, in the form of a pair of dotted decimal quads.
- All address. A catch-all for clients that are note placed into other, higher priority groups.
- 0.0.0.0. This address is for groups to which initial membership is not considered. For example, for groups that clients switch to after their initial bind.
- IP address of the local host.

LDAP_URL
A valid LDAP URL as specified by RFC 2255 (http://www.ietf.org/rfc/rfc2255.txt).

MEMORY_SIZE
A memory size specified in gigabytes (G), megabytes (M),kilobytes (k), or bytes (b). Unlike DURATION properties, MEMORY_SIZE properties cannot combine multiple specifiers. However, MEMORY_SIZE properties allow decimal values, for example, 1.5M.

NAME
A valid cn (common name).

OCTAL_MODE
A three-digit, octal file permissions specifier. The first digit specifies permissions for the server user ID, the second for the server group ID, the last for other users. Each digit consists of a bitmask defining read (4), write (2), execute (1), or no access (0) permissions, thus 640 specifies read-write access for the server user, read-only access for other users of the server group, and no access for other users.

`PASSWORD_FILE`
The full path to the file from which the bind password should be read.

`PATH`
A valid, absolute file system path.

`STRING`
A DirectoryString value, as specified by RFC 2252 (`http://www.ietf.org/rfc/rfc2252.txt`).

`SUPPORTED_SSL_CIPHER`
An SSL cipher supported by the server. See the Reference for a list of supported ciphers.

`SUPPORTED_SSL_PROTOCOL`
An SSL protocol supported by the server. See the Reference for a list of supported protocols.

`TIME`
A time of the form *hhmm* in 24-hour format, where *hh* stands for hours and *mm* stands for minutes.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Evolving

See Also

dsconf(1M), all-ids-threshold(5dsconf), db-path(5dsconf), moddn-enabled(5dsconf), referral-url(5dsconf)

Name

suffix, db-name, entry-cache-count, entry-cache-size, entry-count, parent-suffix-dn, referral-mode, repl-accept-client-update-enabled, repl-cl-max-age, repl-cl-max-entry-count, repl-id, repl-manager-bind-dn, repl-purge-delay, repl-rewrite-referrals-enabled, repl-role, require-index-enabled – DS suffix configuration (SUF) properties

Description

Each Directory Server suffix you create is configured according to the suffix properties documented here and in the documentation specified under the SEE ALSO section.

PROPERTY: `db-name`

Syntax	`STRING`
Default Value	*suffixName*
Is readable	Yes
Is modifiable	No
Is multi-valued	Yes

This property specifies the the suffix used to process requests involving the database.

PROPERTY: `entry-cache-count`

Syntax	`INTEGER | unlimited`
Default Value	`unlimited`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the number of entries allowed in the entry cache of the suffix.

PROPERTY: `entry-cache-size`

Syntax	`MEMORY_SIZE`
Default Value	`10M`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the maximum amount of memory Directory Server requests for the entry cache of the suffix.

PROPERTY: `entry-count`

Syntax	`INTEGER`
Default Value	`0`
Is readable	Yes
Is modifiable	No
Is multi-valued	No

This property specifies the number of entries stored in the suffix.

PROPERTY: `parent-suffix-dn`

Syntax	`DN | undefined`
Default Value	DN of the parent entry
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the DN of the parent entry for the suffix. The value of this property must logically be a parent of the suffix.

For instance, if you have a suffix with DN `dc=com` and a suffix `dc=example,dc=com`, you can set `dc=com` as the `parent-suffix-dn` of `dc=example,dc=com`, and subtree searches with based DN `dc=com` then also travers `dc=example,dc=com`.

PROPERTY: `referral-mode`

Syntax	`disabled | enabled | only-on-write`
Default Value	`disabled`
Is readable	Yes
Is modifiable	Yes, if the suffix is not replicated
Is multi-valued	No

This property specifies how referrals are used when a client makes a request involving the suffix.

`disabled` Handle requests locally; do not return referral URLs.

`enabled` Return referral URLs to client requests.

`only-on-write` Return referral URLs to client requests only for write operations.

PROPERTY: repl-accept-client-update-enabled

Syntax	on | off
Default Value	on
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether the replicated suffix accepts write operations from client applications, or instead returns referral URLs.

PROPERTY: repl-cl-max-age

Syntax	DURATION | undefined
Default Value	7d
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the maximum age for a record in the replication changelog. Older records are purged.

PROPERTY: repl-cl-max-entry-count

Syntax	INTEGER
Default Value	0 (meaning undefined)
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the maximum number of records in the replication changelog. When the limit is exceeded, older records are purged.

PROPERTY: repl-id

Syntax	INTEGER
Default Value	None
Is readable	Yes
Is modifiable	Yes, using the subcommands to manage replication

Is multi-valued	No

This property specifies the replica identification number, 1-65534 for a supplier, 65535 for a consumer or a hub. Once set, this property cannot be modified.

PROPERTY: `repl-manager-bind-dn`

Syntax	`DN	undefined`
Default Value	`undefined`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	Yes

This property specifies the bind DNs of users allowed to bind to perform replication operations on the suffix.

PROPERTY: `repl-purge-delay`

Syntax	`DURATION	never`
Default Value	`7d`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies the maximum age of tombstone entries used by replication. Tombstone entries are entries marked for deletion that have not yet been removed, and also replication state information associated with the entries. When setting this attribute, ensure that the purge delay is longer than the longest replication cycle in your replication policy to avoid incurring conflict resolution problems and divergence between replicas.

PROPERTY: `repl-rewrite-referrals-enabled`

Syntax	`on	off`
Default Value	`off`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property specifies whether referrals you set can be automatically overwritten by the server to reference replicas.

PROPERTY: `repl-role`

Syntax	`not-replicated	master	hub	consumer`
Default Value	`not-replicated`
Is readable	Yes
Is modifiable	Yes, using the subcommands to manage replication
Is multi-valued	No

This property specifies the role played by the suffix in a replicated topology.

`not-replicated` The suffix is not part of a replicated toplogy.

`master` This suffix is a supplier of replication updates in a replicated topology. It can accept both read and write operations.

`hub` This suffix is a supplier of replication updates in a replicated topology. It can accept read operations and replication updates.

`consumer` This suffix is a dedicated consumer of replication updates in a replicated topology. It can accept read operations and replication updates, but not writes from clients.

To promote a replica, use the `dsconf promote-repl` command. To demote a replica, use the `dsconf demote-repl` command.

PROPERTY: `require-index-enabled`

Syntax	`on|off`
Default Value	`off`
Is readable	Yes
Is modifiable	Yes
Is multi-valued	No

This property determines whether unindexed searches are allowed. When `on`, unindexed searches return `LDAP_UNWILLING_TO_PERFORM`.

Description

Syntax values shown in lower case or partly in lower case are literal values.

Those shown in upper case are syntax types, defined as follows:

`ATTR_NAME`
A valid attribute type name such as `cn` or `objectClass`.

`BOOLEAN`
`true` or `false`.

DN

A valid distinguished name such as ou=People,dc=example,dc=com.

DURATION

A duration specified in months (M), weeks (w), days (d), hours (h), minutes (m), seconds (s), and miliseconds (ms), or some combination with multiple specifiers. For example, you can specify one week as 1w, 7d, 168h, 10080m, or 604800s. You can also specify one week as 1w0d0h0m0s.

DURATION properties typically do not each support all duration specifiers (Mwdhms). Examine the output of dsconf help-properties for the property to determine which duration specifiers are supported.

EMAIL_ADDRESS

A valid e-mail address.

HOST_NAME

An IP address or host name.

INTEGER

A positive integer value between 0 and the maximum supported integer value in the system address space. On 32-bit systems, 2147483647. On 64-bit systems, 9223372036854775807.

INTERVAL

An interval value of the form *hhmm-hhmm 0123456*, where the first element specifies the starting hour, the next element the finishing hour in 24-hour time format, from 0000-2359, and the second specifies days, starting with Sunday (0) to Saturday (6).

IP_RANGE

An IP address or range of address in one of the following formats:

- IP address in dotted decimal form.
- IP address and bits, in the form of network number/mask bits.
- IP address and quad, in the form of a pair of dotted decimal quads.
- All address. A catch-all for clients that are note placed into other, higher priority groups.
- 0.0.0.0. This address is for groups to which initial membership is not considered. For example, for groups that clients switch to after their initial bind.
- IP address of the local host.

LDAP_URL

A valid LDAP URL as specified by RFC 2255 (http://www.ietf.org/rfc/rfc2255.txt).

MEMORY_SIZE

A memory size specified in gigabytes (G), megabytes (M),kilobytes (k), or bytes (b). Unlike DURATION properties, MEMORY_SIZE properties cannot combine multiple specifiers. However, MEMORY_SIZE properties allow decimal values, for example, 1.5M.

`NAME`
A valid `cn` (common name).

`OCTAL_MODE`
A three-digit, octal file permissions specifier. The first digit specifies permissions for the server user ID, the second for the server group ID, the last for other users. Each digit consists of a bitmask defining read (`4`), write (`2`), execute (`1`), or no access (`0`) permissions, thus `640` specifies read-write access for the server user, read-only access for other users of the server group, and no access for other users.

`PASSWORD_FILE`
The full path to the file from which the bind password should be read.

`PATH`
A valid, absolute file system path.

`STRING`
A DirectoryString value, as specified by RFC 2252 (`http://www.ietf.org/rfc/rfc2252.txt`).

`SUPPORTED_SSL_CIPHER`
An SSL cipher supported by the server. See the Reference for a list of supported ciphers.

`SUPPORTED_SSL_PROTOCOL`
An SSL protocol supported by the server. See the Reference for a list of supported protocols.

`TIME`
A time of the form *hhmm* in 24-hour format, where *hh* stands for hours and *mm* stands for minutes.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory-client
Stability Level	Evolving

See Also

dsconf(1M), all-ids-threshold(5dsconf), db-path(5dsconf), enabled(5dsconf), moddn-enabled(5dsconf), referral-url(5dsconf)

Name

useAuthzIdForAuditAttrs – record proxied authorization information

Description

Specifies whether Directory Server records the authentication ID, such as the bind DN, of the proxy acting on behalf of the user, or the authorization ID of the user for whom the proxy is requesting the operation.

When `useAuthzIdForAuditAttrs` is set to `on`, Directory Server records the authorization ID in the `creatorsName` or `modifiersName` during a write operation on an entry. By default Directory Server records the authentication ID.

Entry DN	`cn=config`
Valid range	`on` | `off`
Default value	`off`
Syntax	DirectoryString
Example	`useAuthzIdForAuditAttrs: on`

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

See Also

dse.ldif(4)

REFERENCE

Directory Proxy Server Configuration

Name

aci-data-view – Directory Proxy Server configuration property

Description

Syntax	`dnReference`
Default value	disabled
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the data view that Directory Proxy Server uses to store access controls.

The value of this property is the name of one of the following configuration entities: jdbc-data-view, join-data-view, ldap-data-view, ldif-data-view.

The default behavior for this property is as follows: No virtual access control policy.

This property is used to configure the following features: `virtual-aci`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

aci-manager-bind-dn – Directory Proxy Server configuration property

Description

Syntax	`dn`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the distinguished name of the identity used by Directory Proxy Server to access the access control data view.

This property takes a Distinguished Name (DN) value.

The default behavior for this property is as follows: Proxy uses an anonymous access to access the access control data view.

This property is used to configure the following features: `virtual-aci`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

aci-manager-bind-pwd – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	No
Is multivalued	No
Requires restart	No

This property specifies the password of the identity used by Directory Proxy Server to access the access control data view.

This property is read-only. To change the password, use the `aci-manager-bind-pwd-file` property.

The default behavior for this property is as follows: Proxy uses an anonymous access to access the access control data view.

This property is used to configure the following features: `virtual-aci`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

aci-manager-bind-pwd-file – Directory Proxy Server configuration property

Description

Syntax	`password`
Default value	No default is defined.
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the file from which to read the password of the identity used by Directory Proxy Server to access the access control data view. The temporary file is read once, and the password is stored for future use.

This property takes a path to a file that contains the password value.

This property is used to configure the following features: `virtual-aci`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

aci-source – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the name of the set of access controls that will apply to the connection handler.

The default behavior for this property is as follows: ACIs are unset. Proxy will reject WRITE operations on non-LDAP data views.

This property is used to configure the following features: `connection-handler`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

action – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property defines the transformation action. A transformation action describe what the transformation does to its target entry or entries.

This property can take the following values in addition to the default.

`add-attr`
Add an attribute

`remove-attr`
Remove an attribute

`add-attr-value`
Add a value to an attribute

`def-value`
Set a default value

`remove-attr-value`
Remove a value of an attribute

`attr-value-mapping`
Attribute value mapping

This property is used to configure the following features: `virtual-transformation`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

add-weight – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	disabled
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the proportion of add requests that are sent to the attached data source.

This property takes an integer.

The value of this property must be at least 1.

This property can also take the following values:

`disabled`
Do not forward any add requests to the data source

This property is used to configure the following features: `attached-ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

allow-add-operations – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	true
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not associated connection handlers accept LDAP add operations.

This property is `true` or `false`.

This property is used to configure the following features: `request-filtering-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

allow-bind-operations – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	true
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not associated connection handlers accept LDAP bind operations.

This property is `true` or `false`.

This property is used to configure the following features: `request-filtering-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

allow-cert-based-auth – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	allow
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not clients must present their own certificates when establishing connections to Directory Proxy Server.

This property can take the following values in addition to the default.

`deny`
Clients are not allowed to use certificate based authentication

`allow`
Clients are allowed to use certificate based authentication

`require`
Clients must use certificate based authentication

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

allow-compare-operations – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	true
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not associated connection handlers accept LDAP compare operations.

This property is `true` or `false`.

This property is used to configure the following features: `request-filtering-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

allow-delete-operations – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	true
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not associated connection handlers accept LDAP delete operations.

This property is `true` or `false`.

This property is used to configure the following features: `request-filtering-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

allowed-auth-methods – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	anonymous
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies a set of authentication methods. Clients must use one of the specified authentication methods in order for the connection to be accepted by the connection handler.

This property can take the following values in addition to the default.

`anonymous`
Anonymous authentication

`simple`
Simple authentication

`sasl`
SASL/External authentication

This property is used to configure the following features: `connection-handler`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name allowed-comparable-attrs – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	all
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies a set of LDAP attribute types that can be compared in an LDAP search filter or compare operation.

The default behavior for this property is as follows: All attribute types can be compared

This property is used to configure the following features: `request-filtering-policy`.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also `dpconf`(1M)

Name

allowed-ldap-controls – Directory Proxy Server configuration property

Description

Syntax	`oid`
Default value	Default behavior is not defined.
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies a set of LDAP control OIDs. The control OIDs determine which LDAP controls are forwarded by Directory Proxy Server.

This property can take the following values in addition to the default.

`proxy-auth-v1`
Proxy authorization v1

`proxy-auth-v2`
Proxy authorization v2

`persistent-search`
Persistent search

`manage-dsa`
Manage DSA

`auth-request`
Authentication request

`real-attributes-only`
Real attributes only

`chaining-loop-detection`
Chaining loop detection

`vlv-request`
Virtual list view (VLV) request

`server-side-sorting`
Server side sorting

`get-effective-rights`
Get effective rights

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

allowed-ldap-ports – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	ldap
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies a set of IP port numbers. A client connection must come through one of the specified ports in order for the connection to be accepted by the connection handler.

This property can take the following values in addition to the default.

`ldap`
The LDAP port of Directory Proxy Server

`ldaps`
The LDAPS port of Directory Proxy Server

This property is used to configure the following features: `connection-handler`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

allowed-search-scopes – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	base
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies a set of LDAP search scopes that are accepted by associated connection handlers.

This property can take the following values in addition to the default.

`base`
: Base entry searches

`one-level`
: Base + first level searches

`subtree`
: Subtree searches

This property is used to configure the following features: `request-filtering-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

allowed-subtrees – Directory Proxy Server configuration property

Description

Syntax	`dn`
Default value	
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies a set of subtrees that can be accessed by clients.

This property takes a Distinguished Name (DN) value.

This property is used to configure the following features: `request-filtering-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

allow-extended-operations – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	true
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not associated connection handlers accept LDAP extended operations.

This property is `true` or `false`.

This property is used to configure the following features: `request-filtering-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

allow-inequality-search-operations – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	true
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not associated connection handlers accept LDAP searches based on inequality filters.

This property is `true` or `false`.

This property is used to configure the following features: `request-filtering-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

allow-ldapv2-clients – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	true
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not clients can connect to Directory Proxy Server by using LDAPv2.

This property is `true` or `false`.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

allow-modify-operations – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	true
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not associated connection handlers accept LDAP modify operations.

This property is `true` or `false`.

This property is used to configure the following features: `request-filtering-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

allow-persistent-searches – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	false
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not Directory Proxy Server allows clients to use persistent searches.

This property is `true` or `false`.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

allow-rename-operations – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	true
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not associated connection handlers accept LDAP modify DN operations.

This property is `true` or `false`.

This property is used to configure the following features: `request-filtering-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name allow-sasl-external-authentication – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	true
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not clients can authenticate to Directory Proxy Server by using SASL/External authentication.

This property is `true` or `false`.

This property is used to configure the following features: `server`.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also `dpconf`(1M)

Name allow-search-operations – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	true
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not associated connection handlers accept LDAP search operations.

This property is `true` or `false`.

This property is used to configure the following features: `request-filtering-policy`.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also `dpconf`(1M)

Name

allow-unauthenticated-operations – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	true
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not Directory Proxy Server allows anonymous clients to perform operations.

This property is `true` or `false`.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

alternate-search-base-dn – Directory Proxy Server configuration property

Description

Syntax	`dn`
Default value	Default behavior is not defined.
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies the DN of an alternate search base.

When an alternate search base is specified in a subordinate data view, search operations targeted at the superior data view are performed in both the superior data view and the subordinate data view.

By default, Directory Proxy Server automatically configures the alternate search base in the subordinate data view. However, the automatic configuration can be disabled and the feature can be configured manually.

This property takes a Distinguished Name (DN) value.

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

attr-name – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property provides the name of a virtual attribute.

This property is used to configure the following features: `jdbc-attr`, `virtual-transformation`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

attr-name-mappings – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property defines a list of attribute name mappings.

When a client makes a request, the mapped attributes are renamed to match the names on the server side. When the result is returned to the client, the attributes are renamed back to match the names on the client side.

The syntax of this string is `<client-attr>#<source-attr>`.

An attribute mapping of the form <client attribute>#<source attribute>

The value of this property must match the pattern `^[a-zA-Z][-a-zA-Z0-9]+#[a-zA-Z][-a-zA-Z0-9]+$`.

The default behavior for this property is as follows: No attribute name mappings

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

attrs – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies a list of attributes for a search data hiding rule. The rule-action(5dpconf) property defines whether the specified attributes are filtered out of the search result, or whether the unspecified attributes are filtered out of the search result.

The syntax of this string is `ATTR_NAME`.

An attribute name

The value of this property must match the pattern `^[a-zA-Z][-a-zA-Z0-9]+$`.

The default behavior for this property is as follows: No filtering is applied.

This property is used to configure the following features: `search-data-hiding-rule`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

base-dn – Directory Proxy Server configuration property

Description

Syntax	`dn`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the DN of the subtree represented by the data view.

This property takes a Distinguished Name (DN) value.

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

bind-dn – Directory Proxy Server configuration property

Description

Syntax	`dn`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the DN used by the proxy to bind to the LDAP data source when this data source is configured to use proxy authorization.

This property takes a Distinguished Name (DN) value.

The default behavior for this property is as follows: Proxy does not bind to the LDAP data source.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

bind-dn-filters – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	any
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies a set of regular expressions. The bind DN of a client must match at least one regular expression in order for the connection to be accepted by the connection handler.

The default behavior for this property is as follows: All client bind DNs are accepted

This property is used to configure the following features: `connection-handler`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

bind-pwd – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	No
Is multivalued	No
Requires restart	No

This property specifies the password used by the proxy to bind to the LDAP data source when this data source is configured to use proxy authorization.

This property is read-only. To change the password, use the `bind-pwd-file` property.

The default behavior for this property is as follows: The proxy will not use any password to bind to the LDAP data source

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

bind-pwd-attr – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	userPassword
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the name of the attribute used to contain authentication passwords.

This property is used to configure the following features: `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

bind-pwd-file – Directory Proxy Server configuration property

Description

Syntax	`password`
Default value	No default is defined.
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the file from which to read the password for proxy authorization. The temporary file is read once, and the password is stored for future use.

This property takes a path to a file that contains the password value.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

bind-weight – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	disabled
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the proportion of bind requests that are sent to the attached data source.

This property takes an integer.

The value of this property must be at least 1.

This property can also take the following values:

`disabled`
  Do not forward any bind requests to the data source

This property is used to configure the following features: `attached-ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

cert-data-view-routing-custom-list – Directory Proxy Server configuration property

Description

Syntax	`dnReference`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies the set of data views that Directory Proxy Server uses to find certificates if the cert-data-view-routing-policy(5dpconf) property is set to `custom`.

The value of this property is the name of one of the following configuration entities: jdbc-data-view, join-data-view, ldap-data-view, ldif-data-view.

The default behavior for this property is as follows: If the cert-data-view-routing-policy is custom, proxy has no route to map the certificate.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

cert-data-view-routing-policy – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	all-routable
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag that indicates whether or not Directory Proxy Server should use all routable data views or the list of data views specified by cert-data-view-routing-custom-list(5dpconf) when searching for certificates.

This property can take the following values in addition to the default.

`all-routable`
All routable data views

`custom`
Customized set of data views

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

cert-search-attr-mappings – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies mappings that map attributes in the certificate subject to attributes in the LDAP server.

The syntax of this string is `<subject-attr>:<user-attr>`.

An attribute mapping of the form <subject attribute>:<user attribute>

The value of this property must match the pattern `^[a-zA-Z][-a-zA-Z0-9]+:[a-zA-Z][-a-zA-Z0-9]+$`.

The default behavior for this property is as follows: No attributes in the certificate subject are mapped

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

cert-search-base-dn – Directory Proxy Server configuration property

Description

Syntax	`dn`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the base DN of a search operation that finds user entries when a user's name is not specified in their certificate.

This property takes a Distinguished Name (DN) value.

The default behavior for this property is as follows: This property is required when proxy should not use a user certificate subject as the user DN.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

cert-search-bind-dn – Directory Proxy Server configuration property

Description

Syntax	`dn`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies an optional identity to be used when searching for certificates.

This property takes a Distinguished Name (DN) value.

The default behavior for this property is as follows: Proxy uses an anonymous access to bind when searching for certificates.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

cert-search-bind-pwd – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	No
Is multivalued	No
Requires restart	No

This property specifies the password of the optional identity to be used when searching for certificates.

This property is read-only. To change the password, use the `cert-search-bind-pwd-file` property.

The default behavior for this property is as follows: Proxy uses an anonymous access to bind when searching for certificates.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

cert-search-bind-pwd-file – Directory Proxy Server configuration property

Description

Syntax	`password`
Default value	No default is defined.
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the file from which to read the password of the optional identity used when searching for certificates. The temporary file is read once, and the password is stored for future use.

This property takes a path to a file that contains the password value.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

cert-search-user-attr – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	userCertificate
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is the name of an LDAP attribute used to contain certificates in user entries.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

client-affinity-policy – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	write-affinity-after-write
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the algorithm that determines when requests from the same client should be directed to the same LDAP data source.

This property can take the following values in addition to the default.

`write-affinity-after-write`
Affinity for write requests after the first write request

`read-write-affinity-after-write`
Affinity for all requests after the first write request

`read-write-affinity-after-any`
Affinity for all requests after the first read request or write request

`read-affinity-after-write`
Affinity for the first read request after a write request

This property is used to configure the following features: `ldap-data-source-pool`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

client-affinity-timeout – Directory Proxy Server configuration property

Description

Syntax	`duration`
Default value	20000
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the duration of the client affinity.

The duration is expressed in milliseconds.

This property is used to configure the following features: `ldap-data-source-pool`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

client-cred-mode – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	use-client-identity
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies how client credentials are used to authenticate the client to an LDAP data source.

This property can take the following values in addition to the default.

`use-specific-identity`
Use the identity specified by the bind-dn(5dpconf) and bind-pwd(5dpconf) properties.

`use-client-identity`
Use the identity provided by the client.

`use-proxy-auth`
Use the identity specified by the bind-dn(5dpconf) and bind-pwd(5dpconf) properties, and include the client identity in the proxyAuth control.

`use-proxy-auth-for-write`
Use the identity specified by the bind-dn(5dpconf) and bind-pwd(5dpconf) properties, and include the client identity in the proxyAuth control for write operations only.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

compare-weight – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	disabled
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the proportion of compare requests that are sent to the attached data source.

This property takes an integer.

The value of this property must be at least 1.

This property can also take the following values:

`disabled`
Do not forward any compare requests to the data source

This property is used to configure the following features: `attached-ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

configuration-manager-bind-dn – Directory Proxy Server configuration property

Description

Syntax	`dn`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the distinguished name of the Proxy Manager that is the user allowed to manage the configuration of Directory Proxy Server.

This property takes a Distinguished Name (DN) value.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

configuration-manager-bind-pwd – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	No default is defined.
Must be set	No
Is modifiable	No
Is multivalued	No
Requires restart	No

This property specifies the password of the Proxy Manager that is the user allowed to manage the configuration of Directory Proxy Server.

This property is read-only. To change the password, use the `configuration-manager-bind-pwd-file` property.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

configuration-manager-bind-pwd-file – Directory Proxy Server configuration property

Description

Syntax	`password`
Default value	No default is defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the file from which to read the Proxy manager bind password. The temporary file is read once, and the password is stored for future use.

This property takes a path to a file that contains the password value of at least 8 characters in length.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

connection-idle-timeout – Directory Proxy Server configuration property

Description

Syntax	`duration`
Default value	3600
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum length of time a client connection can remain idle before being closed.

The duration is expressed in seconds.

This property is used to configure the following features: `ldap-listener`, `ldaps-listener`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

connection-pool-wait-timeout – Directory Proxy Server configuration property

Description

Syntax	`duration`
Default value	3000
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum length of time that Directory Proxy Server waits for a connection to an LDAP server to become available if a connection pool is empty when a request is made.

The duration is expressed in milliseconds.

The value of this property must be at least 1.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

connection-read-data-timeout – Directory Proxy Server configuration property

Description

Syntax	`duration`
Default value	2000
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum length of time that the listener can wait for new data to be available.

The duration is expressed in milliseconds.

This property is used to configure the following features: `ldap-listener`, `ldaps-listener`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

connection-write-data-timeout – Directory Proxy Server configuration property

Description

Syntax	`duration`
Default value	3600000
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum length of time that the listener can wait to send results back to clients.

The duration is expressed in milliseconds.

This property is used to configure the following features: `ldap-listener`, `ldaps-listener`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

connect-timeout – Directory Proxy Server configuration property

Description

Syntax	`duration`
Default value	10000
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum length of time that a connection between Directory Proxy Server and a data source is attempted before the connection attempt fails.

The duration is expressed in milliseconds.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

contains-shared-entries – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	false
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property determines what should be done if an entry in a secondary data view is used by more than one entry in a primary data view.

This property is applicable to secondary data views only.

If it is set to TRUE, the secondary data view entry is deleted when the virtual entry is deleted. If the entry does not exist in the secondary data view, it is created when the virtual entry is created.

This property is `true` or `false`.

This property is used to configure the following features: `jdbc-data-view`, `jdbc-table`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

custom-distribution-algorithm – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the class name to use for custom distribution algorithm. This property can be set only if `distribution-algorithm property` is set to `none`.

This property accepts the string value that contains Java class name. String value in `PackageName.AlgoClassName` format is valid.

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name data-source-read-timeout – Directory Proxy Server configuration property

Description

Syntax	`duration`
Default value	20000
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum length of time that Directory Proxy Server waits for a data source to complete a read request.

The duration is expressed in milliseconds.

This property is used to configure the following features: `server`.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also `dpconf`(1M)

Name data-view-automatic-routing-mode – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	automatic
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag that indicates whether Directory Proxy Server automatically routes requests, or whether Directory Proxy Server relies on manual routing configuration.

This property can take the following values in addition to the default.

`automatic`
Directory Proxy Server automatically routes requests and ignores data view exclusion bases and alternate search bases.

`limited`
Directory Proxy Server automatically routes requests but will take into consideration data view exclusion bases if present.

`manual`
Directory Proxy Server does not automatically route requests. Instead, Directory Proxy Server routes requests according to the exclusion bases and alternate search bases specified in the data views.

This property is used to configure the following features: `server`.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also `dpconf`(1M)

Name

data-view-routing-custom-list – Directory Proxy Server configuration property

Description

Syntax	`dnReference`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies the list of data views to which Directory Proxy Server routes client requests if data-view-routing-policy(5dpconf) is `custom`.

The value of this property is the name of one of the following configuration entities: jdbc-data-view, join-data-view, ldap-data-view, ldif-data-view.

The default behavior for this property is as follows: Proxy does not route requests.

This property is used to configure the following features: `connection-handler`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name data-view-routing-policy – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether Directory Proxy Server routes client requests to all routable data views or to a custom set of data views.

This property can take the following values in addition to the default.

`all-routable`
All routable data views

`custom`
Customized set of data views

This property is used to configure the following features: `connection-handler`.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also `dpconf`(1M)

Name

db-name – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the database name of the JDBC data source.

This property is used to configure the following features: `jdbc-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

db-pwd – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	No
Is multivalued	No
Requires restart	No

This property specifies the database user password of the JDBC data source.

This property is read-only. To change the password, use the `db-pwd-file` property.

The default behavior for this property is as follows: The proxy will not use any password to connect to the JDBC data source.

This property is used to configure the following features: `jdbc-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

db-pwd-encryption – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	clear-text
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the mechanism used to store authentication passwords.

This property can take the following values in addition to the default.

`clear-text`
Passwords are stored in the clear

`sha`
Passwords are stored using SHA

`ssha`
Passwords are stored using SSHA

This property is used to configure the following features: `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

db-pwd-file – Directory Proxy Server configuration property

Description

Syntax	`password`
Default value	No default is defined.
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the file from which to read the database user password of the JDBC data source. The temporary file is read once, and the password is stored for future use.

This property takes a path to a file that contains the password value.

This property is used to configure the following features: `jdbc-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

db-url – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the database URL of the JDBC data source.

This property is used to configure the following features: `jdbc-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

db-user – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the database user of the JDBC data source.

The default behavior for this property is as follows: No user.

This property is used to configure the following features: `jdbc-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

default-log-level – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	info
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property sets the default log level for all categories of log message.

This property can take the following values in addition to the default.

`error`
Error logging

`warning`
Warning logging

`info`
Informational logging

`all`
All logging levels

`none`
All logging disabled

This property is used to configure the following features: `access-log`, `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

delete-weight – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	disabled
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the proportion of delete requests that are sent to the attached data source.

This property takes an integer.

The value of this property must be at least 1.

This property can also take the following values:

`disabled`
  Do not forward any delete requests to the data source

This property is used to configure the following features: `attached-ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

description – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	Default behavior is not defined.
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property enables you to attach a description to the feature.

This property is used to configure the following features: `connection-handler`, `jdbc-data-source`, `jdbc-data-source-pool`, `jdbc-data-view`, `join-data-view`, `ldap-data-source`, `ldap-data-source-pool`, `ldap-data-view`, `ldif-data-view`, `request-filtering-policy`, `resource-limits-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

distribution-algorithm – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the built-in algorithm used to distribute operations to data views that have the same base DN.

This property can take the following values in addition to the default.

`pattern-matching`
Requests are distributed to data views based on the match between the parameters of the requests and one or more patterns.

`lexicographic`
Requests are distributed to data views based on the lexicographic value of the RDN specified in the request. Lexicographic bounds are taken from the value of the first RDN beneath the base DN of the data view.

`numeric`
Requests are distributed to data views based on the numeric value of the RDN specified in the request. The numeric value is taken from the value of the first RDN beneath the base DN of the data view.

`replication`
Requests are distributed to data views based on the role of the data view in replication. The algorithm forces all write operations to be sent to all data sources in the data source pool, and all read operations to be sent to a single data source.

The default behavior for this property is as follows: No distribution algorithm is enabled

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also `dpconf`(1M)

Name

dn-join-rule – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property determines how the DN of entries in the secondary data view are constructed.

To be taken in account by the server, this property must be set on join data view if the `join-rule-control-enabled` property for join data view is set to `true`; otherwise it must be set on secondary data views. Only one DN join rule can be defined.

The default behavior for this property is as follows: No DN join rule is enabled

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

dn-mapping-attrs – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property defines a list of attributes that contain DNs of entries.

When a DN is renamed by setting the `dn-mapping-source-base-dn` property, attributes in the portion of the DIT affected by renaming must also be renamed if those attributes contain DNs.

The default behavior for this property is as follows: No DN valued attributes to be mapped

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

dn-mapping-source-base-dn – Directory Proxy Server configuration property

Description

Syntax	`dn`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property defines a DN mapping.

When a client makes a request, the DN is rewritten to match that on the server side. When the result is returned to the client, the DN is changed back to match the client side.

This property takes a Distinguished Name (DN) value.

The default behavior for this property is as follows: No source suffix - do not perform DN mapping

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

dn-pattern – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies a DN pattern that controls how DNs are constructed in the data view.

This property is used to configure the following features: `jdbc-object-class`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

domain-name-filters – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	any
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies a set of domain name suffixes. A client's network domain must match at least one of the suffixes in order for the connection to be accepted by the connection handler.

The domain name can be in one of the following formats:

- Full name. For example, box.eng.sun.com.
- Suffix name. For example, .eng.sun.com.
- Fully qualified name of the local host.

The default behavior for this property is as follows: All domains are accepted

This property is used to configure the following features: `connection-handler`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

driver-class – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the JDBC driver class of the JDBC data source.

This property is used to configure the following features: `jdbc-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

driver-url – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	Yes
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies the JDBC driver jar of the JDBC data source.

The default behavior for this property is as follows: The proxy will not use any password to connect to the JDBC data source.

This property is used to configure the following features: `jdbc-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

email-alerts-enabled – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	false
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not Directory Proxy Server should use email based alert notification.

This property is `true` or `false`.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

email-alerts-message-from-address – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	local
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the from-address that alert messages should use.

The syntax of this string is `EMAIL_ADDRESS`.

A valid email address

The value of this property must match the pattern `^.+@.+$`.

The default behavior for this property is as follows: The default sender is dps@localhost

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

email-alerts-message-subject – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	Proxy Server Administrative Alert
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the subject line that alert messages should use.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name email-alerts-message-subject-includes-alert-code – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	false
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not Directory Proxy Server should add the alert code to the subject line for alert messages.

This property is `true` or `false`.

This property is used to configure the following features: `server`.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also `dpconf`(1M)

Name

email-alerts-message-to-address – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	root@localhost
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies the to-address that alert messages should use.

The syntax of this string is `EMAIL_ADDRESS`.

A valid email address

The value of this property must match the pattern `^.+@.+$`.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

email-alerts-smtp-host – Directory Proxy Server configuration property

Description

Syntax	`ipAddress`
Default value	localhost
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the host name of the SMTP server to which alert messages should be sent.

This property takes an IP address or host name.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name email-alerts-smtp-port – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	smtp
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the port number of the SMTP server to which alert messages should be sent.

This property takes an integer.

The value of this property must be at least 1.

The value of this property must be no greater than 65535.

This property can also take the following values:

`smtp`
Standard smtp port

`smtps`
Standard smtp over SSL port

This property is used to configure the following features: `server`.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also `dpconf`(1M)

Name

enable-client-affinity – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	false
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag that indicates whether or not consecutive requests from the same client should be directed to the same LDAP data source.

This property is `true` or `false`.

This property is used to configure the following features: `ldap-data-source-pool`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name enabled-admin-alerts – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies the list of enabled administration alerts.

This property can take the following values in addition to the default.

`info-server-startup`
Server startup

`info-server-shutdown-clean`
Clean server shutdown

`error-server-shutdown-abrupt`
Abrupt server shutdown

`info-configuration-reload`
Configuration reloaded

`warning-configuration-reload-failure-no-impact`
Configuration reload failure due to bad configuration - run-time configuration not impacted

`error-configuration-reload-failure-with-impact`
Configuration reload failure due to bad configuration - run-time configuration possibly impacted

`warning-data-source-unavailable`
Data source is currently unavailable

`info-data-source-available`
Data source is available again

`warning-listener-unavailable`
Unable to listen for incoming connections or requests

`warning-data-sources-inconsistent`
Inconsistency detected between data sources

The default behavior for this property is as follows: No administration alerts are enabled

This property is used to configure the following features: `server`.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also `dpconf`(1M)

Name

enable-data-view-affinity – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	false
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not consecutive requests from the same client should be directed exclusively to the same data view.

This property is `true` or `false`.

This property is used to configure the following features: `connection-handler`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

enabled-ssl-cipher-suites – Directory Proxy Server configuration property

Description

Syntax	`selectionEnumeration`
Default value	JRE
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies the list of enabled SSL cipher suites.

This property takes its possible values from an external component.

The default behavior for this property is as follows: All SSL cipher suites enabled by the Java Run Time running the proxy.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

enabled-ssl-protocols – Directory Proxy Server configuration property

Description

Syntax	`selectionEnumeration`
Default value	JRE
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies the list of enabled SSL protocols.

This property takes its possible values from an external component.

The default behavior for this property is as follows: All SSL protocols enabled by the Java Run Time running the proxy.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

enable-log-rotation – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	true
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies whether log files are rotated or not.

This property is `true` or `false`.

This property is used to configure the following features: `access-log`, `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

enable-remote-user-mapping – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	false
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not Directory Proxy Server should map the bind DN of a user to an alternate bind DN. The identity mapping is configured in the user entry in a remote LDAP server.

This property is `true` or `false`.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

enable-user-mapping – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	false
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not Directory Proxy Server should map the user identity to the identity of an alternate user. The identity mapping is configured in the Directory Proxy Server.

This property is `true` or `false`.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name encrypt-configuration – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	true
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not Directory Proxy Server should encrypt passwords that are stored in the configuration.

This property is `true` or `false`.

This property is used to configure the following features: `server`.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also `dpconf`(1M)

Name

excluded-subtrees – Directory Proxy Server configuration property

Description

Syntax	`dn`
Default value	Default behavior is not defined.
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies the DNs of subtrees which are excluded by a data view.

When a subordinate data view is created, Directory Proxy Server automatically excludes the subordinate subtree from the superior data view. However, the automatic configuration can be disabled and the feature can be configured manually.

This property takes a Distinguished Name (DN) value.

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

extension-jar-file-url – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies the jar file that contains custom plugins, such as custom distribution plugins.

The default behavior for this property is as follows: Proxy uses no extension jar file.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

filter-join-rule – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property defines the relationship between the primary and secondary data views.

A filter join rule indicates how an entry should be retrieved from the secondary data view based on something in the primary data view.

For example, `uid=\${primary-view-name.uid}` is a valid property value.

To be taken in account by the server, this property must be set on join data view if the `join-rule-control-enabled` property for join data view is set to `true`; otherwise it must be set on secondary data views.

A filter join rule takes the form of an LDAP filter that is used to construct an attribute from one or more attributes from the primary data view.

The default behavior for this property is as follows: No filter join rule is applied

This property is used to configure the following features: `jdbc-data-view`, `jdbc-table`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

internal-value – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property defines the physical value of the attribute.

The default behavior for this property is as follows: For some transformations, this property is required for the proxy to apply the transformation.

This property is used to configure the following features: `virtual-transformation`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

ip-address-filters – Directory Proxy Server configuration property

Description

Syntax	`ipAddressMask`
Default value	any
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies a set of IPv4 or IPv6 address masks. The IP address of a client connection must match at least one of the masks in order for the connection to be accepted by the connection handler. The IP address can be in one of the following formats:

- IP address in dotted decimal form.
- IP address and bits, in the form of network number/mask bits.
- IP address and quad, in the form of a pair of dotted-decimal quads.
- All addresses, a catch-all for clients that are not placed into other, higher priority, groups.
- 0.0.0.0. This address is for groups to which initial membership is not considered. For example, for groups that clients switch to after their initial bind.
- IP address of the local host.

This property takes an IP address such as `168.192.0.*`.

The default behavior for this property is as follows: All IP addresses are accepted

This property is used to configure the following features: `connection-handler`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

is-enabled – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	true
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not the data view is accepting requests.

This property is `true` or `false`.

This property is used to configure the following features: `connection-handler`, `jdbc-data-source`, `jdbc-data-view`, `join-data-view`, `ldap-data-source`, `ldap-data-view`, `ldap-listener`, `ldaps-listener`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

is-read-only – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	false
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not the data view should accept read operations only.

This property is `true` or `false`.

This property is used to configure the following features: `jdbc-data-source`, `jdbc-data-view`, `join-data-view`, `ldap-data-source`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name is-restart-required – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	false
Must be set	No
Is modifiable	No
Is multivalued	No
Requires restart	No

This property is a flag indicating whether Directory Proxy Server must be restarted in order for configuration changes to take effect.

This property is `true` or `false`.

This property is used to configure the following features: `server`.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also `dpconf`(1M)

Name

is-routable – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	true
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not the data view can be accessed by a connection handler if the data-view-routing-policy(5dpconf) property of the connection handler is `all-routable`.

This property is `true` or `false`.

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

is-single-row-table – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	true
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies that an LDAP entry has only one matching row in the relational database table.

This property is `true` or `false`.

This property is used to configure the following features: `jdbc-table`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

is-ssl-mandatory – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	false
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not client connections must use SSL in order for them to be accepted by the connection handler.

This property is `true` or `false`.

This property is used to configure the following features: `connection-handler`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

jdbc-data-source-pool – Directory Proxy Server configuration property

Description

Syntax	`dnReference`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the JDBC data source pool that should be used by the JDBC data view.

This property has as its value the name of a `jdbc-data-source-pool` configuration entity.

This property is used to configure the following features: `jdbc-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

join-rule-control-enabled – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	false
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies whether Server uses "filter-join-rule" and "dn-join-rule" property values stored on join views.

This property is `true` or `false`.

This property is used to configure the following features: `join-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

ldap-address – Directory Proxy Server configuration property

Description

Syntax	`ipAddress`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the address of the LDAP data source.

This property takes an IP address or host name.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

ldap-data-source-pool – Directory Proxy Server configuration property

Description

Syntax	`dnReference`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the name of the LDAP data source pool to be used by the LDAP data view.

This property has as its value the name of a `ldap-data-source-pool` configuration entity.

This property is used to configure the following features: `ldap-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

ldap-port – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	ldap
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the LDAP port of the LDAP data source.

This property takes an integer.

The value of this property must be at least 1.

The value of this property must be no greater than 65535.

This property can also take the following values:

`ldap`
Standard ldap port

`ldaps`
Standard ldaps port

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

ldaps-port – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	ldaps
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the LDAPS port of the LDAP data source.

This property takes an integer.

The value of this property must be at least 1.

The value of this property must be no greater than 65535.

This property can also take the following values:

`ldap`
Standard ldap port

`ldaps`
Standard ldaps port

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

ldap-syntax – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	cis
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property defines the syntax used to construct the LDAP attribute from an entry in the relational database table.

This property is used to configure the following features: `jdbc-attr`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

ldif-data-source – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the name of a file on the Directory Proxy Server filesystem where the LDIF data is contained.

This property is used to configure the following features: `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

lexicographic-attrs – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	all
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies the set of attributes that are examined by the distribution algorithm when distribution-algorithm(5dpconf) is `lexicographic`.

The default behavior for this property is as follows: All attributes

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

lexicographic-lower-bound – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the lower bound of the distribution when distribution-algorithm(5dpconf) is `lexicographic`.

For example, consider a configuration with a first data view that handles [A-M] inclusive, and a second data view that handles [N-Z] inclusive. For the first data view, you set the lower bound to A, and the upper bound to M. For the second data view, you set the lower bound to N, and the upper bound to Z.

The default behavior for this property is as follows: No lower limit

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

lexicographic-upper-bound – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the upper bound of the distribution when distribution-algorithm(5dpconf) is `lexicographic`.

For example, consider a configuration with a first data view that handles [A-M] inclusive, and a second data view that handles [N-Z] inclusive. For the first data view, you set the lower bound to A, and the upper bound to M. For the second data view, you set the lower bound to N, and the upper bound to Z.

The default behavior for this property is as follows: No upper limit

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

listen-address – Directory Proxy Server configuration property

Description

Syntax	`ipAddress`
Default value	0.0.0.0
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the IP address that Directory Proxy Server should listen on.

This property takes an IP address or host name.

This property is used to configure the following features: `ldap-listener`, `ldaps-listener`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

listen-port – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the IP port that Directory Proxy Server should listen on.

This property takes an integer.

The value of this property must be at least 1.

The value of this property must be no greater than 65535.

This property can also take the following values:

`ldap`
Standard ldap port

`ldaps`
Standard ldaps port

This property is used to configure the following features: `ldap-listener`, `ldaps-listener`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

load-balancing-algorithm – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	proportional
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the algorithm that determines how operations are distributed to the data sources attached to a data source pool.

This property can take the following values in addition to the default.

`failover`
Requests are distributed exclusively to the data source with the highest weight. If that data source fails, requests are distributed exclusively to the data source with the next highest weight.

`saturation`
Requests are distributed to the data source with the highest weight until the data source approaches its saturation level. Requests are then sent to the data source with the next highest weight .

When the data source with the highest weight drops below its saturation level, Directory Proxy Server resumes sending requests to that data source.

`proportional`
Requests are distributed to data sources in proportion to the weight of a data source and its cumulative load.

`operational-affinity`
Requests are allocated a hash value according to the type of the request and the properties of the request. Hash values are allocated to data sources in proportion to the weight of a data source and its cumulative load.

This property is used to configure the following features: `ldap-data-source-pool`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-buffer-size – Directory Proxy Server configuration property

Description

Syntax	`dataSize`
Default value	0b
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the size of the error log buffer. When the buffer is full, it is flushed to disk.

This property is expressed in bytes.

The value of this property must be at least 0b.

The value of this property must be no greater than 100k.

This property is used to configure the following features: `access-log`, `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-file-name – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the path name to the log file.

This property is used to configure the following features: `access-log`, `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-file-perm – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	600
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the permissions on the log file.

The syntax of this string is `OCTAL_MODE`.

A unix style octal permission, for example, 600.

The value of this property must match the pattern `^[0-7][0-7][0-7]$`.

This property is used to configure the following features: `access-log`, `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name log-level-client-connections – Directory Proxy Server configuration property

Description

Syntax	`accessLogLevel`
Default value	inherited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the log level for events related to client connections.

This property can take the following values.

`all` All messages are included in the log file. In most cases, this setting produces the same results as the info setting. In certain situations, this setting enables additional debugging messages to be logged.

`info` Errors, warnings and informational messages are included in the log file.

`inherited` The log level is inherited from the value of the `default-log-level` property.

`none` No messages are included in the log file.

This property is used to configure the following features: `access-log`.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also `dpconf`(1M)

Name

log-level-client-disconnections – Directory Proxy Server configuration property

Description

Syntax	`accessLogLevel`
Default value	inherited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the log level for events related to client disconnections.

This property can take the following values.

`all` All messages are included in the log file. In most cases, this setting produces the same results as the info setting. In certain situations, this setting enables additional debugging messages to be logged.

`info` Errors, warnings and informational messages are included in the log file.

`inherited` The log level is inherited from the value of the `default-log-level` property.

`none` No messages are included in the log file.

This property is used to configure the following features: `access-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-level-client-operations – Directory Proxy Server configuration property

Description

Syntax	`accessLogLevel`
Default value	inherited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the log level for events related to client operations.

This property can take the following values.

`all` All messages are included in the log file. In most cases, this setting produces the same results as the info setting. In certain situations, this setting enables additional debugging messages to be logged.

`info` Errors, warnings and informational messages are included in the log file.

`inherited` The log level is inherited from the value of the `default-log-level` property.

`none` No messages are included in the log file.

This property is used to configure the following features: `access-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-level-configuration – Directory Proxy Server configuration property

Description

Syntax	`logLevel`
Default value	inherited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the logging level for events related to configuration.

This property can take the following values.

`all`	All messages are included in the log file. In most cases, this setting produces the same results as the info setting. In certain situations, this setting enables additional debugging messages to be logged.
`error`	Only error messages are included in the log file.
`info`	Errors, warnings and informational messages are included in the log file.
`inherited`	The log level is inherited from the value of the `default-log-level` property.
`none`	No messages are included in the log file.
`warning`	Error messages and warning messages are included in the log file.

This property is used to configure the following features: `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-level-connection-handlers – Directory Proxy Server configuration property

Description

Syntax	`accessLogLevel`
Default value	inherited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the log level for events related to connection handlers.

This property can take the following values.

`all` All messages are included in the log file. In most cases, this setting produces the same results as the info setting. In certain situations, this setting enables additional debugging messages to be logged.

`info` Errors, warnings and informational messages are included in the log file.

`inherited` The log level is inherited from the value of the `default-log-level` property.

`none` No messages are included in the log file.

This property is used to configure the following features: `access-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-level-data-source – Directory Proxy Server configuration property

Description

Syntax	`logLevel`
Default value	inherited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the logging level for events related to data sources.

This property can take the following values.

`all`	All messages are included in the log file. In most cases, this setting produces the same results as the info setting. In certain situations, this setting enables additional debugging messages to be logged.
`error`	Only error messages are included in the log file.
`info`	Errors, warnings and informational messages are included in the log file.
`inherited`	The log level is inherited from the value of the `default-log-level` property.
`none`	No messages are included in the log file.
`warning`	Error messages and warning messages are included in the log file.

This property is used to configure the following features: `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-level-data-sources – Directory Proxy Server configuration property

Description

Syntax	`accessLogLevel`
Default value	inherited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the log level for events related to data sources.

This property can take the following values.

`all` All messages are included in the log file. In most cases, this setting produces the same results as the info setting. In certain situations, this setting enables additional debugging messages to be logged.

`info` Errors, warnings and informational messages are included in the log file.

`inherited` The log level is inherited from the value of the `default-log-level` property.

`none` No messages are included in the log file.

This property is used to configure the following features: `access-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-level-data-sources-detailed – Directory Proxy Server configuration property

Description

Syntax	`accessLogLevel`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the log level for detailed events related to data sources.

This property can take the following values.

`all` All messages are included in the log file. In most cases, this setting produces the same results as the info setting. In certain situations, this setting enables additional debugging messages to be logged.

`info` Errors, warnings and informational messages are included in the log file.

`inherited` The log level is inherited from the value of the `default-log-level` property.

`none` No messages are included in the log file.

This property is used to configure the following features: `access-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-level-internal – Directory Proxy Server configuration property

Description

Syntax	`logLevel`
Default value	inherited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

The logging level for events related to problems with the internal server.

This property can take the following values.

`all` All messages are included in the log file. In most cases, this setting produces the same results as the info setting. In certain situations, this setting enables additional debugging messages to be logged.

`error` Only error messages are included in the log file.

`info` Errors, warnings and informational messages are included in the log file.

`inherited` The log level is inherited from the value of the `default-log-level` property.

`none` No messages are included in the log file.

`warning` Error messages and warning messages are included in the log file.

This property is used to configure the following features: `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-level-operation-decode – Directory Proxy Server configuration property

Description

Syntax	`logLevel`
Default value	inherited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the logging level for events related to client operation decoding.

This property can take the following values.

`all` All messages are included in the log file. In most cases, this setting produces the same results as the info setting. In certain situations, this setting enables additional debugging messages to be logged.

`error` Only error messages are included in the log file.

`info` Errors, warnings and informational messages are included in the log file.

`inherited` The log level is inherited from the value of the `default-log-level` property.

`none` No messages are included in the log file.

`warning` Error messages and warning messages are included in the log file.

This property is used to configure the following features: `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-level-operation-processing – Directory Proxy Server configuration property

Description

Syntax	`logLevel`
Default value	inherited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the logging level for events related to client operation processing.

This property can take the following values.

`all` All messages are included in the log file. In most cases, this setting produces the same results as the info setting. In certain situations, this setting enables additional debugging messages to be logged.

`error` Only error messages are included in the log file.

`info` Errors, warnings and informational messages are included in the log file.

`inherited` The log level is inherited from the value of the `default-log-level` property.

`none` No messages are included in the log file.

`warning` Error messages and warning messages are included in the log file.

This property is used to configure the following features: `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-level-plugin – Directory Proxy Server configuration property

Description

Syntax	`logLevel`
Default value	inherited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the logging level for events related to plugins.

This property can take the following values.

`all`	All messages are included in the log file. In most cases, this setting produces the same results as the info setting. In certain situations, this setting enables additional debugging messages to be logged.
`error`	Only error messages are included in the log file.
`info`	Errors, warnings and informational messages are included in the log file.
`inherited`	The log level is inherited from the value of the `default-log-level` property.
`none`	No messages are included in the log file.
`warning`	Error messages and warning messages are included in the log file.

This property is used to configure the following features: `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-level-shutdown – Directory Proxy Server configuration property

Description

Syntax	`logLevel`
Default value	inherited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the logging level for events related to server shutdown.

This property can take the following values.

`all` All messages are included in the log file. In most cases, this setting produces the same results as the info setting. In certain situations, this setting enables additional debugging messages to be logged.

`error` Only error messages are included in the log file.

`info` Errors, warnings and informational messages are included in the log file.

`inherited` The log level is inherited from the value of the `default-log-level` property.

`none` No messages are included in the log file.

`warning` Error messages and warning messages are included in the log file.

This property is used to configure the following features: `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-level-startup – Directory Proxy Server configuration property

Description

Syntax	`logLevel`
Default value	inherited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the logging level for events related to server startup.

This property can take the following values.

`all` All messages are included in the log file. In most cases, this setting produces the same results as the info setting. In certain situations, this setting enables additional debugging messages to be logged.

`error` Only error messages are included in the log file.

`info` Errors, warnings and informational messages are included in the log file.

`inherited` The log level is inherited from the value of the `default-log-level` property.

`none` No messages are included in the log file.

`warning` Error messages and warning messages are included in the log file.

This property is used to configure the following features: `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-min-size – Directory Proxy Server configuration property

Description

Syntax	`dataSize`
Default value	0b
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

If the log-rotation-policy(5dpconf) is `periodic`, this property specifies a minimum file size. The log files are rotated at the specified interval if the file size is bigger than the specified size.

This property is expressed in bytes.

The value of this property must be at least 0b.

The value of this property must be no greater than 2g.

This property is used to configure the following features: `access-log`, `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-rotation-frequency – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	1h
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the interval at which log files are rotated when log-rotation-policy(5dpconf) is `periodic`.

This property is set in conjunction with the following properties: log-rotation-start-time(5dpconf) and log-rotation-start-day(5dpconf).

The syntax of this string is `<count>[mwdh]`.

The value for log-rotation-frequency(5dpconf) is a time period of the form <count>[mwdh]. For example, a value of `2w` means that the logs are rotated every 2 weeks.

The value of this property must match the pattern `^[0-9]+[mMwWdDhH]$`.

This property is used to configure the following features: `access-log`, `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-rotation-policy – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the policy used to rotate log files.

This property can take the following values in addition to the default.

`size`
Rotate log files when they reach the size specified by the log-rotation-size(5dpconf) property.

`periodic`
Rotate log files at the time and interval specified by the following properties:

- log-rotation-start-time(5dpconf)
- log-rotation-start-day(5dpconf)
- log-rotation-frequency(5dpconf)

If the rotation policy is `periodic` and log-rotation-size(5dpconf) is set, the log file is rotated at the specified interval IF the file size is bigger than the specified size.

This property is used to configure the following features: `access-log`, `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-rotation-size – Directory Proxy Server configuration property

Description

Syntax	`dataSize`
Default value	100m
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

If the log-rotation-policy(5dpconf) is `size`, this property specifies the file size at which log files are automatically rotated.

This property is expressed in bytes.

The value of this property must be at least 1m.

The value of this property must be no greater than 2g.

This property is used to configure the following features: `access-log`, `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-rotation-start-day – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	1
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the day-of-the-week or day-of-the-month that log files are rotated when log-rotation-policy(5dpconf) is `periodic`.

This property is set in conjunction with the following properties: log-rotation-start-time(5dpconf) and log-rotation-frequency(5dpconf).

This property takes an integer.

The value of this property must be at least 1.

The value of this property must be no greater than 31.

This property is used to configure the following features: `access-log`, `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-rotation-start-time – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	0000
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the time of day at which log files are rotated when the log-rotation-policy(5dpconf) is `periodic`.

This property is set in conjunction with the following properties: log-rotation-start-day(5dpconf) and log-rotation-frequency(5dpconf).

The syntax of this string is `TIME`.

The value for log-rotation-start-time is a time of day of the form hhmm.

The value of this property must match the pattern `^((2[0-3])|([0-1][0-9]))[0-5][0-9]$`.

This property is used to configure the following features: `access-log`, `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

log-search-filters – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	true
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies whether or not search filters are included in log messages.

This property is `true` or `false`.

This property is used to configure the following features: `access-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

mapped-bind-dn – Directory Proxy Server configuration property

Description

Syntax	`dn`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the distinguished name of the user that the client is mapped to if enable-user-mapping(5dpconf) is `true`.

The distinguished name of the client is specified by user-bind-dn(5dpconf).

This property takes a Distinguished Name (DN) value.

The default behavior for this property is as follows: The proxy maps to the identity defined as the anonymous mapping.

This property is used to configure the following features: `user-mapping`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

mapped-bind-pwd – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	No
Is multivalued	No
Requires restart	No

This property specifies the password of the user that the client is mapped to if enable-user-mapping(5dpconf) is `true`.

The password of the client is specified by user-bind-pwd(5dpconf).

This property is read-only. To change the password, use the `mapped-bind-pwd-file` property.

The default behavior for this property is as follows: The proxy will not use any password associated to this mapping

This property is used to configure the following features: `user-mapping`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

mapped-bind-pwd-file – Directory Proxy Server configuration property

Description

Syntax	password
Default value	No default is defined.
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the file from which to read the password of the user that the client is mapped to if enable-user-mapping(5dpconf) is true.

The password of the client is specified by user-bind-pwd(5dpconf). The temporary file is read once, and the password is stored for future use.

This property takes a path to a file that contains the password value.

This property is used to configure the following features: user-mapping.

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

dpconf(1M)

Name

max-age – Directory Proxy Server configuration property

Description

Syntax	`duration`
Default value	unlimited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum age (in months, weeks or days) that a log file can reach before it is deleted.

The duration is expressed in seconds.

The value of this property must be at least 1d.

This property is used to configure the following features: `access-log`, `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

max-client-connections – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	unlimited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum number of simultaneous connections from a single client permitted by associated connection handlers.

This property takes an integer.

The value of this property must be at least 1.

The default behavior for this property is as follows: No limit

This property is used to configure the following features: `resource-limits-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name max-connection-queue-size – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	128
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum size of the listener's new connection queue. When the queue is full, new connections are rejected.

This property takes an integer.

The value of this property must be at least 1.

This property is used to configure the following features: `ldap-listener`, `ldaps-listener`.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also `dpconf`(1M)

Name

max-connections – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	unlimited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum number of simultaneous connections permitted by associated connection handlers.

This property takes an integer.

The value of this property must be at least 1.

The default behavior for this property is as follows: No limit

This property is used to configure the following features: `resource-limits-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

max-ldap-message-size – Directory Proxy Server configuration property

Description

Syntax	`dataSize`
Default value	unlimited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum size of an LDAP message. Messages above the maximum size are not accepted by the listener.

This property is expressed in bytes.

The value of this property must be at least 4k.

The value of this property must be no greater than 2g.

This property is used to configure the following features: `ldap-listener`, `ldaps-listener`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

max-log-files – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	10
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum number of log files that are preserved.

This property takes an integer.

The value of this property must be at least 0.

This property is used to configure the following features: `access-log`, `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

max-simultaneous-operations-per-connection – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	unlimited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum number of simultaneous operations per connection that is allowed by the associated connection handlers.

This property takes an integer.

The value of this property must be at least 1.

The default behavior for this property is as follows: No limit

This property is used to configure the following features: `resource-limits-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

max-size – Directory Proxy Server configuration property

Description

Syntax	`dataSize`
Default value	unlimited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum log size in bytes, kilobytes, Megabytes or Gigabytes.

This property is expressed in bytes.

The value of this property must be at least 1M.

This property is used to configure the following features: `access-log`, `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

max-total-operations-per-connection – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	unlimited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum number of operations per connection that is allowed by the associated connection handlers.

This property takes an integer.

The value of this property must be at least 1.

The default behavior for this property is as follows: No limit

This property is used to configure the following features: `resource-limits-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

min-free-disk-space-size – Directory Proxy Server configuration property

Description

Syntax	`dataSize`
Default value	1M
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the minimum allowed free disk space for logs in bytes, kilobytes, Megabytes, or Gigabytes.

This property is expressed in bytes.

The value of this property must be at least 1M.

The value of this property cannot be unlimited.

This property is used to configure the following features: `access-log`, `error-log`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

minimum-search-filter-substring-length – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	unlimited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the minimum length of a substring in a search filter.

This property takes an integer.

The value of this property must be at least 1.

This property can also take the following values:

`unlimited`
No limit

`prohibited`
Substring filters prohibited

This property is used to configure the following features: `resource-limits-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

model – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property defines the transformation model. The transformation model is determined by the direction of a transformation, in other words, whether the transformation is applied during the request, during the response, or both.

This property can take the following values in addition to the default.

`mapping`
  Mapping

`write`
  Store and forget

`read`
  Default virtual value

This property is used to configure the following features: `virtual-transformation`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

modify-dn-weight – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	disabled
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the proportion of modify DN requests that are sent to the attached data source.

This property takes an integer.

The value of this property must be at least 1.

This property can also take the following values:

`disabled`
Do not forward any modify DN requests to the data source

This property is used to configure the following features: `attached-ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

modify-weight – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	disabled
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the proportion of modify requests that are sent to the attached data source.

This property takes an integer.

The value of this property must be at least 1.

This property can also take the following values:

`disabled`
Do not forward any modify requests to the data source

This property is used to configure the following features: `attached-ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

monitoring-bind-timeout – Directory Proxy Server configuration property

Description

Syntax	`duration`
Default value	5000
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum length of time that the availability monitor waits to establish a connection to the LDAP data source.

The duration is expressed in milliseconds.

The value of this property must be at least 1.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

monitoring-entry-dn – Directory Proxy Server configuration property

Description

Syntax	`dn`
Default value	
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the DN of a target entry in a search operation. The availability monitor uses the search operation to test a connection to the data source.

This property takes a Distinguished Name (DN) value.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

monitoring-entry-timeout – Directory Proxy Server configuration property

Description

Syntax	`duration`
Default value	5000
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum length of time that the availability monitor tries to retrieve the target entry in a search operation. The availability monitor uses the search operation to test a connection to the data source.

The duration is expressed in milliseconds.

The value of this property must be at least 1.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

monitoring-inactivity-timeout – Directory Proxy Server configuration property

Description

Syntax	`duration`
Default value	120
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

The availability monitor polls inactive connections to keep them alive. This property specifies how long a connection can be inactive before the availability monitor establishes a dedicated connection to detect the data source.

The duration is expressed in seconds.

The value of this property must be at least 1.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

monitoring-interval – Directory Proxy Server configuration property

Description

Syntax	`duration`
Default value	30
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the polling interval. If a connection is found to be down, the availability monitor polls the connection at this interval to detect its recovery.

The duration is expressed in seconds.

The value of this property must be at least 1.

The value of this property cannot be unlimited.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

monitoring-mode – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	proactive
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the monitoring mode of a data source.

This property can take the following values in addition to the default.

`proactive`
The availability monitor checks the availability of the data source continuously.

`reactive`
The availability monitor checks the availability of the data source only after a client request times out, or when an I/O error has been detected.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

monitoring-search-filter – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	(|(objectClass=*)(objectClass=ldapSubEntry))
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the filter used in a search operation. The availability monitor uses the search operation to test a connection to the data source.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

non-viewable-attr – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property provides a list of attributes that are not exposed by the data view.

The default behavior for this property is as follows: No restriction is applied on the list of viewable attributes.

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

non-writable-attr – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property provides a list of attributes that cannot be written through the data view.

The default behavior for this property is as follows: No restriction is applied on the list of writable attributes.

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

number-of-search-threads – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	20
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the number of connections that should be made to a data source so that search operations can be performed in parallel.

This property takes an integer.

The value of this property must be at least 1.

This property can also take the following values:

`unlimited`
No limit

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

number-of-threads – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	2
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the number of threads allocated to the listener to handle simultaneous client connections and requests.

This property takes an integer.

The value of this property must be at least 1.

The value of this property must be no greater than 64.

This property is used to configure the following features: `ldap-listener`, `ldaps-listener`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

number-of-worker-threads – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	50
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the number of worker threads available for processing operations in the work queue.

This property takes an integer.

The value of this property must be at least 1.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

num-bind-incr – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	10
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the number of new connections that are created when the server needs more connections for bind operations.

This property takes an integer.

The value of this property must be at least 1.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

num-bind-init – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	10
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the initial number of connections that should be made to an LDAP data source to perform bind operations.

This property takes an integer.

The value of this property must be at least 0.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

num-bind-limit – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	1024
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum number of connections that can be made to an LDAP data source to perform bind operations.

This property takes an integer.

The value of this property must be at least 1.

This property can also take the following values:

`unlimited`
: This value means no limit is set for this property.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

numeric-attrs – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	all
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies the set of attributes that are examined by the distribution algorithm when the distribution-algorithm(5dpconf) is `numeric`.

The default behavior for this property is as follows: All attributes

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

numeric-default-data-view – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	false
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not the associated data view should act as a default data view in the numeric distribution set and handle requests that contain non-numeric target RDNs.

This property is `true` or `false`.

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

numeric-lower-bound – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the lower bound of distribution when the distribution-algorithm(5dpconf) is `numeric`.

This property takes an integer.

The default behavior for this property is as follows: No lower limit

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

numeric-upper-bound – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the upper bound of distribution when the distribution-algorithm(5dpconf) is `numeric`.

This property takes an integer.

The default behavior for this property is as follows: No upper limit

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

num-read-incr – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	10
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the number of new connections that are created when the server needs more connections for read operations.

This property takes an integer.

The value of this property must be at least 1.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

num-read-init – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	10
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the initial number of connections that should be made to an LDAP data source to perform read operations.

This property takes an integer.

The value of this property must be at least 0.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

num-read-limit – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	1024
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum number of connections that can be made to an LDAP data source to perform read operations.

This property takes an integer.

The value of this property must be at least 1.

This property can also take the following values:

`unlimited`
This value means no limit is set for this property.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

num-write-incr – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	10
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the number of new connections that are created when the server needs more connections for write operations.

This property takes an integer.

The value of this property must be at least 1.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

num-write-init – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	10
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies the initial number of connections that should be made to an LDAP data source to perform write operations.

This property takes an integer.

The value of this property must be at least 0.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

num-write-limit – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	1024
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum number of connections that can be made to an LDAP data source to perform write operations.

This property takes an integer.

The value of this property must be at least 1.

This property can also take the following values:

`unlimited`
This value means no limit is set for this property.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name one-level-search-base-dn – Directory Proxy Server configuration property

Description

Syntax	`dn`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies the list of one-level search bases to which the search-size-limit property applies. Custom search limits are defined for a specific resource limits policy.

This property takes a Distinguished Name (DN) value.

The default behavior for this property is as follows: The search-size-limit property does not apply to any one-level search.

This property is used to configure the following features: `custom-search-size-limit`.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also `dpconf`(1M)

Name

pattern-matching-base-object-search-filter – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	all
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies a pattern for a search filter. The filter of a base-level search request must match the specified pattern for the request to be handled by the data view.

The default behavior for this property is as follows: Match all base-object search filters

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

pattern-matching-dn-regular-expression – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	all
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies a pattern for a regular expression. The target DN of a request must match this pattern in order for the data view to handle the request. The pattern is relative to the base DN of the data view.

The default behavior for this property is as follows: Match all operation target DNs

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

pattern-matching-one-level-search-filter – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	all
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies a pattern for a search filter. The filter of a one-level search request must match the specified pattern for the request to be handled by the data view.

The default behavior for this property is as follows: Match all one-level search filters

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

pattern-matching-subtree-search-filter – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	all
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies a pattern for a search filter. The filter of a subtree-level search request must match the specified pattern for the request to be handled by the data view.

The default behavior for this property is as follows: Match all subtree search filters

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

primary-table – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the primary JDBC table from which the object class obtains its list of entries.

This property is used to configure the following features: `jdbc-object-class`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

primary-view – Directory Proxy Server configuration property

Description

Syntax	`dnReference`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property defines the primary data view that forms the source of a join data view.

The value of this property is the name of one of the following configuration entities: jdbc-data-view, join-data-view, ldap-data-view, ldif-data-view.

This property is used to configure the following features: `join-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

priority – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	99
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the priority of the connection handler. A connection is evaluated against connection handlers in order of the priority of the connection handler, as follows:

- Priority 1 is the highest priority connection handler.
- Priority 100 is the lowest priority connection handler. Priority 100 is reserved for the default connection handler.

This property takes an integer.

The value of this property must be at least 1.

The value of this property must be no greater than 99.

This property is used to configure the following features: `connection-handler`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

process-bind – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	No default is defined.
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies whether binds are permitted on a data view.

This property is `true` or `false`.

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

prohibited-comparable-attrs – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies a set of LDAP attribute types that cannot be compared in an LDAP search filter or compare operation.

The default behavior for this property is as follows: None - all attribute types can be compared

This property is used to configure the following features: `request-filtering-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

prohibited-subtrees – Directory Proxy Server configuration property

Description

Syntax	`dn`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies a set of subtrees that cannot be accessed by clients.

This property takes a Distinguished Name (DN) value.

The default behavior for this property is as follows: No subtrees are prohibited

This property is used to configure the following features: `request-filtering-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name proxied-auth-check-timeout – Directory Proxy Server configuration property

Description

Syntax	`duration`
Default value	1800000
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies a timeout used during proxy authorization.

When a client operation contains a proxy authorization control, Directory Proxy Server checks that the `clientDN` has the right to impersonate the `clientPauthDN`.

If client-cred-mode(5dpconf) is set to `use-proxy-auth`, Directory Proxy Server checks that the `clientDN` has the relevant ACIs in the LDAP server by using the `getEffectiveRights` command.

The result is cached in the Directory Proxy Server and renewed when `proxied-auth-check-timeout` expires.

The duration is expressed in milliseconds.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

proxied-auth-use-v1 – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	false
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether Directory Proxy Server will use proxy authorization control version 1 or version 2, as follows:

- If the flag is `true`, Directory Proxy Server uses proxy authorization control v1.
- If the flag is `false`, Directory Proxy Server uses proxy authorization control v2.

This property is `true` or `false`.

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

referral-bind-policy – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	default
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the bind policy that is applied when following referrals.

This property can take the following values in addition to the default.

`user`
Use credentials if available

`anonymous`
Always anonymous

The default behavior for this property is as follows: Use the settings specified in the default connection handler.

This property is used to configure the following features: `resource-limits-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

referral-hop-limit – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	default
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum number of hops that are allowed when following referrals.

This property takes an integer.

The value of this property must be at least 1.

This property can also take the following values:

`unlimited`
  No limit

The default behavior for this property is as follows: Use the settings specified in the default connection handler.

This property is used to configure the following features: `resource-limits-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

referral-policy – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	default
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the policy that is applied when a referral is returned by an LDAP server.

This property can take the following values in addition to the default.

`follow`
Follow referrals

`forward`
Forward referrals to client

`discard`
Discard referrals

The default behavior for this property is as follows: Use the settings specified in the default connection handler

This property is used to configure the following features: `resource-limits-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

remote-user-mapping-bind-dn-attr – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the name of the attribute that contains an alternate bind DN. The attribute is contained in a user entry on a remote LDAP server. The attribute is used to perform remote user mapping when enable-remote-user-mapping(5dpconf) is `true`.

The default behavior for this property is as follows: This property is required when proxy performs remote user mapping.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

replication-role – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	master
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the role that the data view plays in replication distribution.

This property can take the following values in addition to the default.

`consumer`
The data view simulates a replication consumer and handles read operations only.

`master`
The data view simulates a replication master and handles read and write operations.

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

request-filtering-policy – Directory Proxy Server configuration property

Description

Syntax	`dnReference`
Default value	no-filtering
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the name of a request filtering policy which is to be used by the connection handler.

This property has as its value the name of a `request-filtering-policy` configuration entity.

The default behavior for this property is as follows: No request filtering policy - all requests are permitted.

This property is used to configure the following features: `connection-handler`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

resource-limits-policy – Directory Proxy Server configuration property

Description

Syntax	`dnReference`
Default value	no-limits
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the name of a resource limits policy which is to be used by the connection handler.

This property has as its value the name of a `resource-limits-policy` configuration entity.

The default behavior for this property is as follows: No resource limits policy - no resource limits apply.

This property is used to configure the following features: `connection-handler`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

rule-action – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	hide-entry
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies how the result of a search operation should be returned.

This property can take the following values in addition to the default.

`hide-entry`
Do not return target entries

`hide-attrs`
Return target entries, filtering out the attributes specified by the attrs property

`show-attrs`
Return target entries, filtering out the attributes not specified by the attrs property

This property is used to configure the following features: `search-data-hiding-rule`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

schema-check-enabled – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	false
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not the connection handler should perform a schema check.

This property is `true` or `false`.

This property is used to configure the following features: `connection-handler`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

scriptable-alerts-command – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	No default is defined.
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the command to use for handling alert messages.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

scriptable-alerts-enabled – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	false
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not Directory Proxy Server should use a customizable script for alert notification.

This property is `true` or `false`.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

search-mode – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	parallel
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies how searches that span multiple data sources are performed.

This property can take the following values in addition to the default.

`parallel`
Perform searches in parallel

`sequential`
Perform searches sequentially

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

search-size-limit – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	unlimited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum number of entries that can be returned by a search operation.

This property takes an integer.

The value of this property must be at least 1.

This property can also take the following values:

`unlimited`
This value means no limit is set for this property.

This property is used to configure the following features: `custom-search-size-limit`, `resource-limits-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

search-time-limit – Directory Proxy Server configuration property

Description

Syntax	`duration`
Default value	unlimited
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum duration of a search operation.

The duration is expressed in milliseconds.

The value of this property cannot be unlimited.

The default behavior for this property is as follows: No limit

This property is used to configure the following features: `resource-limits-policy`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name search-wait-timeout – Directory Proxy Server configuration property

Description

Syntax	`duration`
Default value	10000
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the maximum length of time that Directory Proxy Server waits for a search thread to become available.

The duration is expressed in milliseconds.

This property is used to configure the following features: `server`.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also `dpconf`(1M)

Name

search-weight – Directory Proxy Server configuration property

Description

Syntax	`integer`
Default value	disabled
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the proportion of search requests that are sent to the attached data source.

This property takes an integer.

The value of this property must be at least 1.

This property can also take the following values:

`disabled`
  Do not forward any search requests to the data source

This property is used to configure the following features: `attached-ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

secondary-table – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies an optional additional JDBC table from which the object class obtains additional information about its entries.

The default behavior for this property is as follows: No additional JDBC table is considered.

This property is used to configure the following features: `jdbc-object-class`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

secondary-view – Directory Proxy Server configuration property

Description

Syntax	`dnReference`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property defines the secondary data view that forms the source of a join data view.

The value of this property is the name of one of the following configuration entities: jdbc-data-view, join-data-view, ldap-data-view, ldif-data-view.

This property is used to configure the following features: `join-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

sql-column – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property provides the column in the relational database table from which the LDAP attribute is obtained.

This property is used to configure the following features: `jdbc-attr`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

sql-syntax – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	VARCHAR
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property defines the syntax used to construct an entry in the relational database table from an LDAP entry.

This property is used to configure the following features: `jdbc-attr`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name sql-table – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	Default behavior is not defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the name of the relational database table.

This property is used to configure the following features: `jdbc-table`.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also `dpconf`(1M)

Name

ssl-client-cert-alias – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the alias of the certificate used to negotiate SSL connections with data sources.

The default behavior for this property is as follows: Proxy applies chooses an alias based on the public key type and the list of certificate issuer authorities recognized by the peer (if any).

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

ssl-policy – Directory Proxy Server configuration property

Description

Syntax	`enumeration`
Default value	never
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property specifies whether SSL should be used for connections between Directory Proxy Server and a data source.

This property can take the following values in addition to the default.

`always`
Always use SSL

`client`
Use SSL if the client is using SSL

`never`
Never use SSL

This property is used to configure the following features: `ldap-data-source`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

ssl-server-cert-alias – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the alias of the certificate used to negotiate SSL connections with clients.

The default behavior for this property is as follows: Proxy applies chooses an alias based on the public key type and the list of certificate issuer authorities recognized by the peer (if any). After instance creation, the value of this property is defaultServerCert.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

subtree-search-base-dn – Directory Proxy Server configuration property

Description

Syntax	`dn`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies the list of subtree search bases to which the search-size-limit property applies. Custom search limits are defined for a specific resource limits policy.

This property takes a Distinguished Name (DN) value.

The default behavior for this property is as follows: The search-size-limit property does not apply to any subtree search.

This property is used to configure the following features: `custom-search-size-limit`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

super-class – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies a list of additional object classes to be returned as part of the objectclass attribute.

The default behavior for this property is as follows: No additional object class is returned.

This property is used to configure the following features: `jdbc-object-class`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

supported-ssl-cipher-suites – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	JRE
Must be set	No
Is modifiable	No
Is multivalued	Yes
Requires restart	No

This property specifies the list of SSL cipher suites that are supported by Directory Proxy Server.

The default behavior for this property is as follows: All SSL cipher suites supported by the Java Run Time running the proxy.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

supported-ssl-protocols – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	JRE
Must be set	No
Is modifiable	No
Is multivalued	Yes
Requires restart	No

This property specifies the list of SSL protocols that are supported by Directory Proxy Server.

The default behavior for this property is as follows: All SSL protocols supported by the Java Run Time running the proxy.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

syslog-alerts-enabled – Directory Proxy Server configuration property

Description

Syntax	boolean
Default value	false
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not Directory Proxy Server should use the system log for alert notification.

This property is true or false.

This property is used to configure the following features: server.

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

dpconf(1M)

Name

syslog-alerts-facility – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	USER
Must be set	No
Is modifiable	No
Is multivalued	No
Requires restart	No

This property specifies the syslog message category that alert messages should use.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

syslog-alerts-host – Directory Proxy Server configuration property

Description

Syntax	`ipAddress`
Default value	localhost
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the host name of the syslogd daemon that alert messages should be sent to.

This property takes an IP address or host name.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

target-attr-value-assertions – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies a list of attribute:value assertions in the form `attrName:attrValue`. The search data hiding rule applies to entries that match one or more of the specified assertions.

The syntax of this string is `<attr>#<value>`.

An attribute value assertion of the form <attribute>#<value>

The value of this property must match the pattern `^[a-zA-Z][-a-zA-Z0-9]+#.+$`.

The default behavior for this property is as follows: No assertion is defined.

This property is used to configure the following features: `search-data-hiding-rule`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

target-dn-regular-expressions – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies a list of name patterns. The search data hiding rule applies to entries whose name matches one or more of the specified patterns.

The default behavior for this property is as follows: No pattern is defined.

This property is used to configure the following features: `search-data-hiding-rule`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

target-dns – Directory Proxy Server configuration property

Description

Syntax	`dn`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property specifies a list of entry names. The search data hiding rule applies to all the listed entries.

This property takes a Distinguished Name (DN) value.

The default behavior for this property is as follows: No DN is defined.

This property is used to configure the following features: `search-data-hiding-rule`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

use-cert-subject-as-bind-dn – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	true
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not Directory Proxy Server should use a user certificate subject as the user DN.

This property is `true` or `false`.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

use-external-schema – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	false
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	Yes

This property is a flag indicating whether or not Directory Proxy Server should use an external LDAP schema.

This property is `true` or `false`.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

user-bind-dn – Directory Proxy Server configuration property

Description

Syntax	`dn`
Default value	none
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the distinguished name of the client that is mapped if enable-user-mapping(5dpconf) is `true`.

The distinguished name of the user that the client is mapped to is specified by mapped-bind-dn(5dpconf).

This property takes a Distinguished Name (DN) value.

The default behavior for this property is as follows: Proxy does no apply user mapping.

This property is used to configure the following features: `user-mapping`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

user-bind-pwd – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	No
Is multivalued	No
Requires restart	No

This property specifies the password of the client that is mapped if enable-user-mapping(5dpconf) is `true`.

The password of the user to which the client is mapped is specified by mapped-bind-pwd(5dpconf).

This property is read-only. To change the password, use the `user-bind-pwd-file` property.

The default behavior for this property is as follows: The proxy will not associate any password to the mapped identity.

This property is used to configure the following features: `user-mapping`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

user-bind-pwd-file – Directory Proxy Server configuration property

Description

Syntax	`password`
Default value	No default is defined.
Must be set	Yes
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the file from which to read the password of the client that is mapped if enable-user-mapping(5dpconf) is `true`.

The password of the user that the client is mapped to is specified by mapped-bind-pwd(5dpconf). The temporary file is read once, and the password stored for future use.

This property takes a path to a file that contains the password value.

This property is used to configure the following features: `user-mapping`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

user-filter – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	any
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies an LDAP search filter. The entry of the bound client must match the LDAP search filter in order for the connection to be accepted by the connection handler.

For example, the following filter could be used as a criteria for a connection handler: "`uid>=1000`".

Bound clients with a uid that matches the filter can be allocated to the connection handler.

The default behavior for this property is as follows: All users are accepted

This property is used to configure the following features: `connection-handler`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

user-mapping-anonymous-bind-dn – Directory Proxy Server configuration property

Description

Syntax	dn
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the bind DN to which anonymous users are mapped if enable-user-mapping(5dpconf) is `true`.

This property takes a Distinguished Name (DN) value.

The default behavior for this property is as follows: No mapping is applied to anonymous users.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

user-mapping-anonymous-bind-pwd – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	No
Is multivalued	No
Requires restart	No

This property specifies the authentication password for anonymous user mapping if enable-user-mapping(5dpconf) is `true`.

This property is read-only. To change the password, use the `user-mapping-anonymous-bind-pwd-file` property.

The default behavior for this property is as follows: No mapping is applied to anonymous users.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

user-mapping-anonymous-bind-pwd-file – Directory Proxy Server configuration property

Description

Syntax	`password`
Default value	No default is defined.
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the file from which to read the authentication password for anonymous user mapping if enable-user-mapping(5dpconf) is `true`. The temporary file is read once, and the password is stored for future use.

This property takes a path to a file that contains the password value.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

user-mapping-default-bind-dn – Directory Proxy Server configuration property

Description

Syntax	`dn`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies a default bind DN to which a user identity is mapped if enable-user-mapping(5dpconf) is `true` but the mapping fails.

User mapping can fail when a client identity is mapped to a non-existent alternative identity or when there has been a configuration error.

This property takes a Distinguished Name (DN) value.

The default behavior for this property is as follows: No mapping is applied.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

user-mapping-default-bind-pwd – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	No
Is multivalued	No
Requires restart	No

This property specifies the default bind password to use if enable-user-mapping(5dpconf) is `true` but the mapping fails.

User mapping can fail when a client identity is mapped to a non-existent alternative identity or when there has been a configuration error.

This property is read-only. To change the password, use the `user-mapping-default-bind-pwd-file` property.

The default behavior for this property is as follows: No mapping is applied to anonymous users.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

user-mapping-default-bind-pwd-file – Directory Proxy Server configuration property

Description

Syntax	password
Default value	No default is defined.
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property specifies the file from which to read the default bind password if enable-user-mapping(5dpconf) is true but the mapping fails. The temporary file is read once, and the password is stored for future use.

This property takes a path to a file that contains the password value.

This property is used to configure the following features: server.

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

dpconf(1M)

Name

use-tcp-no-delay – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	true
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not `TCP_NODELAY` should be enabled for connections between clients and a listener.

This property is `true` or `false`.

This property is used to configure the following features: `ldap-data-source`, `ldap-listener`, `ldaps-listener`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

verify-certs – Directory Proxy Server configuration property

Description

Syntax	`boolean`
Default value	false
Must be set	No
Is modifiable	Yes
Is multivalued	No
Requires restart	No

This property is a flag indicating whether or not Directory Proxy Server should verify that the client entry contains the SSL client certificate.

This property is `true` or `false`.

This property is used to configure the following features: `server`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

viewable-attr – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	all except non-viewable-attr
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property provides a list of attributes that are exposed by the data view.

The default behavior for this property is as follows: All attributes are viewable

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

view-value – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	none
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property defines the virtual value of the attribute.

The default behavior for this property is as follows: For some transformations, this property is required for the proxy to apply the transformation.

This property is used to configure the following features: `virtual-transformation`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

Name

writable-attr – Directory Proxy Server configuration property

Description

Syntax	`string`
Default value	all except non-writable-attr
Must be set	No
Is modifiable	Yes
Is multivalued	Yes
Requires restart	No

This property provides a list of attributes that can be written through the data view.

The default behavior for this property is as follows: All attributes are writable

This property is used to configure the following features: `jdbc-data-view`, `join-data-view`, `ldap-data-view`, `ldif-data-view`.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-proxy
Stability Level	Evolving

See Also

`dpconf`(1M)

REFERENCE

File Formats

Name certmap.conf – Map certificates to directory entries

Synopsis

```
Location: instance-path/alias/certmap.conf

# This is a comment.
certmap default default
[default:property1 [value1]]
[default:property2 [value2]]
[...]

[certmap name issuerDN
[name:property1 [value1]]
[name:property2 [value2]]
...
]
```

Description The `certmap.conf` file defines how Directory Server maps certificates to directory entries.

Comment lines are those starting with `#`.

CERTIFICATE MAPPINGS

The `certmap.conf` file consists of a series of certificate maps. It begins with a default certificate map, starting with the line:

```
certmap default default
```

Each subsequent certificate map starts with a line identifying the name of the map and the certificate authority issuer DN of the certificates to which the map applies.

The *issuerDN* string specified in the certificate map must correspond *exactly* to the issuer DN shown in the certificates. In particular, whitespace in the issuer DN is significant.

PROPERTIES AND VALUES

A certificate map also optionally specifies values for the following properties.

`DNComps` Specifies a comma separated list of relative distinguished name components of the base DN for an LDAP search to find the user entry matching the certificate. The components are taken from the subject DN of the certificate.

When the value of this property value is left empty, the base DN is the null suffix. In this particular case, searching against the null suffix in Directory Server searches every suffix in the directory. Thus leaving DNComps empty can have negative impact on performance.

The default behavior, when this property is commented out or not specified, is to take as the base DN the subject DN of the certificate.

`FilterComps` Specifies a comma separated list of LDAP attributes to form a filter for an LDAP search to find the user entry matching the certificate. The values for the filter are taken from the certificate, which can hold the following attributes.

`c`	Country
`cn`	Common name
`email`	Email address
`l`	Location
`o`	Organization
`ou`	Organizational unit
`st`	State
`uid`	UNIX user ID

For example, consider a certificate map named `example` containing the following `FilterComps` specification.

```
example:FilterComps e,uid
```

Then searches for the user entry matching the certificate use the filter `"(&amp;(mail=`*email-addr-from-cert*`)(`*uid-from-cert*`))"`.

The default behavior, when this property is commented out or not specified, is to use the filter `"(objectclass=*)"`.

`verifycert` Specifies whether the client application certificate is checked to make sure it is valid and not revoked.

This property can be usefully set to `on` if the directory stores client application certificates.

The default behavior is the same as `off`, meaning client certificates are not checked to be valid and not revoked.

`CmapLdapAttr` Specifies the name of the LDAP attribute in the directory containing the subject DN of the certificate.

The implied default value is `certSubjectDN`, not a standard LDAP attribute.

Note – If the LDAP attribute used is not of syntax DN, its value must match the subject DN provided *exactly* as the LDAP server does normalize DN values that are not stored in attributes with DN syntax.

`library` Specifies a shared plug-in library or DLL containing custom certificate mapping code.

There is no default.

`InitFn` Specifies the initialization function for the custom certificate mapping code in the library referenced by the value of the `library` property.

There is no default.

Examples

The following `certmap.conf` file specifies both a default certificate map, and an additional certificate map for certificates from the US subsidiary of Example.com.

```
# Example certmap.conf
certmap default default

certmap examplecerts ou=Example.com, o=examplecerts, c=US
examplecerts:DNComps ou,o,c
examplecerts:FilterComps e
examplecerts:verifycert on
```

When the server gets a certificate issued by any certificate authority other than the US subsidiary of Example.com, it uses the default mapping. If the certificate however has been issued by the US subsidiary of Example.com, the server looks for entries under the branch for the organizational unit and searches for entries using the client email address. It also verifies that such certificates are valid and that they are not revoked.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

See Also

dsadm(1M)

Name dse.ldif – Directory Server configuration file

Synopsis `Location:` *instance-path*`/config/dse.ldif`

`Location:` *instance-path*`/conf_bk/dse.ldif`

Description Directory Server stores its configuration as directory entries under `cn=config`. You can therefore change the server configuration by modifying configuration entries over LDAP, rather than by editing configuration files. Configuring Directory Server in this way allows you to reconfigure a remote server while it continues to serve other directory clients.

The `dse.ldif` file defines the configuration for a Directory Server instance. The `dse.ldif` file includes a set of entries under `cn=config`. These entries make up the modular parts of the Directory Server instance configuration.

Directory Server stores its schema under `cn=schema`, not as part of the rest of the server configuration. For an introduction to the schema available under `cn=schema`, see `Intro`(5DSSD).

Note – Neither the `dse.ldif` file nor the `cn=config` suffix constitute a public interface for configuring a Directory Server instance. Use dsconf(1M) instead.

The `dse.ldif` file has the following characteristics.

- The `dse.ldif` file is read only once at startup. Thereafter, the server configuration is based on the in-memory LDAP image of the configuration entries. Modifications to the `dse.ldif` file while the server is running are erased.
- Modification of the configuration with Directory Service Control Center or from the command line changes the LDAP image of the configuration. Some directory features read the current configuration when invoked and do not require the server to be restarted.
- Directory Server writes the `dse.ldif` file whenever the LDAP image of the configuration is changed. Some directory features read their configuration only when the server starts. Writing the file ensures the change is present.

  The existing `dse.ldif` file is copied to `dse.ldif.bak`, and the existing `dse.ldif.bak` is overwritten. Therefore, any manual changes to the `dse.ldif` file are lost if the configuration is changed through LDAP before the server is restarted.
- After every successful startup of the directory, the `dse.ldif` file is copied to `dse.ldif.startOK` in the same location. If your server cannot start because of a faulty configuration, restore the `dse.ldif` file from the `dse.ldif.startOK` file.
- The following restrictions apply to modifications to the server configuration.
  - Rather than delete configuration entries and add them again, you modify their attributes.
  - Some modifications only take effect after the server is restarted. See *ATTRIBUTES REQUIRING RESTART* in the manual page for details.
  - The `cn=monitor` entry cannot be modified.

- The server ignores invalid attribute values.

Extended Description

Directory Server has a modular configuration, with a number of distinct branches under the `cn=config` Directory Information Tree. The primary branches are below the following DNs.

`cn=encryption,cn=config`	Configuration attributes related to encryption
`cn=features,cn=config`	Access control for many server features, also configuration for internationalized matching and searching
`cn=mapping tree,cn=config`	Configuration for suffixes and replica
`cn=Password Policy,cn=config`	Default password policy configuration
`cn=plugins,cn=config`	Plug-in configuration entries for plug-in based server functionality, databases, indexes
`cn=replication,cn=config`	Default replication bind information for `cn=Replication Manager`, also formerly used for replication configuration
`cn=`*suffixName*`,cn=config`	Suffix configuration attributes
`cn=tasks,cn=config`	Used by the server to manage online import, backup, and so forth
`cn=uniqueid generator,cn=config`	Configuration attributes for providing unique IDs

About Configuration Attributes

The `dse.ldif` file contains all configuration information including directory specific entries created by Directory Server at startup, and directory specific entries related to the database, also created by Directory Server at startup. The file includes the Root DSE, named by `""`, and the entire contents of `cn=config`. When the server generates the `dse.ldif` file, it lists the entries in hierarchical order. It does so in the order that the entries appear in the directory under `cn=config`.

Within a configuration entry, each attribute is represented as an attribute name. The value of the attribute corresponds to the attribute's configuration.

The following example shows part of the `dse.ldif` file for a Directory Server instance. The example indicates, among other things, that schema checking has been turned *on*. This is represented by the attribute `nsslapd-schemacheck`, which takes the value `on`.

```
dn: cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsslapdConfig
nsslapd-accesslog-logging-enabled: on
nsslapd-enquote-sup-oc: on
nsslapd-localhost: myServer.example.com
```

```
nsslapd-errorlog: /local/ds/logs/errors
nsslapd-schemacheck: on
nsslapd-port: 389
nsslapd-localuser: nobody
...
```

See *CONFIGURATION ATTRIBUTES* in this manual page for a list of configuration attribute manual pages.

Access Control For Configuration Entries

When Directory Server is installed, a default set of Access Control Instructions, ACIs, is implemented for all entries under `cn=config`. The following extract from the `dse.ldif` file shows an example of these default ACIs.

```
aci: (targetattr != "aci") (targetscope = "base") (version 3.0;
 aci "Enable read access to rootdse for anonymous users";
 allow(read,search,compare) userdn="ldap:///anyone"; )
aci: (targetattr = "*") (version 3.0;  acl "Enable full access
 for Administrators group";  allow (all)(groupdn = "
 ldap:///cn=Administrators,cn=config"); )
aci: (targetattr = "userPassword") ( version 3.0;
 acl "allow userpassword self  modification";
 allow (write) userdn = "ldap:///self";)
```

By default, both the `cn=Directory Manager` user and the `cn=admin,cn=Administrators,cn=config` user have access to modify configuration entries. ACI syntax is covered elsewhere in the Directory Server Enterprise Edition documentation.

Configuration Attributes

This section lists configuration attributes by their location in the configuration Directory Information Tree.

Attributes of `cn=config`

General configuration entries are stored under the `cn=config` entry. The `cn=config` entry is an instance of the `nsslapdConfig` object class, which inherits from the `extensibleObject` object class. For attributes to be taken into account by the server, the entry must contain the `nsslapdConfig` object class, the `extensibleObject` object class and the `top` object class.

See the following manual pages.

- nsslapd-accesslog-level(5dsconf)
- nsslapd-allidsthreshold(5dsconf)
- nsslapd-attribute-name-exceptions(5dsconf)
- nsslapd-berbufsize(5dsconf)
- nsslapd-certmap-basedn(5dsconf)
- nsslapd-config(5dsconf)
- nsslapd-ds4-compatible-schema(5dsconf)
- nsslapd-enquote-sup-oc(5dsconf)
- nsslapd-groupevalnestlevel(5dsconf)
- nsslapd-idletimeout(5dsconf)

- nsslapd-instancedir(5dsconf)
- nsslapd-ioblocktimeout(5dsconf)
- nsslapd-lastmod(5dsconf)
- nsslapd-listenBacklog(5dsconf)
- nsslapd-listenhost(5dsconf)
- nsslapd-localhost(5dsconf)
- nsslapd-localuser(5dsconf)
- nsslapd-maxbersize(5dsconf)
- nsslapd-maxconnections(5dsconf)
- nsslapd-maxdescriptors(5dsconf)
- nsslapd-maxpsearch(5dsconf)
- nsslapd-maxthreadsperconn(5dsconf)
- nsslapd-nagle(5dsconf)
- nsslapd-port(5dsconf)
- nsslapd-privatenamespaces(5dsconf)
- nsslapd-readonly(5dsconf)
- nsslapd-referral(5dsconf)
- nsslapd-referralmode(5dsconf)
- nsslapd-reservedescriptors(5dsconf)
- nsslapd-return-exact-case(5dsconf)
- nsslapd-rootdn(5dsconf)
- nsslapd-rootpw(5dsconf)
- nsslapd-rootpwstoragescheme(5dsconf)
- nsslapd-schema-repl-useronly(5dsconf)
- nsslapd-schemacheck(5dsconf)
- nsslapd-securePort(5dsconf)
- nsslapd-securelistenhost(5dsconf)
- nsslapd-security(5dsconf)
- nsslapd-sizelimit(5dsconf)
- nsslapd-threadnumber(5dsconf)
- nsslapd-timelimit(5dsconf)
- nsslapd-versionstring(5dsconf)
- useAuthzIdForAuditAttrs(5dsconf)

Attributes of cn=encryption,cn=config

Encryption related attributes are stored under the cn=encryption,cn=config entry. This entry is an instance of the nsEncryptionConfig object class. For encryption related attributes to be taken into account by the server, this object class, in addition to the top object class, must be present in the entry.

See the following manual pages.

- nsSSL2(5dsconf)
- nsSSL3(5dsconf)
- nsSSL3ciphers(5dsconf)

- nsSSLClientAuth(5dsconf)
- nsSSLServerAuth(5dsconf)
- nsSSLSessionTimeout(5dsconf)

Attributes of cn=mapping tree,cn=config

Configuration attributes for suffixes and replication are stored under the branch cn=mapping tree,cn=config.

Configuration attributes related to suffixes are found under the suffix subentry, which has a DN of the following form.

```
cn="suffixName",cn=mapping tree,cn=config
```

Suffix configuration entries therefore have CNs such as cn="dc=example,dc=com". Suffix configuration entries are instances of the nsMappingTree object class, which inherits from the extensibleObject object class. For suffix configuration attributes to be taken into account by the server, these object classes, in addition to the top object class, must be present in the entry. See the following man pages about suffix configuration entry attributes.

- nsslapd-backend(5dsconf)
- nsslapd-distribution-plugin(5dsconf)
- nsslapd-distribution-funct(5dsconf)
- nsslapd-referral(5dsconf)
- nsslapd-state(5dsconf)

Replication configuration attributes are stored under an entry with a DN of the following form.

```
cn=replica,cn="suffixName",cn=mapping tree,cn=config
```

Replication agreement attributes are stored under an entry with a DN of the following form.

```
cn=replicationAgreementName,cn=replica,cn="suffixName",
 cn=mapping tree,cn=config
```

See replication(5dsconf) for details.

Attributes of cn=Password Policy

The default password policy entry for a Directory Server instance has DN cn=Password Policy,cn=config. For help configuring password policy, see the *Directory Server Administration Guide*.

For details concerning password policy entries, see pwpolicy(5dssd). Entries having the object classes described in pwdPolicy(5dsoc), and in sunPwdPolicy(5dsoc) are used to configure password policy.

For instructions concerning legacy password policy functionality, see the *Directory Server Migration Guide*. Legacy password policy functionality is configured using entries of the object class described in passwordPolicy(5dsoc).

Plug-In Configuration Under cn=plugins

Many of the features of Directory Server are designed as discrete modules that plug into the core server. The configuration for each part of Directory Server plug-in functionality has its own separate entry and set of attributes under the subtree `cn=plugins,cn=config`. The following example shows the configuration entry for the Telephone Syntax plug-in.

```
dn: cn=Telephone Syntax,cn=plugins,cn=config
objectclass: top
objectclass: nsSlapdPlugin
objectclass: ds-signedPlugin
objectclass: extensibleObject
cn: Telephone Syntax
nsslapd-pluginPath: /opt/SUNWdsee/ds6/lib/syntax-plugin.so
nsslapd-pluginInitfunc: tel_init
nsslapd-pluginType: syntax
nsslapd-pluginEnabled: on
...
```

Some of these attributes are common to all plug-ins and some may be particular to a specific plug-in.

Chained Suffix Plug-In Configuration

All plug-in configuration information used by the chained suffix instances is stored under the `cn=chaining database,cn=plugins,cn=config` entry.

The following global chained suffix configuration attributes common to all instances are stored under `cn=config,cn=chaining database,cn=plugins,cn=config`.

- nsActiveChainingComponents(5dsconf)
- nsMaxResponseDelay(5dsconf)
- nsMaxTestResponseDelay(5dsconf)
- nsTransmittedControls(5dsconf)

Default instance chained suffix attributes are stored under `cn=default instance config,cn=chaining database,cn=plugins,cn=config`.

- nsAbandonedSearchCheckInterval(5dsconf)
- nsBindConnectionsLimit(5dsconf)
- nsBindRetryLimit(5dsconf)
- nsBindTimeout(5dsconf)
- nsCheckLocalACI(5dsconf)
- nsConcurrentBindLimit(5dsconf)
- nsConcurrentOperationsLimit(5dsconf)
- nsConnectionLife(5dsconf)
- nsOperationConnectionsLimit(5dsconf)
- nsProxiedAuthorization(5dsconf)
- nsReferralOnScopedSearch(5dsconf)

- nsslapd-sizelimit(5dsconf)
- nsslapd-timelimit(5dsconf)

Instance-specific chained suffix attributes are stored under `cn=`*chainedSuffix*`,cn=chaining database,cn=plugins,cn=config`.

- nsFarmServerURL(5dsconf)
- nshoplimit(5dsconf)
- nsMultiplexorBindDN(5dsconf)
- nsMultiplexorCredentials(5dsconf)

The following list shows the chained suffix attributes used for monitoring activity on instances. These attributes are stored under `cn=monitor,cn=`*dbName*`, cn=chaining database,cn=plugins,cn=config`.

`nsAddCount`	Number of add operations received.
`nsDeleteCount`	Number of delete operations received.
`nsModifyCount`	Number of modify operations received.
`nsRenameCount`	Number of rename operations received.
`nsSearchBaseCount`	Number of base level searches received.
`nsSearchOneLevelCount`	Number of one-level searches received.
`nsSearchSubtreeCount`	Number of subtree searches received.
`nsAbandonCount`	Number of abandon operations received.
`nsBindCount`	Number of bind requests received.
`nsUnbindCount`	Number of unbinds received.
`nsCompareCount`	Number of compare operations received.
`nsOperationConnectionCount`	Number of open connections for normal operations.
`nsBindConnectionCount`	Number of open connections for bind operations.

Database Plug-In Configuration

Database plug-in configuration entries are stored under `cn=ldbm database,cn=plugins,cn=config`. That entry is a server plug-in configuration entry for databases, and therefore takes the same attributes as other plug-in entries.

Key entries beneath the plug-in configuration entry are listed as follows.

`cn=`*attr*`,cn=default indexes,cn=config,cn=ldbm database, cn=plugins,cn=config`
: Configuration entries for default indexes. Notice that each individual attribute type indexed has its own entry, and that the attribute type is identified by common name, CN. See the following man pages concerning attributes for such entries.

- cn(5dsat)
- description(5dsat)
- nsIndexType(5dsconf)
- nsMatchingRule(5dsconf)
- nsSystemIndex(5dsconf)

`cn=`*attr*`,cn=index,cn=`*dbName*`,cn=ldbm database, cn=plugins,cn=config`

Configuration entries for indexing for attributes of the suffix whose backend database has CN *dbName*. Such entries take the same configuration attributes as configuration entries for default indexes.

All indexes, except system-essential ones, can be removed, but care should be taken not to cause unnecessary disruptions.

`cn=config,cn=ldbm database,cn=plugins,cn=config`

Global configuration information for all databases. See the following man pages concerning attributes for such entries.

- nsLookthroughLimit(5dsconf)
- nsslapd-db-checkpoint-interval(5dsconf)
- nsslapd-db-circular-logging(5dsconf)
- nsslapd-db-durable-transactions(5dsconf)
- nsslapd-db-home-directory(5dsconf)
- nsslapd-db-idl-divisor(5dsconf)
- nsslapd-db-locks(5dsconf)
- nsslapd-db-logbuf-size(5dsconf)
- nsslapd-db-logdirectory(5dsconf)
- nsslapd-db-logfile-size(5dsconf)
- nsslapd-db-page-size(5dsconf)
- nsslapd-db-transaction-batch-val(5dsconf)
- nsslapd-db-tx-max(5dsconf)
- nsslapd-dbcachesize(5dsconf)
- nsslapd-dbncache(5dsconf)
- nsslapd-disk-full-threshold(5dsconf)
- nsslapd-disk-low-threshold(5dsconf)
- nsslapd-exclude-from-export(5dsconf)
- nsslapd-import-cachesize(5dsconf)
- nsslapd-mode(5dsconf)

`cn=database,cn=monitor,cn=ldbm database, cn=plugins,cn=config`

Entry for read-only database performance monitoring attributes. All of the values for these attributes are 32-bit integers.

`nsslapd-db-abort-rate`	Number of transactions that have been aborted.
`nsslapd-db-active-txns`	Number of transactions that are currently active (used by the database.)

`nsslapd-db-cache-hit`	Requested pages found in the cache.
`nsslapd-db-cache-region-wait-rate`	Number of times that a thread of control was forced to wait before obtaining the region lock.
`nsslapd-db-cache-size-bytes`	Total cache size in bytes.
`nsslapd-db-cache-try`	Total cache lookups.
`nsslapd-db-clean-pages`	Clean pages currently in the cache.
`nsslapd-db-commit-rate`	Number of transactions that have been committed.
`nsslapd-db-configured-locks`	Configured number of locks.
`nsslapd-db-configured-txns`	Configured number of transactions.
`nsslapd-db-current-locks`	Number of locks currently used by the database.
`nsslapd-db-deadlock-rate`	Number of deadlocks detected.
`nsslapd-db-dirty-pages`	Dirty pages currently in the cache.
`nsslapd-db-hash-buckets`	Number of hash buckets in buffer hash table.
`nsslapd-db-hash-elements-examine-rate`	Total number of hash elements traversed during hash table lookups.
`nsslapd-db-hash-search-rate`	Total number of buffer hash table lookups.
`nsslapd-db-lock-conflicts`	Total number of locks not immediately available due to conflicts.
`nsslapd-db-lockers`	Number of current lockers.
`nsslapd-db-lock-region-wait-rate`	Number of times that a thread of control was forced to wait before obtaining the region lock.
`nsslapd-db-lock-request-rate`	Total number of locks requested.
`nsslapd-db-log-bytes-since-checkpoint`	Number of bytes written to this log since the last checkpoint.
`nsslapd-db-log-flush-commit`	The number of log flushes that contained a transaction commit record.

`nsslapd-db-log-flush-count`	The number of times the log has been flushed to disk.
`nsslapd-db-log-max-commit-per-flush`	The maximum number of commits contained in a single log flush.
`nsslapd-db-log-min-commit-per-flush`	The minimum number of commits contained in a single log flush that contained a commit.
`nsslapd-db-log-region-wait-rate`	Number of times that a thread of control was forced to wait before obtaining the region lock.
`nsslapd-db-log-write-count`	The number of times the log has been written to disk.
`nsslapd-db-log-write-count-fill`	The number of times the log has been written to disk because the in-memory log record cache filled up.
`nsslapd-db-log-write-rate`	Number of bytes written to the log since the last checkpoint.
`nsslapd-db-longest-chain-length`	Longest chain ever encountered in buffer hash table lookups.
`nsslapd-db-max-locks`	Maximum number of locks used by the database since the last startup.
`nsslapd-db-max-txns`	Maximum number of transactions used since the last startup.
`nsslapd-db-page-create-rate`	Pages created in the cache.
`nsslapd-db-page-read-rate`	Pages read into the cache.
`nsslapd-db-page-ro-evict-rate`	Clean pages forced from the cache.
`nsslapd-db-page-rw-evict-rate`	Dirty pages forced from the cache.
`nsslapd-db-pages-in-use`	All pages, clean or dirty, currently in use.
`nsslapd-db-page-trickle-rate`	Dirty pages written using the `memp_trickle` interface.
`nsslapd-db-page-write-rate`	Pages read into the cache.
`nsslapd-db-txn-region-wait-rate`	Number of times that a thread of control was force to wait before obtaining the region lock.

`cn=`*dbName*`,cn=ldbm database,cn=plugins,cn=config`
Configuration information for databases backing suffixes you define. The *dbName* is by default a contraction of the common name for the suffix. For example, if the suffix has CN `cd=example,dc=com`, the *dbName* might be `example`. See the following man pages concerning attributes for such entries.

- nsslapd-cachesize(5dsconf)
- nsslapd-cachememsize(5dsconf)
- nsslapd-directory(5dsconf)
- nsslapd-readonly(5dsconf)
- nsslapd-require-index(5dsconf)
- nsslapd-suffix(5dsconf)

Virtual list view, VLV, index entries are found beneath this entry.

A VLV index provides fast searches against a known result set and sort ordering. To do this, the object class `vlvSearch` is needed to define the VLV search, and the object class `vlvIndex` is needed to order the search. See the following manual pages for details on the VLV configuration entry object classes and attributes.

- vlvBase(5dsat)
- vlvEnabled(5dsat)
- vlvFilter(5dsat)
- vlvScope(5dsat)
- vlvSort(5dsat)
- vlvUses(5dsat)
- vlvIndex(5dsoc)
- vlvSearch(5dsoc)

`cn=default indexes,cn=config,cn=ldbm database, cn=plugins,cn=config`
Configuration entry for default indexing for all suffixes. Default indexes are configured per backend in order to optimize Directory Server functionality for the majority of deployments.

`cn=monitor,cn=`*dbName*`,cn=ldbm database, cn=plugins,cn=config`
Entry for database monitoring attributes, listing database statistics for monitoring activity on the *dbName*database. These attributes are provided for each file that makes up your database.

`dbentrycount`
Total number of entries in the database, including entries created by replication.

`dbfilename-number`
This attribute indicates the name of the file and provides a sequential integer identifier, starting at 0, for the file. All associated statistics for the file are given the same numerical identifier.

`dbfilecachehit`
Number of times that a search requiring data from this file was performed and data successfully obtained from the cache.

`dbfilecachemiss`	Number of times that a search requiring data from this file was performed and that the data could not be obtained from the cache.
`dbfilepagein`	Number of pages brought to the cache from this file.
`dbfilepageout`	Number of pages for this file written from cache to disk.
`entrycachehitratio`	Ratio that indicates the number of entry cache tries to successful entry cache lookups.
`entrycachehits`	Total number of successful entry cache lookups.
`ldapentrycount`	Number of user entries in the database.
`maxentrycachecount`	Maximum number of directory entries that are allowed to be maintained in the entry cache.
`maxentrycachesize`	Maximum memory size allowed for entry cache, in bytes.

`cn=monitor,cn=ldbm database,cn=plugins,cn=config`

Entry for database monitoring attributes, listing database statistics for monitoring activity on databases.

`dbcachehits`	Requested pages found in the database.
`dbcachetries`	Total requested pages found in the database cache.
`dbcachehitratio`	Percentage of requested pages found in the database cache, `hits/tries`.
`dbcachepagein`	Pages read into the database cache.
`dbcachepageout`	Pages written from the database cache to the backing file.
`dbcacheroevict`	Clean pages forced from the cache.
`dbcacherwevict`	Dirty pages forced from the cache.

DSML Front End Plug-In Configuration Attributes

The front end plug-in enables you to access directory data by methods other than LDAP. Directory Server provides a DSML front end plug-in that enables access using DSMLv2 over HTTP/SOAP. Attributes for the DSML front end plug-in are stored under `cn=DSMLv2-SOAP-HTTP,cn=frontends,cn=plugins,cn=config`. See the following manual pages for details.

- ds-hdsml-clientauthmethod(5dsconf)
- ds-hdsml-dsmlschemalocation(5dsconf)
- ds-hdsml-iobuffersize(5dsconf)
- ds-hdsml-poolmaxsize(5dsconf)
- ds-hdsml-poolsize(5dsconf)

- ds-hdsml-port(5dsconf)
- ds-hdsml-requestmaxsize(5dsconf)
- ds-hdsml-responsemsgsize(5dsconf)
- ds-hdsml-rooturl(5dsconf)
- ds-hdsml-secureport(5dsconf)
- ds-hdsml-soapschemalocation(5dsconf)

Retro Changelog Plug-In Configuration

The following manual pages describe attributes used when configuring the retro changelog plug-in.

- nsslapd-changelogdir(5dsconf)
- nsslapd-changelogmaxage(5dsconf)
- nsslapd-changelogmaxentries(5dsconf)

Server Plug-In Configuration Entries

All plug-ins are instances of the `nsSlapdPlugin` object class, which in turn inherits from the `extensibleObject` object class. For plug-in configuration attributes to be taken into account by the server, both of these object classes, in addition to the `top` object class, must be present in the entry.

See nsslapd-plugin(5dsconf) for an overview of the plug-ins provided with Directory Server, including configurable options, configurable arguments, default setting, dependencies, general performance related information, and further reading.

Attributes of `cn=uniqueid generator,cn=config`

Unique ID generator configuration attributes are stored under the entry with DN `cn=uniqueid generator,cn=config`. The `cn=uniqueid generator,cn=config` entry is an instance of the `extensibleObject` object class. For unique ID generator configuration attributes to be taken into account by the server, this object class, in addition to the `top` object class, must be present in the entry.

The principal unique ID generator attribute is nsState(5dsconf).

Attributes

This section lists configuration elements whose modifications cannot take effect dynamically, while the server is still running. After modifying these parameters, you must restart the server. The following list shoiws the configuration attributes concerned, with their full DNs, and provides a brief description of their functions.

Any plug-in configuration attribute
  Changing plug-in settings.

`cn=config:nsslapd-port`
  Changing the port number.

`cn=config:nsslapd-secureport`
  Changing the secure port number.

`cn=config:nsslapd-security`
Enabling or disabling use of SSL, TLS, and attribute encryption.

`cn=config:nsslapd-changelogdir`
Modifying the change log database path.

`cn=config:nsslapd-changelogsuffix`
Modifying the change log suffix.

`cn=config:nsslapd-return-exact-case`
Modifying whether the server returns exact case matches for attribute names.

`cn=config,cn=ldbm database,cn=plugins,cn=config:nsslapd-allidsthreshold`
Changing the all IDs threshold value.

`cn=config,cn=ldbm database,cn=plugins,cn=config:nsslapd-dbcachesize`
Modifying the size of the database cache.

`cn=config,cn=ldbm database,cn=plugins,cn=config:nsslapd-dbncache`
Modifying whether the database cache memory is split into equally sized pieces.

`cn=config,cn=ldbm database,cn=plugins,cn=config:nsslapd-directory`
Changing the path to the database instance.

`cn=config,cn=ldbm database,cn=plugins,cn=config:nsslapd-db-locks`
Changing the number of locks available in the database.

`cn=encryption,cn=config:nssslsessiontimeout`
Changing the lifetime of an SSL session.

`cn=encryption,cn=config:nssslclientauth`
Enabling or disabling client authentication.

`cn=encryption,cn=config:nssslserverauth`
Enabling or disabling server authentication.

`cn=encryption,cn=config:nsssl2`
Enabling or disabling SSL Version 2 for Directory Server.

`cn=encryption,cn=config:nsssl3`
Enabling or disabling SSL Version 3 for Directory Server.

`cn=RSA,cn=encryption,cn=config:nsssltoken`
Changing the SSL token.

`cn=RSA,cn=encryption,cn=config:nssslpersonalityssl`
Changing the SSL personality.

`cn=RSA,cn=encryption,cn=config:nssslactivation`
Enabling or disabling the SSL encryption module.

`cn=`*suffixName*`,cn=ldbm database,cn=plugins,cn=config:nsslapd-cachesize`
Modifying the number of entries held in the entry cache.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal as a supported interface after this release

Name hosts_access – Format of host access control files for Directory Server Enterprise Edition

Synopsis *instance-path*/config/hosts.allow
instance-path/config/hosts.deny

Description This manual page describes a simple access control language that is based on client (host name/address, user name), and server (process name, host name/address) patterns. Examples are given at the end. The impatient reader is encouraged to skip to the EXAMPLES section for a quick introduction.

In the following text, daemon is the the process name of a network daemon process, and client is the name and/or address of a host requesting service.

Note that the version of hosts_access supplied with Directory Server Enterprise Edition is different from the version delivered with Solaris. The Directory Server Enterprise Edition version of hosts_access has the following characteristics:

- Shell commands are not available on Microsoft Windows.
- `hosts_options` are not available on any OS.
- IPv6 is supported on all platforms except Windows.
- PARANOID mode is not available.
- You cannot replace the lib with your own lib, as it is statically linked to the server.
- There is no support of NIS Netgroups. Any '@' symbols in rules are ignored.
- The `daemon_list` process name is the port number of the server. For example, `3389:eng.example.com` or `636:192.168.11.254`. Port numbers are server properties: `ldap-port`, `ldap-secure-port`, `dsml-port`, `dsml-secure-port`. Use the dsconf command to view and modify these properties.
- Server instances can share files by pointing the instance name at the same file (*instance-path*/config/hosts.allow). Use the dsconf command to view and modify the server property `host-access-dir-path`. For example, to have all server instances pointing to `/etc/hosts.{deny,allow}`, run the following command on all servers: `$ dsconf set-server-prop –h host –p port host-access-dir-path: /etc`
- You can make changes to the hosts_access or hosts_deny files without needing to restart the server. You can safely ignore the `dsconf` server restart message.
- Two conditions must be met in order for acess to be allowed. Firstly, the permissions of the file(s) must be owned by `nsslapd-localuser` or `root` and secondly write permission must be allowed for the owner, but not for group or other. Ensure that you use correct permissions, as incorrect permissons on shared files can cause problems. Note that these conditions are not checked on Windows platforms.

ACCESS CONTROL FILES The access control software consults two files. The search stops at the first match:

- Access will be granted when a (daemon,client) pair matches an entry in the *instance-path*/config/hosts.allow file.

- Otherwise, access will be denied when a (daemon,client) pair matches an entry in the *instance-path*`/config/hosts.deny` file.
- Otherwise, access will be granted.

A non-existing access control file is treated as if it were an empty file. Thus, access control can be turned off by providing no access control files.

ACCESS CONTROL RULES

Each access control file consists of zero or more lines of text. These lines are processed in order of appearance. The search terminates when a match is found.

- A newline character is ignored when it is preceded by a backslash character. This permits you to break up long lines so that they are easier to edit.
- Blank lines or lines that begin with a '#' character are ignored. This permits you to insert comments and whitespace so that the tables are easier to read.
- All other lines should satisfy the following format:

  Hosts are identified by server port numbers. If there is no port number match or wildcard, the access control check skips that line of the file.

  List elements should be separated by blanks and/or commas.

  All access control checks are case insensitive.

PATTERNS

The access control language implements the following patterns:

- A string that begins with a '.' character. A host name is matched if the last components of its name match the specified pattern. For example, the pattern '.tue.nl' matches the host name 'wzv.win.tue.nl'.
- A string that ends with a '.' character. A host address is matched if its first numeric fields match the given string. For example, the pattern '131.155.' matches the address of (almost) every host on the Eindhoven University network (131.155.x.x).
- An expression of the form 'n.n.n.n/m.m.m.m' is interpreted as a 'net/mask' pair. A host address is matched if 'net' is equal to the bitwise AND of the address and the 'mask'. For example, the net/mask pattern '131.155.72.0/255.255.254.0' matches every address in the range '131.155.72.0' through '131.155.73.255'.
- When using IPv6 for matching, be aware that an expression of the form `[n:n:n:n:n:n:n:n]/m` is interpreted as a `[net]/prefixlen` pair. An IPv6 host address is matched if prefixlen bits of net is equal to the prefixlen bits of the address. For example, the [net]/prefixlen pattern `[3ffe:505:2:1::]/64` matches every address in the range `3ffe:505:2:1::` through `3ffe:505:2:1:ffff:ffff:ffff:ffff`.

WILDCARDS

Wildcards '`*`' and '`?`' can be used to match hostnames or IP addresses. However, this method of matching cannot be used in conjunction with the following: net/mask matching, hostname matching beginning with '`.`' , IP address matching ending with '`.`' or a IPv6 rule (begins with '`[`').

The access control language supports explicit wildcards:

ALL — The universal wildcard, always matches.

LOCAL — Matches any host whose name does not contain a dot character.

UNKNOWN — Matches any user whose name is unknown, and matches any host whose name or address are unknown. This pattern should be used with care: host names may be unavailable due to temporary name server problems. A network address will be unavailable when the software cannot figure out what type of network it is talking to.

Sun does not recommend that you use the UNKNOWN wildcard. Directory Server always fills in both host and address, so there is never a case when the host name is unknown. The user is unavailable because of no NIS netgroups support.

KNOWN — Matches any user whose name is known, and matches any host whose name and address are known. This pattern should be used with care: host names may be unavailable due to temporary name server problems. A network address will be unavailable when the software cannot figure out what type of network it is talking to.

In Directory Server the user is always marked as unknown and is unavailable because of the NIS Netgroup restriction.

OPERATORS

EXCEPT — Intended use is of the form: ‘list_1 EXCEPT list_2'; this construct matches anything that matches list_1 unless it matches list_2. The EXCEPT operator can be used in daemon_lists and in client_lists. The EXCEPT operator can be nested: if the control language would permit the use of parentheses, ‘a EXCEPT b EXCEPT c' would parse as ‘(a EXCEPT (b EXCEPT c))'.

SHELL COMMANDS

Note that shell commands are not available on Microsoft Windows.

- If the first-matched access control rule contains a shell command, that command is subjected to %*letter* substitutions (see next section). The result is executed by a /bin/sh child process with standard input, output and error connected to /dev/null. Specify an ‘&' at the end of the command if you do not want to wait until it has completed.
- Shell commands should not rely on the PATH setting of the inetd. Instead, they should use absolute path names, or they should begin with an explicit PATH=whatever statement.

% EXPANSIONS

The following expansions are available within shell commands:

%a (%A) — The client (server) host address.

%c — Client information: user@host, user@address, a host name, or just an address, depending on how much information is available.

%d — The daemon process name (argv[0] value).

%h (%H) — The client (server) host name or address, if the host name is unavailable.

%n (%N) The client (server) host name (or "unknown").

%p The daemon process id.

%s Server information: daemon@host, daemon@address, or just a daemon name, depending on how much information is available.

%u The client user name (or "unknown").

%% Expands to a single ‘%' character.

Characters in % expansions that may confuse the shell are replaced by underscores.

Examples

The language is flexible enough that different types of access control policy can be expressed with a minimum of fuss. Although the language uses two access control tables, the most common policies can be implemented with one of the tables being trivial or even empty.

When reading the examples below it is important to realize that the allow table is scanned before the deny table, that the search terminates when a match is found, and that access is granted when no match is found at all.

The examples use host and domain names. They can be improved by including address and/or network/netmask information, to reduce the impact of temporary name server lookup failures.

MOSTLY CLOSED

In this case, access is denied by default. Only explicitly authorized hosts are permitted access.

The default policy (no access) is implemented with a trivial deny file:

```
instance-path/config/hosts.deny:  ALL: ALL
```

This denies all service to all hosts, unless they are permitted access by entries in the allow file.

The explicitly authorized hosts are listed in the allow file. For example:

```
instance-path/config/hosts.allow:
ALL: LOCAL ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
```

The first rule permits access from hosts in the local domain (no ‘.' in the host name). The second rule permits access from all hosts in the .foobar.edu domain (notice the leading period), with the exception of terminalserver.foobar.edu.

MOSTLY OPEN

Here, access is granted by default; only explicitly specified hosts are refused service.

The default policy (access granted) makes the allow file redundant so that it can be omitted. The explicitly non-authorized hosts are listed in the deny file. For example:

```
instance-path/config/hosts.deny:
ALL: some.host.name, .some.domain
ALL EXCEPT 1389: other.host.name, .other.domain
```

The first rule denies some hosts and domains all services; the second rule still permits connections to directory port 1389 from other hosts and domains.

BOOBY TRAPS The next example permits requests to Directory Server port 1389 from hosts in the local domain (notice the leading dot). Requests from any other hosts are denied. Instead of the requested file, a finger probe is sent to the offending host. The result is mailed to the superuser.

```
instance-path/config/hosts.allow:
1389: LOCAL, .my.domain
```

```
instance-path/config/hosts.deny:
ALL: (/usr/sfw/sbin/safe_finger -l @%h | \
      /usr/ucb/mail -s %d-%h root) &
```

The above example assumes that the safe_finger command is installed in /usr/sfw/sbin. For Solaris, the safe_finger command is in the SUNWtcpd package. The default location for the safe_finger command in the SUNWtcpd package is "/usr/sfw/sbin". For other operating systems the safe_finger command should be installed in a suitable place. The safe_finger command limits possible damage from data sent by the remote finger server, and gives better protection than the standard finger command. Shell commands for Windows is not supported, so Windows users should not use this rule.

The expansion of the %h (client host) and %d (service name) sequences is described in the section on shell commands.

Warning: do not booby-trap your finger daemon, unless you are prepared for infinite finger loops.

On network firewall systems this trick can be carried even further. The typical network firewall only provides a limited set of services to the outer world. All other services can be "bugged" just like the above tftp example. The result is an excellent early-warning system.

Diagnostics

An error is reported when a syntax error is found in a host access control rule; when the length of an access control rule exceeds the capacity of an internal buffer (2048); when an access control rule is not terminated by a newline character; when the result of %*letter* expansion would overflow an internal buffer; when a system call fails that should not. All problems are written to the Directory Server instance access log.

Files

instance-path/`config/hosts.allow`, (daemon,client) pairs are granted access.
instance-path/`config/hosts.deny`, (daemon,client) pairs are denied access.

See Also

tcpd(1M) tcp/ip daemon wrapper program. tcpdchk(1M), tcpdmatch(1M), test programs.

Bugs

If a name server lookup times out, the host name will not be available to the access control software, even though the host is registered.

Author

Wietse Venema (wietse@wzv.win.tue.nl)

Department of Mathematics and Computing Science

Eindhoven University of Technology

Den Dolech 2, P.O. Box 513,

5600 MB Eindhoven, The Netherlands

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External

REFERENCE

LDAP Schema Collections

Name adminserv – Schema definitions for Administration Server

Description This collection includes attribute types and object classes used by Administration Server.

Object Classes This collection includes the object classes documented in the following additional pages:

nsLicenseUser(5dsoc)

Attribute Types This collection includes the attribute types documented in the following additional pages:

nsLicenseEndTime(5dsat), nsLicenseStartTime(5dsat), nsLicensedFor(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release.

Name attributes – Schema definitions for User Attributes

Description Directory Server supports many user attributes you may use directly or in your own object classes.

Attribute Types This collection includes the attribute types documented in the following additional pages:

abstract(5dsat), aliasedObjectName(5dsat), associatedDomain(5dsat), associatedName(5dsat), attributeTypes(5dsat), audio(5dsat), authorCn(5dsat), authorSn(5dsat), authorityRevocationList(5dsat), bootFile(5dsat), bootParameter(5dsat), buildingName(5dsat), businessCategory(5dsat), c(5dsat), cACertificate(5dsat), carLicense(5dsat), certificateRevocationList(5dsat), changeLog(5dsat), changeNumber(5dsat), changeTime(5dsat), changeType(5dsat), changes(5dsat), cn(5dsat), co(5dsat), cosAttribute(5dsat), cosIndirectSpecifier(5dsat), cosPriority(5dsat), cosspecifier(5dsat), costargettree(5dsat), costemplatedn(5dsat), crossCertificatePair(5dsat), dITContentRules(5dsat), dITStructureRules(5dsat), dNSRecord(5dsat), dSAQuality(5dsat), dc(5dsat), deleteOldRdn(5dsat), deltaRevocationList(5dsat), departmentNumber(5dsat), description(5dsat), destinationIndicator(5dsat), displayName(5dsat), distinguishedName(5dsat), ditRedirect(5dsat), dmdName(5dsat), documentAuthor(5dsat), documentIdentifier(5dsat), documentLocation(5dsat), documentPublisher(5dsat), documentStore(5dsat), documentTitle(5dsat), documentVersion(5dsat), drink(5dsat), employeeNumber(5dsat), employeeType(5dsat), enhancedSearchGuide(5dsat), facsimileTelephoneNumber(5dsat), gecos(5dsat), generationQualifier(5dsat), gidNumber(5dsat), givenName(5dsat), homeDirectory(5dsat), homePhone(5dsat), homePostalAddress(5dsat), host(5dsat), houseIdentifier(5dsat), info(5dsat), initials(5dsat), internationaliSDNNumber(5dsat), ipHostNumber(5dsat), ipNetmaskNumber(5dsat), ipNetworkNumber(5dsat), ipProtocolNumber(5dsat), ipServicePort(5dsat), ipServiceProtocol(5dsat), janetMailbox(5dsat), javaClassName(5dsat), javaClassNames(5dsat), javaCodebase(5dsat), javaDoc(5dsat), javaFactory(5dsat), javaReferenceAddress(5dsat), javaSerializedData(5dsat), jpegPhoto(5dsat), keyWords(5dsat), knowledgeInformation(5dsat), l(5dsat), labeledUri(5dsat), lastModifiedBy(5dsat), lastModifiedTime(5dsat), loginShell(5dsat), macAddress(5dsat), mail(5dsat), mailPreferenceOption(5dsat), manager(5dsat), matchingRuleUse(5dsat), matchingRules(5dsat), member(5dsat), memberCertificateDescription(5dsat), memberNisNetgroup(5dsat), memberURL(5dsat), memberUid(5dsat), mobile(5dsat), multiLineDescription(5dsat), name(5dsat), nameForms(5dsat), newRdn(5dsat), newSuperior(5dsat), nisMapEntry(5dsat), nisMapName(5dsat), nisNetgroupTriple(5dsat), nsLicensedFor(5dsat), nsRoleFilter(5dsat), nsRoleScopeDn(5dsat), o(5dsat), objectClass(5dsat), objectClasses(5dsat), obsoletedByDocument(5dsat), obsoletesDocument(5dsat), oncRpcNumber(5dsat), organizationalStatus(5dsat), ou(5dsat), otherMailbox(5dsat), owner(5dsat), pager(5dsat), passwordChange(5dsat), passwordCheckSyntax(5dsat), passwordExp(5dsat), passwordExpireWithoutWarning(5dsat), passwordInHistory(5dsat), passwordLockout(5dsat), passwordLockoutDuration(5dsat), passwordMaxAge(5dsat), passwordMaxFailure(5dsat), passwordMinAge(5dsat), passwordMinLength(5dsat), passwordMustChange(5dsat), passwordResetDuration(5dsat), passwordResetFailureCount(5dsat), passwordRootdnMayBypassModsChecks(5dsat), passwordStorageScheme(5dsat), passwordUnlock(5dsat), passwordWarning(5dsat), personalSignature(5dsat), personalTitle(5dsat), photo(5dsat), physicalDeliveryOfficeName(5dsat), postOfficeBox(5dsat), postalAddress(5dsat), postalCode(5dsat), preferredDeliveryMethod(5dsat), preferredLanguage(5dsat), presentationAddress(5dsat), protocolInformation(5dsat), pwdAttribute(5dsat), pwdIsLockoutPrioritized(5dsat), pwdKeepLastAuthTime(5dsat), ref(5dsat), registeredAddress(5dsat), roleOccupant(5dsat), roomNumber(5dsat), searchGuide(5dsat), secretary(5dsat), seeAlso(5dsat), serialNumber(5dsat), shadowExpire(5dsat), shadowFlag(5dsat), shadowInactive(5dsat), shadowLastChange(5dsat), shadowMax(5dsat), shadowMin(5dsat), shadowWarning(5dsat), singleLevelQuality(5dsat), sn(5dsat), st(5dsat), street(5dsat), subject(5dsat), subtreeMaximumQuality(5dsat), subtreeMinimumQuality(5dsat), supportedAlgorithms(5dsat), supportedApplicationContext(5dsat), targetDn(5dsat), telephoneNumber(5dsat), teletexTerminalIdentifier(5dsat), telexNumber(5dsat), textEncodedORAddress(5dsat), title(5dsat), uid(5dsat), uidNumber(5dsat), uniqueIdentifier(5dsat), uniqueMember(5dsat), updatedByDocument(5dsat), updatesDocument(5dsat), userCertificate(5dsat), userClass(5dsat), userPKCS12(5dsat), userPassword(5dsat), userSMIMECertificate(5dsat), x121Address(5dsat), x500UniqueIdentifier(5dsat)

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name changelog – Schema definitions for Changelog Internet Draft

Description The Changelog Internet Draft describes how to store LDAP change records in LDAP entries. Directory Server supports schema definitions corresponding to those described in the draft.

Object Classes This collection includes the object classes documented in the following additional pages:

changeLogEntry(5dsoc)

Attribute Types This collection includes the attribute types documented in the following additional pages:

changeLog(5dsat), changeNumber(5dsat), changeType(5dsat), changes(5dsat), deleteOldRdn(5dsat), newRdn(5dsat), newSuperior(5dsat), targetDn(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External

Name dirserv – Schema definitions for Directory Server

Description This collection includes attribute types and object classes used by Directory Server.

Object Classes This collection includes the object classes documented in the following additional pages:

cosClassicDefinition(5dsoc), cosDefinition(5dsoc), cosIndirectDefinition(5dsoc), cosPointerDefinition(5dsoc), cosSuperDefinition(5dsoc), costemplate(5dsoc), groupOfCertificates(5dsoc), groupOfURLs(5dsoc), nsComplexRoleDefinition(5dsoc), nsFilteredRoleDefinition(5dsoc), nsManagedRoleDefinition(5dsoc), nsNestedRoleDefinition(5dsoc), nsRoleDefinition(5dsoc), nsSimpleRoleDefinition(5dsoc), passwordPolicy(5dsoc), pwdPolicy(5dsoc), sunPwdPolicy(5dsoc), vlvIndex(5dsoc), vlvSearch(5dsoc)

Attribute Types This collection includes the attribute types documented in the following additional pages:

accountUnlockTime(5dsat), aci(5dsat), changeTime(5dsat), copiedFrom(5dsat), copyingFrom(5dsat), cosAttribute(5dsat), cosIndirectSpecifier(5dsat), cosPriority(5dsat), cosspecifier(5dsat), costargettree(5dsat), costemplatedn(5dsat), ds-pluginDigest(5dsat), ds-pluginSignature(5dsat), memberCertificateDescription(5dsat), memberURL(5dsat), nsRole(5dsat), nsRoleDN(5dsat), nsRoleFilter(5dsat), nsRoleScopeDn(5dsat), nsds5ReplConflict(5dsat), passwordAllowChangeTime(5dsat), passwordChange(5dsat), passwordCheckSyntax(5dsat), passwordExp(5dsat), passwordExpWarned(5dsat), passwordExpirationTime(5dsat), passwordExpireWithoutWarning(5dsat), passwordHistory(5dsat), passwordInHistory(5dsat), passwordLockout(5dsat), passwordLockoutDuration(5dsat), passwordMaxAge(5dsat), passwordMaxFailure(5dsat), passwordMinAge(5dsat), passwordMinLength(5dsat), passwordMustChange(5dsat), passwordNonRootMayResetUserpwd(5dsat), passwordPolicySubentry(5dsat), passwordResetDuration(5dsat), passwordResetFailureCount(5dsat), passwordRetryCount(5dsat), passwordUnlock(5dsat), passwordWarning(5dsat), pwdAllowUserChange(5dsat), pwdAttribute(5dsat), pwdChangedTime(5dsat), pwdCheckQuality(5dsat), pwdExpireWarning(5dsat), pwdFailureCountInterval(5dsat), pwdGraceAuthNLimit(5dsat), pwdInHistory(5dsat), pwdLockout(5dsat), pwdLockoutDuration(5dsat), pwdMaxAge(5dsat), pwdMaxFailure(5dsat), pwdMinAge(5dsat), pwdMinLength(5dsat), pwdMustChange(5dsat), pwdSafeModify(5dsat), retryCountResetTime(5dsat), supportedSSLCiphers(5dsat), vlvBase(5dsat), vlvEnabled(5dsat), vlvFilter(5dsat), vlvScope(5dsat), vlvSort(5dsat), vlvUses(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name idpilot – Schema definitions for Internet directory pilot

Description The Internet Directory Pilot defines a number of common directory attribute types and object classes, mostly defined in RFC 1274. Directory Server supports schema definitions corresponding to those described in the pilot.

Object Classes This collection includes the object classes documented in the following additional pages:

RFC822localPart(5dsoc)

Attribute Types This collection includes the attribute types documented in the following additional pages:

dNSRecord(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External

Name numsub – Schema definitions for numSubordinates Internet Draft

Description The numSubordinates Internet Draft defines an operational attribute to hold the number of immediate subordinates of a directory entry. Directory Server supports the schema definition corresponding to those described in the draft.

Attribute Types This collection includes the attribute types documented in the following additional pages:

numSubordinates(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External

Name

objects – Schema definitions for Object Classes

Description

Directory Server supports many object classes you may use either directly or as the basis for your own object classes. An object class that inherits from another object class must appear after this object class in `99user.ldif`, otherwise Directory Server will not start.

Note – Schema provided with Directory Server differ from RFC 2256 with regard to the groupOfNames and groupOfUniqueNames object classes. In the schema provided, the member and uniquemember attribute types are optional, while RFC 2256 specifies that at least one value for these types must be present in the respective object class.

Also, the LDAP RFCs (and X.500 standards) allow for an object class to have more than one superior. This behavior is not currently supported by Directory Server.

Object Classes

This collection includes the object classes documented in the following additional pages:

RFC822localPart(5dsoc), account(5dsoc), alias(5dsoc), applicationEntity(5dsoc), bootableDevice(5dsoc), changeLogEntry(5dsoc), cosClassicDefinition(5dsoc), cosDefinition(5dsoc), cosIndirectDefinition(5dsoc), cosPointerDefinition(5dsoc), cosSuperDefinition(5dsoc), costemplate(5dsoc), country(5dsoc), dSA(5dsoc), dcObject(5dsoc), device(5dsoc), document(5dsoc), documentSeries(5dsoc), domain(5dsoc), domainRelatedObject(5dsoc), extensibleObject(5dsoc), friendlyCountry(5dsoc), groupOfCertificates(5dsoc), groupOfNames(5dsoc), groupOfURLs(5dsoc), groupOfUniqueNames(5dsoc), ieee802Device(5dsoc), inetOrgPerson(5dsoc), ipHost(5dsoc), ipNetwork(5dsoc), ipProtocol(5dsoc), ipService(5dsoc), javaContainer(5dsoc), javaMarshalledObject(5dsoc), javaNamingReference(5dsoc), javaObject(5dsoc), javaSerializedObject(5dsoc), labeledURIObject(5dsoc), ldapSubEntry(5dsoc), locality(5dsoc), newPilotPerson(5dsoc), nisMap(5dsoc), nisNetgroup(5dsoc), nisObject(5dsoc), nsComplexRoleDefinition(5dsoc), nsFilteredRoleDefinition(5dsoc), nsLicenseUser(5dsoc), nsManagedRoleDefinition(5dsoc), nsNestedRoleDefinition(5dsoc), nsRoleDefinition(5dsoc), nsSimpleRoleDefinition(5dsoc), oncRpc(5dsoc), organization(5dsoc), organizationalPerson(5dsoc), organizationalRole(5dsoc), organizationalUnit(5dsoc), passwordPolicy(5dsoc), person(5dsoc), pilotObject(5dsoc), pilotOrganization(5dsoc), posixAccount(5dsoc), posixGroup(5dsoc), pwdPolicy(5dsoc), referral(5dsoc), residentialPerson(5dsoc), room(5dsoc), shadowAccount(5dsoc), simpleSecurityObject(5dsoc), strongAuthenticationUser(5dsoc), subschema(5dsoc), sunPwdPolicy(5dsoc), top(5dsoc)

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name operational – Schema definitions for Operational Attributes

Description Directory Server supports many operational attributes. Operational attributes may be available for use on every entry in the directory, regardless of whether they are defined for the object class of the entry. Operational attributes are returned in a search operation only if they are specifically requested.

Attribute Types This collection includes the attribute types documented in the following additional pages:

accountUnlockTime(5dsat), aci(5dsat), changeHasReplFixupOp(5dsat), changeIsReplFixupOp(5dsat), copiedFrom(5dsat), copyingFrom(5dsat), deletedEntryAttrs(5dsat), ds-pluginDigest(5dsat), ds-pluginSignature(5dsat), isMemberOf(5dsat), ldapSyntaxes(5dsat), namingContexts(5dsat), nsIdleTimeout(5dsat), nsLookThroughLimit(5dsat), nsRole(5dsat), nsds5ReplConflict(5dsat), nsRoleDN(5dsat), nsSizeLimit(5dsat), nsTimeLimit(5dsat), numSubordinates(5dsat), passwordAllowChangeTime(5dsat), passwordExpWarned(5dsat), passwordExpirationTime(5dsat), passwordHistory(5dsat), passwordPolicySubentry(5dsat), passwordRetryCount(5dsat), pwdAccountLockedTime(5dsat), pwdChangedTime(5dsat), pwdFailureTime(5dsat), pwdGraceUseTime(5dsat), pwdHistory(5dsat), pwdLastAuthTime(5dsat), pwdPolicySubentry(5dsat), pwdReset(5dsat), replicaIdentifier(5dsat), replicationCSN(5dsat), retryCountResetTime(5dsat), subschemaSubentry(5dsat), supportedControl(5dsat), supportedExtension(5dsat), supportedLDAPVersion(5dsat), supportedSASLMechanisms(5dsat), targetUniqueId(5dsat), vendorName(5dsat), vendorVersion(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name pwpolicy – Schema definitions for Password Policy Internet Draft

Description The Password Policy Internet Draft describes attribute types and object classes for holding password policy configuration data. Directory Server supports schema definitions corresponding to those described in the draft.

See `sunPwdPolicy`(5dsoc) for additional Directory Server extensions.

Object Classes This collection includes the object classes documented in the following additional pages:

pwdPolicy(5dsoc)

Attribute Types This collection includes the attribute types documented in the following additional pages:

pwdAccountLockedTime(5dsat), pwdAllowUserChange(5dsat), pwdAttribute(5dsat), pwdChangedTime(5dsat), pwdCheckQuality(5dsat), pwdExpireWarning(5dsat), pwdFailureCountInterval(5dsat), pwdFailureTime(5dsat), pwdGraceAuthNLimit(5dsat), pwdGraceUseTime(5dsat), pwdHistory(5dsat), pwdInHistory(5dsat), pwdLockout(5dsat), pwdLockoutDuration(5dsat), pwdMaxAge(5dsat), pwdMaxFailure(5dsat), pwdMinAge(5dsat), pwdMinLength(5dsat), pwdMustChange(5dsat), pwdPolicySubentry(5dsat), pwdReset(5dsat), pwdSafeModify(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External

Name

rfc2079 – Schema definitions for rfc 2079

Description

RFC 2079 defines X.500 attribute types and object classes for holding Uniform Resource Identifiers (URI). Directory Server supports schema definitions corresponding to those described in RFC 2079.

Object Classes

This collection includes the object classes documented in the following additional pages:

labeledURIObject(5dsoc)

Attribute Types

This collection includes the attribute types documented in the following additional pages:

labeledUri(5dsat)

Attributes

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard

Name rfc2247 – Schema definitions for rfc 2247

Description RFC 2247 specifies mappings for DNS domain names onto directory attribute types and object classes. Directory Server supports schema definitions corresponding to those described in RFC 2247.

Object Classes This collection includes the object classes documented in the following additional pages:

dcObject(5dsoc), domain(5dsoc)

Attribute Types This collection includes the attribute types documented in the following additional pages:

dc(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard

Name rfc2307 – Schema definitions for rfc 2307

Description RFC 2307 defines an approach for using LDAP as a Network Information Service. Directory Server supports schema definitions corresponding to those described in RFC 2307.

Object Classes This collection includes the object classes documented in the following additional pages:

bootableDevice(5dsoc), ieee802Device(5dsoc), ipHost(5dsoc), ipNetwork(5dsoc), ipProtocol(5dsoc), ipService(5dsoc), nisMap(5dsoc), nisNetgroup(5dsoc), nisObject(5dsoc), oncRpc(5dsoc), posixAccount(5dsoc), posixGroup(5dsoc), shadowAccount(5dsoc)

Attribute Types This collection includes the attribute types documented in the following additional pages:

bootFile(5dsat), bootParameter(5dsat), gecos(5dsat), gidNumber(5dsat), homeDirectory(5dsat), ipHostNumber(5dsat), ipNetmaskNumber(5dsat), ipNetworkNumber(5dsat), ipProtocolNumber(5dsat), ipServicePort(5dsat), ipServiceProtocol(5dsat), loginShell(5dsat), macAddress(5dsat), memberNisNetgroup(5dsat), memberUid(5dsat), nisMapEntry(5dsat), nisMapName(5dsat), nisNetgroupTriple(5dsat), oncRpcNumber(5dsat), shadowExpire(5dsat), shadowFlag(5dsat), shadowInactive(5dsat), shadowLastChange(5dsat), shadowMax(5dsat), shadowMin(5dsat), shadowWarning(5dsat), uidNumber(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard

Name rfc2713 – Schema definitions for rfc 2713

Description RFC 2713 defines schema for representing Java objects in an LDAP directory. Directory Server supports schema definitions corresponding to those described in RFC 2713.

Object Classes This collection includes the object classes documented in the following additional pages:

javaContainer(5dsoc), javaMarshalledObject(5dsoc), javaNamingReference(5dsoc), javaObject(5dsoc), javaSerializedObject(5dsoc)

Attribute Types This collection includes the attribute types documented in the following additional pages:

javaClassName(5dsat), javaClassNames(5dsat), javaCodebase(5dsat), javaDoc(5dsat), javaFactory(5dsat), javaReferenceAddress(5dsat), javaSerializedData(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard

Name rfc2798 – Schema definitions for rfc 2798

Description RFC 2798 defines the `inetOrgPerson` object class and relevant attribute types. Directory Server supports schema definitions corresponding to those described in RFC 2798.

Object Classes This collection includes the object classes documented in the following additional pages:

inetOrgPerson(5dsoc)

Attribute Types This collection includes the attribute types documented in the following additional pages:

carLicense(5dsat), departmentNumber(5dsat), displayName(5dsat), employeeNumber(5dsat), employeeType(5dsat), jpegPhoto(5dsat), preferredLanguage(5dsat), userPKCS12(5dsat), userSMIMECertificate(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard

Name rfc3045 – Schema definitions for rfc 3045

Description RFC 3045 specifies how to store vendor information in the LDAP root DSE. Directory Server supports schema definitions corresponding to those described in RFC 3045.

Attribute Types This collection includes the attribute types documented in the following additional pages:

vendorName(5dsat), vendorVersion(5dsat)

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard

Name rfc3296 – Schema definitions for rfc 3296

Description RFC 3296 defines schema and protocol elements for representing and manipulating named subordinate references in LDAP directories. Directory Server supports schema definitions corresponding to those described in RFC 3296.

Object Classes This collection includes the object classes documented in the following additional pages:

referral(5dsoc)

Attribute Types This collection includes the attribute types documented in the following additional pages:

ref(5dsat)

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard

Name rfc4512 – Schema definitions for rfc 4512

Description RFC 4512 describes the X.500 Directory Information Models, as used in LDAP.

Object Classes This collection includes the object classes documented in the following additional pages:

extensibleObject(5dsoc), subschema(5dsoc)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard

Name rfc4517 – Schema definitions for rfc 4517

Description RFC 4517 specifies a set of syntaxes and matching rules for LDAP v3 attribute values. Directory Server supports schema definitions corresponding to those described in RFC 4517.

Attribute Types This collection includes the attribute types documented in the following additional pages:

attributeTypes(5dsat), dITContentRules(5dsat), dITStructureRules(5dsat), ldapSyntaxes(5dsat), matchingRuleUse(5dsat), matchingRules(5dsat), nameForms(5dsat), namingContexts(5dsat), objectClasses(5dsat), subschemaSubentry(5dsat), supportedControl(5dsat), supportedExtension(5dsat), supportedLDAPVersion(5dsat), supportedSASLMechanisms(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard

Name rfc4519 – Schema definitions for rfc 4519

Description RFC 4519 summarizes X.500 user schema for use with LDAP v3 directories. Directory Server supports schema definitions corresponding to those described in RFC 4519.

Object Classes This collection includes the object classes documented in the following additional pages:

alias(5dsoc), applicationEntity(5dsoc), country(5dsoc), dSA(5dsoc), device(5dsoc), groupOfNames(5dsoc), groupOfUniqueNames(5dsoc), locality(5dsoc), organization(5dsoc), organizationalPerson(5dsoc), organizationalRole(5dsoc), organizationalUnit(5dsoc), person(5dsoc), residentialPerson(5dsoc), strongAuthenticationUser(5dsoc), top(5dsoc)

Attribute Types This collection includes the attribute types documented in the following additional pages:

aliasedObjectName(5dsat), authorityRevocationList(5dsat), businessCategory(5dsat), c(5dsat), cACertificate(5dsat), certificateRevocationList(5dsat), cn(5dsat), crossCertificatePair(5dsat), deltaRevocationList(5dsat), description(5dsat), destinationIndicator(5dsat), distinguishedName(5dsat), dmdName(5dsat), enhancedSearchGuide(5dsat), facsimileTelephoneNumber(5dsat), generationQualifier(5dsat), givenName(5dsat), houseIdentifier(5dsat), initials(5dsat), internationaliSDNNumber(5dsat), knowledgeInformation(5dsat), l(5dsat), member(5dsat), name(5dsat), o(5dsat), objectClass(5dsat), ou(5dsat), owner(5dsat), physicalDeliveryOfficeName(5dsat), postOfficeBox(5dsat), postalAddress(5dsat), postalCode(5dsat), preferredDeliveryMethod(5dsat), presentationAddress(5dsat), protocolInformation(5dsat), registeredAddress(5dsat), roleOccupant(5dsat), searchGuide(5dsat), seeAlso(5dsat), serialNumber(5dsat), sn(5dsat), st(5dsat), street(5dsat), supportedAlgorithms(5dsat), supportedApplicationContext(5dsat), telephoneNumber(5dsat), teletexTerminalIdentifier(5dsat), telexNumber(5dsat), title(5dsat), uniqueMember(5dsat), userCertificate(5dsat), userPassword(5dsat), x121Address(5dsat), x500UniqueIdentifier(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard

Name rfc4523 – Schema definitions for rfc 4523

Description RFC 4523 describes schema for representing X.509 certificates, X.521 security information, and related elements in directories accessible using the Lightweight Directory Access Protocol (LDAP). Directory Server supports schema definitions corresponding to those described in RFC 4523.

Object Classes This collection includes the object classes documented in the following additional pages:

certificationAuthority, certificationAuthority-V2, cRLDistributionPoint, strongAuthenticationUser(5dsoc), userSecurityInformation

Attribute Types This collection includes the attribute types documented in the following additional pages:

authorityRevocationList(5dsat), cACertificate(5dsat), certificateRevocationList(5dsat), crossCertificatePair(5dsat), deltaRevocationList(5dsat), supportedAlgorithms(5dsat), userCertificate(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard

Name rfc4524 – Schema definitions for rfc 4524

Description RFC 4524 specifies schema for use in the COSINE and Internet X.500 pilots. Directory Server supports schema definitions corresponding to those described in RFC 4524.

Object Classes This collection includes the object classes documented in the following additional pages:

account(5dsoc), document(5dsoc), documentSeries(5dsoc), domainRelatedObject(5dsoc), friendlyCountry(5dsoc), pilotObject(5dsoc), pilotOrganization(5dsoc), room(5dsoc), simpleSecurityObject(5dsoc)

Attribute Types This collection includes the attribute types documented in the following additional pages:

associatedDomain(5dsat), associatedName(5dsat), audio(5dsat), buildingName(5dsat), co(5dsat), dSAQuality(5dsat), ditRedirect(5dsat), documentAuthor(5dsat), documentIdentifier(5dsat), documentLocation(5dsat), documentPublisher(5dsat), documentTitle(5dsat), documentVersion(5dsat), drink(5dsat), homePhone(5dsat), homePostalAddress(5dsat), host(5dsat), info(5dsat), janetMailbox(5dsat), lastModifiedBy(5dsat), lastModifiedTime(5dsat), mail(5dsat), mailPreferenceOption(5dsat), manager(5dsat), mobile(5dsat), organizationalStatus(5dsat), otherMailbox(5dsat), pager(5dsat), personalSignature(5dsat), personalTitle(5dsat), photo(5dsat), roomNumber(5dsat), secretary(5dsat), singleLevelQuality(5dsat), subtreeMaximumQuality(5dsat), subtreeMinimumQuality(5dsat), textEncodedORAddress(5dsat), uid(5dsat), uniqueIdentifier(5dsat), userClass(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard

Name sasl – Schema definitions for sasl

Description This collection reflects the object class and attributes for SASL configuration. The `dse.ldif` entry governing SASL configuration has DN `cn=SASL, cn=security, cn=config`.

Object Classes This collection includes the object classes documented in the following additional pages:

dsSaslConfig(5dsoc)

Attribute Types This collection includes the attribute types documented in the following additional pages:

dsSaslMaxBufSize(5dsat), dsSaslMaxSSF(5dsat), dsSaslMinSSF(5dsat), dsSaslPluginsEnable(5dsat), dsSaslPluginsPath(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name subentry – Schema definitions for LDAP Subentry Internet Draft

Description The LDAP Subentry Internet Draft describes an object class that may be used to indicate operations and management-related entries in the directory, called LDAP Subentries. Directory Server supports schema definitions corresponding to those described in the draft.

Object Classes This collection includes the object classes documented in the following additional pages:

ldapSubEntry(5dsoc)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External

Name

wppilot – Schema definitions for Internet White Pages Pilot

Description

The Internet White Pages Pilot specifies object classes and attribute types for interoperable white pages including name, email, address, and other contact data. Directory Server supports schema definitions corresponding to those used in the pilot.

Object Classes

This collection includes the object classes documented in the following additional pages:

newPilotPerson(5dsoc)

Attribute Types

This collection includes the attribute types documented in the following additional pages:

abstract(5dsat), authorCn(5dsat), authorSn(5dsat), documentStore(5dsat), keyWords(5dsat), multiLineDescription(5dsat), obsoletedByDocument(5dsat), obsoletesDocument(5dsat), subject(5dsat), updatedByDocument(5dsat), updatesDocument(5dsat)

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External

REFERENCE

LDAP Schema Attribute Types

Name abstract – Pilot attribute type

Synopsis

```
( 0.9.2342.19200300.102.1.9
 NAME ’abstract’
 DESC ’Pilot attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’Internet White Pages Pilot’ )
```

Description Provides an abstract of a document entry.

Syntax Directory String, multi-valued.

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Internet White Pages Pilot

Name accountUnlockTime – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.95
 NAME ’accountUnlockTime’
 DESC ’Sun ONE defined password policy attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
 SINGLE-VALUE
 USAGE directoryOperation
 X-DS-USE ’internal’
 X-ORIGIN ’Sun ONE Directory Server’ )
```

Description Indicates the exact time after which a user can attempt to bind to the directory (after an account lockout). This attribute is used only when the password policy is enabled.

Syntax Generalized Time, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name aci – Sun ONE defined access control information attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.55
 NAME 'aci'
 DESC 'Sun ONE defined access control information attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 USAGE directoryOperation
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Used by Directory Server to evaluate what rights are granted or denied when it receives an LDAP request from a client. Note that this is an operational attribute. It is not returned in a search unless you explicitly request it.

Syntax IA5 String, multi-valued.

Usage Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name aliasedObjectName – Standard LDAP attribute type

Synopsis

```
( 2.5.4.1
 NAME 'aliasedObjectName'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 SINGLE-VALUE
 X-ORIGIN 'RFC 2256' )
```

Description This attribute is defined in RFC 2256, but Directory Server does not support alias dereferencing. The value of `aliasedObjectName` attributes are never used by Directory Server.

Syntax DN, single-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name associatedDomain – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.37
 NAME ’associatedDomain’
 DESC ’Standard LDAP attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’RFC 1274’ )
```

Description Specifies a DNS domain associated with an object in the directory tree. For Example, the entry in the directory tree with a distinguished name `c=US, o=example Corporation` might be associated to the domain `example.com`. Note that all domains should be represented in rfc822 order.

Syntax Directory String, multi-valued.

Examples

```
associatedDomain: example.com
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name associatedName – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.38
 NAME 'associatedName'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-ORIGIN 'RFC 1274' )
```

Description Specifies an entry in the organizational directory tree associated with a DNS domain.

Syntax DN, multi-valued.

Examples `associatedName: c=us`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name attributeTypes – Standard LDAP attribute type

Synopsis

```
( 2.5.21.5
 NAME ’attributeTypes’
 DESC ’Standard LDAP attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’RFC 2252’ )
```

Description Multi-valued attribute that specifies the attribute types used within a subschema. Each value describes a single attribute.

Syntax Directory String, multi-valued.

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2252

Name audio – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.55
 NAME 'audio'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
 X-ORIGIN 'RFC 1274' )
```

Description Contains a sound file in binary format. The attribute uses a u-law encoded sound file.

Syntax Binary, multi-valued.

Examples

```
audio:: AAAAAA==
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name authorCn, documentauthorcommonname – Pilot attribute type

Synopsis

```
( 0.9.2342.19200300.102.1.11
 NAME ( ’authorCn’ ’documentauthorcommonname’ )
 DESC ’Pilot attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’Internet White Pages Pilot’ )
```

Description Contains the common name of the author of a document entry.

Syntax Directory String, multi-valued.

Examples

```
authorCn: Mark Craig
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Internet White Pages Pilot

Name authorityRevocationList – Standard LDAP attribute type

Synopsis

```
( 2.5.4.38
 NAME 'authorityRevocationList'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
 X-ORIGIN 'RFC 2256' )
```

Description Contains a list of CA certificates that have been revoked. This attribute is to be stored and requested in the binary form, as `authorityRevocationList;binary`.

Syntax Binary, multi-valued.

Examples

```
authorityRevocationList;binary:: AAAAAA==
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name authorSn, documentauthorsurname – Pilot attribute type

Synopsis

```
( 0.9.2342.19200300.102.1.12
 NAME ( ’authorSn’ ’documentauthorsurname’ )
 DESC ’Pilot attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’Internet White Pages Pilot’ )
```

Description Contains the surname of the author of a document entry.

Syntax Directory String, multi-valued.

Examples `authorSn: Doe`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Internet White Pages Pilot

Name bootFile – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.24
 NAME 'bootFile'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 X-ORIGIN 'RFC 2307' )
```

Description The name of the boot image.

Syntax IA5 String, multi-valued.

Examples

```
bootFile: mach
```

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name bootParameter – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.23
 NAME 'bootParameter'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 X-ORIGIN 'RFC 2307' )
```

Description Specified boot parameters.

Syntax IA5 String, multi-valued.

Examples

```
bootParameter: root=fs:/nfsroot/peg
bootParameter: swap=fs:/nfsswap/peg
bootParameter: dump=fs:/nfsdump/peg
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name buildingName – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.48
 NAME 'buildingName'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 1274' )
```

Description Defines the building name associated with the entry.

Syntax Directory String, multi-valued.

Examples

```
buildingName: EGNB07
```

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name businessCategory – Standard LDAP attribute type

Synopsis

```
( 2.5.4.15
 NAME 'businessCategory'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2256' )
```

Description Identifies the type of business in which the entry is engaged. This should be a broad generalization such as is made at the corporate division level.

Syntax Directory String, multi-valued.

Examples

```
businessCategory: Engineering
```

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name c, countryName – Standard LDAP attribute type

Synopsis

```
( 2.5.4.6
 NAME ( ’c’ ’countryName’ )
 DESC ’Standard LDAP attribute type’
 SUP name
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-ORIGIN ’RFC 2256’ )
```

Description This attribute is designed to contain the two-character code representing a country name, as defined by ISO, in the directory.

Syntax Directory String, single-valued.

Examples

```
c: FR
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name cACertificate – Standard LDAP attribute type

Synopsis

```
( 2.5.4.37
 NAME 'cACertificate'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
 X-ORIGIN 'RFC 2256' )
```

Description Contains the CA's certificate. This attribute is to be stored and requested in the binary form, as `CACertificate;binary`.

Syntax Binary, multi-valued.

Examples

```
CACertificate;binary:: AAAAAA==
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name carLicense – inetOrgPerson attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.1
 NAME 'carLicense'
 DESC 'inetOrgPerson attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'inetOrgPerson Internet Draft' )
```

Description Identifies the entry's automobile license plate number.

Syntax Directory String, multi-valued.

Examples

```
carLicense: 4MCS389
```

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, RFC 2798

Name certificateRevocationList – Standard LDAP attribute type

Synopsis

```
( 2.5.4.39
 NAME 'certificateRevocationList'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
 X-ORIGIN 'RFC 2256' )
```

Description Contains a list of revoked user certificates. This attribute is to be stored and requested in the binary form, as `certificateRevocationList;binary`.

Syntax Binary, multi-valued.

Examples

```
certificateRevocationList;binary:: AAAAAA==
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name changeHasReplFixupOp – RetroChangelog attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.751
 NAME ’changeHasReplFixupOp’
 DESC ’RetroChangelog attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
 NO-USER-MODIFICATION
 USAGE directoryOperation
 X-ORIGIN ’Sun Directory Server’ )
```

Description This attribute is used for the retro change log.

This attribute is created in an entry when the following conditions are satisfied:

- The retro change log is enabled
- The `changeIsReplFixupOp` attribute is configured to TRUE
- Directory Server has performed an operation to resolve a replication conflict

This attribute stores the following information about an operation performed to resolve a replication conflict:

- Target DN of the operation
- The type of update
- The change made

There is one value of this attribute for each operation performed to resolve a replication conflict.

The value of this attribute is base64 encoded.

Syntax Binary, multi-valued.

Usage Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

The value of this attribute may only be modified by the server.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name changeIsReplFixupOp – RetroChangelog attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.726
 NAME 'changeIsReplFixupOp'
 DESC 'RetroChangelog attribute type'
 EQUALITY booleanMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
 SINGLE-VALUE
 NO-USER-MODIFICATION
 USAGE directoryOperation
 X-ORIGIN 'Sun Directory Server' )
```

Description This attribute is used for the retro change log.

This attribute indicates whether Directory Server has performed an operation to resolve a replication conflict. The value of this attribute is as follows:

TRUE A replication conflict has occured and Directory Server has performed an operation to resolve the conflict. The `changeHasReplFixupOp` attribute has been added to the entry to describe the operation.

FALSE Directory Server has not performed an operation to resolve a replication conflict on this entry.

Syntax Boolean, single-valued.

Usage Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

The value of this attribute may only be modified by the server.

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name changeLog – DN of the entry containing the set of entries comprising the server changelog

Synopsis

```
( 2.16.840.1.113730.3.1.35
 NAME ’changeLog’
 DESC ’DN of the entry containing the set of entries comprising the server changelog’
 EQUALITY distinguishedNameMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-ORIGIN ’Changelog Internet Draft’ )
```

Description The distinguished name of the entry that contains the set of entries comprising the server change log.

Syntax DN, multi-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Changelog Internet Draft

Name changeNumber – Changelog attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.5
 NAME 'changeNumber'
 DESC 'Changelog attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 X-ORIGIN 'Changelog Internet Draft' )
```

Description This single-valued attribute is always present. It contains an integer that uniquely identifies each change made to a directory entry. This number is related to the order in which the change occurred. The higher the number, the later the change.

Syntax Integer, multi-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Changelog Internet Draft

Name changes – Changelog attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.8
 NAME 'changes'
 DESC 'Changelog attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
 X-ORIGIN 'Changelog Internet Draft' )
```

Description For add and modify operations, contains the changes made to the entry, in LDIF format.

Syntax Binary, multi-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Changelog Internet Draft

Name changeTime – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.77
 NAME 'changeTime'
 DESC 'Sun ONE defined attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Defines a time, in a YYMMDDHHMMSS format, when the entry was added.

Syntax Directory String, multi-valued.

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name changeType – Changelog attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.7
 NAME ’changeType’
 DESC ’Changelog attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’Changelog Internet Draft’ )
```

Description Specifies the type of LDAP operation. This attribute can have one of the following values: add, delete, modify, or modRDN.

Syntax Directory String, multi-valued.

Examples

```
changeType: modify
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Changelog Internet Draft

Name cn, commonName – Standard LDAP attribute type

Synopsis

```
( 2.5.4.3
 NAME ( ’cn’ ’commonName’ )
 DESC ’Standard LDAP attribute type’
 SUP name
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’RFC 2256’ )
```

Description Identifies the name of an object in the directory. When the object corresponds to a person, the CN is typically the person's full name.

Syntax Directory String, multi-valued.

Examples When identifying the common name or full name of an entry:

```
commonName: Barbara Jensen
```

or

```
cn: Barbara Jensen
```

When in reference to LDAPReplica or LDAPServer object classes:

```
commonName: replicator.example.com:17430/dc%3Dexample%2Cdc%3Dcom
```

or

```
cn: replicator.example.com:17430/dc%3Dexample%2Cdc%3Dcom
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name co, friendlycountryname – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.43
 NAME ( ’co’ ’friendlycountryname’ )
 DESC ’Standard LDAP attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’RFC 1274’ )
```

Description Contains the name of a country. Often, the country attribute is used to describe a two-character code for a country, and the friendlyCountryName attribute is used to describe the actual country name.

Syntax Directory String, multi-valued.

Examples

```
friendlyCountryName: Ireland
```

or

```
co: Ireland
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name copiedFrom – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.613
 NAME 'copiedFrom'
 DESC 'Sun ONE defined attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 USAGE directoryOperation
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Used by read-only replica to recognize master data source. Contains a reference to the server that holds the master data. Note that this attribute is only used for legacy replication. It is not used for multi-master replication.

Syntax Directory String, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name copyingFrom – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.614
 NAME 'copyingFrom'
 DESC 'Sun ONE defined attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 USAGE directoryOperation
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Used by read-only replica to recognize master data source while replication is in progress. Contains a reference to the server that holds the master data. Note that this attribute is only used for legacy replication. It is not used for multi-master replication.

Syntax Directory String, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name cosAttribute – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.550
 NAME ’cosAttribute’
 DESC ’Sun ONE defined attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE ’internal’
 X-ORIGIN ’Sun ONE Directory Server’ )
```

Description Provides the name of the attribute for which you want to generate a value. You can specify more than one `cosAttribute` value. This attribute is used by all types of CoS definition entries.

The `cosAttribute` attribute allows two qualifiers following the name of the CoS attribute. The *override* qualifier has one of the following values:

- `default` (or no qualifier) - Indicates that the server does not override a real attribute value stored in the entry when it has the same type as the virtual attribute.
- `override` - Indicates that the server always returns the value generated by the CoS, even when there is a value stored with the entry.
- `operational` - Indicates that the attribute will only be returned if it is explicitly requested in the search. Operational attributes do not need to pass a schema check in order to be returned. It also has the same behavior as the override qualifier.

The merge qualifier is either absent or given with the following value:

- `merge-schemes` - Allows the virtual CoS attribute to be multivalued, either from multiple templates or multiple CoS definitions.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name cosIndirectSpecifier – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.577
 NAME 'cosIndirectSpecifier'
 DESC 'Sun ONE defined attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Specifies the attribute values used by an indirect CoS to identify the template entry.

Syntax Directory String, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name cosPriority – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.569
 NAME 'cosPriority'
 DESC 'Sun ONE defined attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Specifies which template provides the attribute value, when CoS templates compete to provide an attribute value. This attribute represents the global priority of a particular template. A priority of zero is the highest priority.

Syntax Integer, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name cosspecifier – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.551
 NAME ’cosspecifier’
 DESC ’Sun ONE defined attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-DS-USE ’internal’
 X-ORIGIN ’Sun ONE Directory Server’ )
```

Description Specifies the attribute value used by a classic CoS, which, along with the template entry's DN, identifies the template entry.

Syntax Directory String, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name costargettree – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.552
 NAME ’costargettree’
 DESC ’Sun ONE defined attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-DS-USE ’internal’
 X-ORIGIN ’Sun ONE Directory Server’ )
```

Description Determines the subtree of the DIT to which the CoS schema applies. The values for this attribute for the schema and for multiple CoS schema may overlap their target trees in an arbitrary fashion.

Syntax Directory String, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name costemplatedn – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.553
 NAME 'costemplatedn'
 DESC 'Sun ONE defined attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 SINGLE-VALUE
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Points to the entry that contains the CoS template.

Syntax DN, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name crossCertificatePair – Standard LDAP attribute type

Synopsis

```
( 2.5.4.40
 NAME 'crossCertificatePair'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
 X-ORIGIN 'RFC 2256' )
```

Description This attribute contains a pair of cross signed certificates. It is to be stored and requested in the binary form, as `crossCertificatePair;binary`.

Syntax Binary, multi-valued.

Examples

```
crossCertificatePair;binary:: AAAAAA==
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name dc, domaincomponent – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.25
 NAME ( ’dc’ ’domaincomponent’ )
 DESC ’Standard LDAP attribute type’
 EQUALITY caseIgnoreIA5Match
 SUBSTR caseIgnoreIA5SubstringsMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-ORIGIN ’RFC 2247’ )
```

Description Specifies one component of a domain name.

Syntax Directory String, single-valued.

Examples

```
domainComponent: example
```

or

```
dc: example
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2247

Name deletedEntryAttrs – RetroChangelog attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.595
 NAME 'deletedEntryAttrs'
 DESC 'RetroChangelog attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
 SINGLE-VALUE
 NO-USER-MODIFICATION
 USAGE directoryOperation
 X-ORIGIN 'Sun Directory Server' )
```

Description This attribute is used for the retro change log. When this attribute is configured and the retro change log enabled, the retro change log records the following information about an entry that has been deleted:

- Attributes specified in the value of `deletedEntryAttrs`
- Corresponding values of the attributes

The value of this attribute is base64 encoded.

Syntax Binary, single-valued.

Usage Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

The value of this attribute may only be modified by the server.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name deleteOldRdn – Changelog attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.10
 NAME 'deleteOldRdn'
 DESC 'Changelog attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
 X-ORIGIN 'Changelog Internet Draft' )
```

Description In the case of `modrdn` operations, specifies whether the old RDN was deleted.

Syntax Boolean, multi-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Changelog Internet Draft

Name deltaRevocationList – Standard LDAP attribute type

Synopsis

```
( 2.5.4.53
 NAME 'deltaRevocationList'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
 X-ORIGIN 'RFC 2256' )
```

Description This attribute contains the *delta revocation list*, a list of newly revoked certificates. It is stored and requested in the binary form, as `deltaRevocationList;binary`.

Syntax Binary, multi-valued.

Examples

```
deltaRevocationList;binary::AAAAAA==
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name departmentNumber – inetOrgPerson attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.2
 NAME ’departmentNumber’
 DESC ’inetOrgPerson attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’inetOrgPerson Internet Draft’ )
```

Description Identifies the entry's department number.

Syntax Directory String, multi-valued.

Examples

```
departmentNumber: 2604
```

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, RFC 2798

Name description – Free form text description of the replication agreement

Synopsis

```
( 2.5.4.13
 NAME 'description'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2256' )
```

Description This attribute allows you to initialize a replica. This attribute is absent by default, however, if you add this attribute with a value of `start`, the server reinitializes the replica and removes the attribute value.

Entry DN is cn=ReplicationAgreementName,cn=replica,cn="suffixName", cn=mapping tree,cn=config.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples

```
description: Replication Agreement between Server A and Server B.
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name destinationIndicator – Standard LDAP attribute type

Synopsis

```
( 2.5.4.27
 NAME ’destinationIndicator’
 DESC ’Standard LDAP attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’RFC 2256’ )
```

Description The country and city associated with the entry needed to provide Public Telegram Service. Generally used in conjunction with `registeredAddress`.

Syntax Directory String, multi-valued.

Examples

```
destinationIndicator: Stow, Ohio, USA
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name

displayName – inetOrgPerson attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.241
 NAME ’displayName’
 DESC ’inetOrgPerson attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-ORIGIN ’inetOrgPerson Internet Draft’ )
```

Description

Preferred name of a person to be used when displaying entries. Especially useful in displaying a preferred name for an entry within a one-line summary list. Since other attribute types, such as cn, are multi-valued, they cannot be used to display a preferred name.

Syntax

Directory String, single-valued.

Examples

```
displayName: Michigan Smith
```

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, RFC 2798

Name distinguishedName, dn – Standard LDAP attribute type

Synopsis

```
( 2.5.4.49
 NAME ( 'dn' 'distinguishedName' )
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-ORIGIN 'RFC 2256' )
```

Description Defines the distinguished name (`dn`) for the entry. Note that the `dn` is not always a mandatory attribute in an entry.

Syntax DN, multi-valued.

Examples

```
dn: cn=Jane Doe, ou=Quality Control, dc=example, dc=com
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name dITContentRules – Standard LDAP attribute type

Synopsis

```
( 2.5.21.2
 NAME 'dITContentRules'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2252' )
```

Description Multi-valued attribute that defines the DIT content rules in force within a subschema. Each value defines one DIT content rule. Each value is tagged by the object identifier of the structural object class to which it pertains.

Note that Directory Server does not support or use this attribute.

Syntax Directory String, multi-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2252

Name ditRedirect – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.54
 NAME 'ditRedirect'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-ORIGIN 'RFC 1274' )
```

Description Used to indicate that the object described by one entry now has a newer entry in the directory tree. This attribute may be used when an individual's place of work changes, and the individual acquires a new organizational DN.

Syntax DN, multi-valued.

Examples

```
ditRedirect: cn=jdoe, dc=example, dc=com
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name

dITStructureRules – Standard LDAP attribute type

Synopsis

```
( 2.5.21.1
 NAME 'dITStructureRules'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2252' )
```

Description

Multi-valued attribute that defines the DIT structure rules in force within a subschema. Each value defines one DIT structure rule.

Note that Directory Server does not support or use this attribute.

Syntax

Directory String, multi-valued.

Attributes

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2252

Name dmdName – LDAP attribute type

Synopsis

```
( 2.5.4.54
 NAME 'dmdName'
 SUP name
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2256' )
```

Description The value of this attribute specifies a directory management domain (DMD), the administrative authority that operates Directory Server.

Syntax Directory String, multi-valued.

Examples `dmdName: example.com`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name dNSRecord – Pilot attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.26
 NAME 'dNSRecord'
 DESC 'Pilot attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 X-ORIGIN 'Internet directory pilot' )
```

Description Specifies DNS resource records, including type A (Address), type MX (Mail Exchange), type NS (Name Server), and type SOA (Start Of Authority) resource records.

Syntax IA5 String, multi-valued.

Examples

```
dNSRecord: IN NS ns.uu.net
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Internet directory pilot

Name documentAuthor – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.14
 NAME ’documentAuthor’
 DESC ’Standard LDAP attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-ORIGIN ’RFC 1274’ )
```

Description Contains the distinguished name of the author of a document entry.

Syntax DN, multi-valued.

Examples

```
documentAuthor: cn=John Doe, dc=example, dc=com
```

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name documentIdentifier – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.11
 NAME 'documentIdentifier'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 1274' )
```

Description Specifies a unique identifier for a document.

Syntax Directory String, multi-valued.

Examples

```
documentIdentifier: L3204REV1
```

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name documentLocation – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.15
 NAME 'documentLocation'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 1274' )
```

Description Defines the location of the original copy of a document entry.

Syntax Directory String, multi-valued.

Examples `documentLocation: Department Library`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name

documentPublisher – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.56
 NAME 'documentPublisher'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-ORIGIN 'RFC 1274' )
```

Description

The person and/or organization that published a document.

Syntax

Directory String, single-valued.

Examples

```
documentPublisher: Southeastern Publishing
```

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name documentStore – Pilot attribute type

Synopsis

```
( 0.9.2342.19200300.102.1.10
 NAME ’documentStore’
 DESC ’Pilot attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’Internet White Pages Pilot’ )
```

Description Defines the place in which a document is stored.

Syntax Directory String, multi-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Internet White Pages Pilot

Name documentTitle – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.12
 NAME 'documentTitle'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 1274' )
```

Description Contains the title of a document entry.

Syntax Directory String, multi-valued.

Examples

```
documentTitle: Directory Server Administration Guide
```

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name documentVersion – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.13
 NAME 'documentVersion'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 1274' )
```

Description Defines the version of a document entry.

Syntax Directory String, multi-valued.

Examples

```
documentVersion: 1.1
```

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name drink, favouriteDrink – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.5
 NAME ( 'drink' 'favouriteDrink' )
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 1274' )
```

Description Describes the favorite drink of a person entry.

Syntax Directory String, multi-valued.

Examples

```
drink: gin
```

or

```
favouriteDrink: gin
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name dSAQuality – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.49
 NAME 'dSAQuality'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-ORIGIN 'RFC 1274' )
```

Description Specifies the purported quality of a DSA. This attribute allows a DSA manager to indicate the expected level of availability of the DSA.

Syntax Directory String, single-valued.

Examples

```
dSAQuality: high
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name ds-pluginDigest – Sun reserved definition

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.57
 NAME ’ds-pluginDigest’
 DESC ’Sun reserved definition’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 USAGE directoryOperation
 X-DS-USE ’internal’
 X-ORIGIN ’Sun ONE Directory Server’ )
```

Description The configuration digest of a signed plug-in. (The plug-in entry DN, ID, version, type, init function, and vendor are hashed together to create the configuration digest.)

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name

ds-pluginSignature – Sun reserved definition

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.7
 NAME ’ds-pluginSignature’
 DESC ’Sun reserved definition’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 USAGE directoryOperation
 X-DS-USE ’internal’
 X-ORIGIN ’Sun ONE Directory Server’ )
```

Description

The configuration signature of a signed plug-in.

Syntax

Directory String, multi-valued.

Usage

Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name

dsSaslMaxBufSize – Sun DS attribute for SASL config

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.791
 NAME 'dsSaslMaxBufSize'
 DESC 'Sun DS attribute for SASL config'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 X-DS-USE 'internal'
 X-ORIGIN 'Sun Directory Server' )
```

Description

Reflects the maximum buffer size for SASL, which limits the size of packets accepted from SASL clients. The default value is 65535, and is not modifiable. The actual value used is negotiated to be less than or equal to this maximum.

Syntax

Integer, single-valued.

Usage

Attribute specific to this Directory Server instance and version of the schema.

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name dsSaslMaxSSF – Sun DS attribute for SASL config

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.790
 NAME 'dsSaslMaxSSF'
 DESC 'Sun DS attribute for SASL config'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 X-DS-USE 'internal'
 X-ORIGIN 'Sun Directory Server' )
```

Description Reflects the maximum security strength factor value for SASL. The default value is `32767`. The value of this attribute must be greater than or equal to the value of `dsSaslMinSSF`.

Some security strength factor values include:

`0`	No protection
`1`	Integrity protection only
`40`	40-bit DES or 40-bit RC2/RC4
`56`	DES or other weak ciphers
`112`	triple DES and other strong ciphers
`128`	128-bit RC2/RC4/Blowfish and other modern strong ciphers
`256`	baseline AES

Syntax Integer, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples `dsSaslMaxSSF: 256`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name dsSaslMinSSF – Sun DS attribute for SASL config

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.789
 NAME 'dsSaslMinSSF'
 DESC 'Sun DS attribute for SASL config'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 X-DS-USE 'internal'
 X-ORIGIN 'Sun Directory Server' )
```

Description Reflects the minimum security strength factor value for SASL. The default value is `0`. The value of this attribute must be less than or equal to the value of `dsSaslMaxSSF`.

Some security strength factor values include:

`0`	No protection
`1`	Integrity protection only
`40`	40-bit DES or 40-bit RC2/RC4
`56`	DES or other weak ciphers
`112`	triple DES and other strong ciphers
`128`	128-bit RC2/RC4/Blowfish and other modern strong ciphers
`256`	baseline AES

Syntax Integer, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples

```
dsSaslMinSSF: 56
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name dsSaslPluginsEnable – Sun ONE defined attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.467
 NAME ’dsSaslPluginsEnable’
 DESC ’Sun ONE defined attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE ’internal’
 X-ORIGIN ’Sun ONE Directory Server’ )
```

Description Specifies the SASL mechanisms of the plug-ins to enable. Possible values are those of the `supportedSASLMechanisms` attribute on the root DSE, including `EXTERNAL`, `DIGEST-MD5`, and `GSSAPI`.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name dsSaslPluginsPath – Sun ONE defined attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.466
 NAME 'dsSaslPluginsPath'
 DESC 'Sun ONE defined attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Specifies the path to the system directory containing libraries implementing the required SASL mechanisms.

Syntax Directory String, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples

```
dsSaslPluginsPath: /lib/sasl
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name employeeNumber – inetOrgPerson attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.3
 NAME 'employeeNumber'
 DESC 'inetOrgPerson attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-ORIGIN 'inetOrgPerson Internet Draft' )
```

Description Identifies the entry's employee number.

Syntax Directory String, single-valued.

Examples `employeeNumber: 3440`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, RFC 2798

Name employeeType – inetOrgPerson attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.4
 NAME 'employeeType'
 DESC 'inetOrgPerson attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'inetOrgPerson Internet Draft' )
```

Description Identifies the entry's type of employment.

Syntax Directory String, multi-valued.

Examples

```
employeeType: Full time
```

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, RFC 2798

Name enhancedSearchGuide – Standard LDAP attribute type

Synopsis

```
( 2.5.4.47
 NAME 'enhancedSearchGuide'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2256' )
```

Description Used by X.500 clients when constructing search filters.

Syntax Directory String, multi-valued.

Examples

```
enhancedSearchGuide: (uid=mhughes)
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name facsimileTelephoneNumber, fax – Standard LDAP attribute type

Synopsis

```
( 2.5.4.23
 NAME ( 'facsimileTelephoneNumber' 'fax' )
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.50
 X-ORIGIN 'RFC 2256' )
```

Description Identifies the fax number at which the entry can be reached. Abbreviation: fax.

Syntax Telephone Number, multi-valued.

Examples

```
facsimileTelephoneNumber: 415-555-1212
```

or:

```
fax: 415-555-1212
```

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name gecos – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.2
 NAME 'gecos'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-ORIGIN 'RFC 2307' )
```

Description The default GECOS.

Syntax Directory String, single-valued.

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name generationQualifier – Standard LDAP attribute type

Synopsis

```
( 2.5.4.44
 NAME ’generationQualifier’
 DESC ’Standard LDAP attribute type’
 SUP name
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’RFC 2256’ )
```

Description Contains the generation Qualifier part of the name, typically appearing in the suffix.

Syntax Directory String, multi-valued.

Examples

```
generationQualifier: Jr
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name gidNumber – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.1
 NAME 'gidNumber'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 X-ORIGIN 'RFC 2307' )
```

Description Group ID number.

Syntax Integer, single-valued.

Examples `gidNumber: 162035`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name givenName – Standard LDAP attribute type

Synopsis

```
( 2.5.4.42
 NAME 'givenName'
 DESC 'Standard LDAP attribute type'
 SUP name
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2256' )
```

Description Identifies the entry's given name, usually a person's first name.

Syntax Directory String, multi-valued.

Examples

```
givenName: Hecuba
```

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name homeDirectory – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.3
 NAME 'homeDirectory'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 SINGLE-VALUE
 X-ORIGIN 'RFC 2307' )
```

Description The home directory of the account.

Syntax IA5 String, single-valued.

Examples

```
homeDirectory: /home/bsmith
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name homePhone – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.20
 NAME ’homePhone’
 DESC ’Standard LDAP attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.50
 X-ORIGIN ’RFC 1274’ )
```

Description Identifies the entry's home phone number.

Syntax Telephone Number, multi-valued.

Examples

```
homePhone: 415-555-1212
```

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name homePostalAddress – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.39
 NAME 'homePostalAddress'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 1274' )
```

Description Identifies the home mailing address of an entry. This field is intended to include multiple lines, but each line within the entry should be separated by a dollar sign ($). To represent an actual dollar sign ($) or backslash (\) within this text, use the escaped hex values \24 and \5c respectively.

Syntax Directory String, multi-valued.

Examples To identify the home mailing address:

```
homePostalAddress: 1234 Ridgeway Drive$Santa Clara, CA$99555
```

Additionally, to represent the string:

```
The dollar ($) value can be found
in the c:\cost file.
```

provide the string:

```
The dollar (\24) value can be found$in the c:\5ccost file.
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name host – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.9
 NAME 'host'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 1274' )
```

Description Defines the hostname of a computer.

Syntax Directory String, multi-valued.

Examples

```
host: myServer
```

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name houseIdentifier – Standard LDAP attribute type

Synopsis

```
( 2.5.4.51
 NAME 'houseIdentifier'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2256' )
```

Description Identifies a building in a location.

Syntax Directory String, multi-valued.

Examples

```
houseIdentifier: B105
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name info – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.4
 NAME 'info'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 1274' )
```

Description Specifies any general information pertinent to an object. It is recommended that specific usage of this attribute type is avoided, and that specific requirements are met by other (possibly additional) attribute types.

Syntax Directory String, multi-valued.

Examples

```
info: not valid
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name

initials – Standard LDAP attribute type

Synopsis

```
( 2.5.4.43
 NAME ’initials’
 DESC ’Standard LDAP attribute type’
 SUP name
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’RFC 2256’ )
```

Description

Identifies the entry's initials. Does not identify the entry's surname.

Syntax

Directory String, multi-valued.

Examples

```
initials: BFA
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name internationaliSDNNumber – Standard LDAP attribute type

Synopsis

```
( 2.5.4.25
 NAME 'internationaliSDNNumber'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 X-ORIGIN 'RFC 2256' )
```

Description Contains the ISDN number of the entry. This is in the internationally agreed format for ISDN addresses given in CCITT Rec. E. 164.

Syntax IA5 String, multi-valued.

Examples

```
internationaliSDNNumber: +SO 812467
```

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name ipHostNumber – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.19
 NAME 'ipHostNumber'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2307' )
```

Description IP address, expressed as a dotted decimal, omitting leading zeros.

Syntax Directory String, multi-valued.

Examples

```
ipHostNumber: 10.0.0.1
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name ipNetmaskNumber – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.21
 NAME 'ipNetmaskNumber'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-ORIGIN 'RFC 2307' )
```

Description IP netmask, expressed as a dotted decimal, omitting leading zeros.

Syntax Directory String, single-valued.

Examples `ipNetmaskNumber: 255.255.255.0`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name ipNetworkNumber – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.20
 NAME 'ipNetworkNumber'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-ORIGIN 'RFC 2307' )
```

Description IP network, expressed as a dotted decimal, omitting leading zeros.

Syntax Directory String, single-valued.

Examples

```
ipNetworkNumber: 192.168
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name ipProtocolNumber – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.17
 NAME 'ipProtocolNumber'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 X-ORIGIN 'RFC 2307' )
```

Description The IP protocol number.

Syntax Integer, single-valued.

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name ipServicePort – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.15
 NAME 'ipServicePort'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 X-ORIGIN 'RFC 2307' )
```

Description The IP service port number.

Syntax Integer, single-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name ipServiceProtocol – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.16
 NAME ’ipServiceProtocol’
 DESC ’Standard LDAP attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’RFC 2307’ )
```

Description The IP service protocol.

Syntax Directory String, multi-valued.

Examples

```
ipServiceProtocol: tcp
ipServiceProtocol: udp
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name isMemberOf – LDAP attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.792
 NAME 'isMemberOf'
 DESC 'Sun defined attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 NO-USER-MODIFICATION
 USAGE directoryOperation
 X-DS-USE 'internal'
 X-ORIGIN 'Sun Directory Server' )
```

Description The values of this attribute are the DNs of static groups to which this entry belongs.

This attribute values are calculated, thus cannot be indexed. Therefore this attribute should not be used in search filters.

Syntax DN, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

The value of this attribute may only be modified by the server.

Examples

```
isMemberOf: cn=bigGroup,ou=groups,dc=example,dc=com
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name janetMailbox – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.46
 NAME 'janetMailbox'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 1274' )
```

Description Specifies an email address. This attribute is intended for the convenience of UK users unfamiliar with rfc822 mail addresses. Entries using this attribute must also include an `rfc822Mailbox` attribute.

Syntax Directory String, multi-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name javaClassName – Fully qualified name of distinguished Java class or interface

Synopsis

```
( 1.3.6.1.4.1.42.2.27.4.1.6
 NAME ’javaClassName’
 DESC ’Fully qualified name of distinguished Java class or interface’
 EQUALITY caseExactMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-ORIGIN ’RFC 2713’ )
```

Description Stores the fully qualified name of the Java object's distinguished class or interface.

Syntax Directory String, single-valued.

Examples

```
javaClassName: java.lang.String
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2713

Name javaClassNames – Fully qualified Java class or interface name

Synopsis

```
( 1.3.6.1.4.1.42.2.27.4.1.13
 NAME 'javaClassNames'
 DESC 'Fully qualified Java class or interface name'
 EQUALITY caseExactMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2713' )
```

Description Stores the Java object's fully qualified class or interface names. It is a multivalued attribute. When more than one value is present, each is the name of a class or interface, or ancestor class or interface, of this object.

Syntax Directory String, multi-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2713

Name javaCodebase – URL(s) specifying the location of class definition

Synopsis

```
( 1.3.6.1.4.1.42.2.27.4.1.7
 NAME 'javaCodebase'
 DESC 'URL(s) specifying the location of class definition'
 EQUALITY caseExactIA5Match
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 X-ORIGIN 'RFC 2713' )
```

Description Stores the Java class definition's locations. It specifies the locations from which to load the class definition for the class specified by the `javaClassName` attribute. If this attribute contains more than one value, each value is an independent codebase.

Syntax IA5 String, multi-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2713

Name

javaDoc – The Java documentation for the class

Synopsis

```
( 1.3.6.1.4.1.42.2.27.4.1.12
 NAME 'javaDoc'
 DESC 'The Java documentation for the class'
 EQUALITY caseExactIA5Match
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 X-ORIGIN 'RFC 2713' )
```

Description

This attribute stores a pointer to the Java documentation for the class. Its value is a URL.

Syntax

IA5 String, multi-valued.

Examples

```
javaDoc: http://java.sun.com/products/j2se/1.5.0/docs/api/java/lang/String.html
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2713

Name javaFactory – Fully qualified Java class name of a JNDI object factory

Synopsis

```
( 1.3.6.1.4.1.42.2.27.4.1.10
 NAME ’javaFactory’
 DESC ’Fully qualified Java class name of a JNDI object factory’
 EQUALITY caseExactMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-ORIGIN ’RFC 2713’ )
```

Description Stores the fully qualified class name of the object factory that can be used to create an instance of the object identified by the `javaClassName` attribute.

Syntax Directory String, single-valued.

Examples

```
javaFactory: com.sun.jndi.ExampleObjectFactory
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2713

Name javaReferenceAddress – Addresses associated with a JNDI Reference

Synopsis

```
( 1.3.6.1.4.1.42.2.27.4.1.11
 NAME 'javaReferenceAddress'
 DESC 'Addresses associated with a JNDI Reference'
 EQUALITY caseExactMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2713' )
```

Description Represents the sequence of addresses of a JNDI reference. Each of its values represents one address, a Java object of type `javax.naming.RefAddr`. Its value is a concatenation of the address type and address contents, preceded by a sequence number.

Syntax Directory String, multi-valued.

Examples

```
ipServiceProtocol: #0#TypeA#ValA
                  #1#TypeB#ValB
                  #2#TypeC##rO0ABXNyABpq
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2713

Name javaSerializedData – Serialized form of a Java object

Synopsis

```
( 1.3.6.1.4.1.42.2.27.4.1.8
 NAME ’javaSerializedData’
 DESC ’Serialized form of a Java object’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.40
 SINGLE-VALUE
 X-ORIGIN ’RFC 2713’ )
```

Description Stores the serialized form of a Java object.

Syntax Octet String, single-valued.

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2713

Name jpegPhoto – inetOrgPerson attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.60
 NAME 'jpegPhoto'
 DESC 'inetOrgPerson attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
 X-ORIGIN 'inetOrgPerson Internet Draft (XXX: syntax should be ...28)' )
```

Description Contains a JPEG photo of the entry.

The syntax for this attribute differs from the standard syntax, which should end with `.28`, meaning JPEG syntax.

Syntax Binary, multi-valued.

Examples

```
jpegPhoto:: AAAAAA==
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, RFC 2798

Name keyWords – Pilot attribute type

Synopsis

```
( 0.9.2342.19200300.102.1.7
 NAME 'keyWords'
 DESC 'Pilot attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'Internet White Pages Pilot' )
```

Description Contains keywords for the entry.

Syntax Directory String, multi-valued.

Examples `keyWords: directory LDAP X.500`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Internet White Pages Pilot

Name knowledgeInformation – Standard LDAP attribute type

Synopsis

```
( 2.5.4.2
 NAME 'knowledgeInformation'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2256' )
```

Description This attribute is no longer used.

Syntax Directory String, multi-valued.

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name l, locality, localityname – Standard LDAP attribute type

Synopsis

```
( 2.5.4.7
 NAME ( ’l’ ’locality’ ’localityname’ )
 DESC ’Standard LDAP attribute type’
 SUP name
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’RFC 2256’ )
```

Description Identifies the county, city, or other geographical area in which the entry is located or with which it is in some other way associated.

Syntax Directory String, multi-valued.

Examples

```
localityName: Santa Clara
```

or

```
l: Santa Clara
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name labeledUri, labeledurl – Uniform Resource Identifier with optional label

Synopsis

```
( 1.3.6.1.4.1.250.1.57
 NAME ( 'labeledUri' 'labeledurl' )
 DESC 'Uniform Resource Identifier with optional label'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 X-ORIGIN 'RFC 2079' )
```

Description Specifies a Uniform Resource Identifier (URI) that is relevant in some way to the entry. Values placed in the attribute should consist of a URI (currently only URLs are supported) optionally followed by one or more space characters and a label.

Syntax IA5 String, multi-valued.

Examples

```
labeledURI: http://www.sun.com
```

```
labeledURI: http://www.sun.com                        Sun website
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2079

Name lastModifiedBy – old variant of modifiersName

Synopsis

```
( 0.9.2342.19200300.100.1.24
 NAME ’lastModifiedBy’
 DESC ’old variant of modifiersName’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-ORIGIN ’RFC 1274’ )
```

Description Specifies the distinguished name of the last user to modify the associated entry.

Syntax DN, multi-valued.

Examples

```
lastModifiedBy: cn=Jane Doe,ou=Quality Control,dc=example,dc=com
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name lastModifiedTime – old variant of modifyTimestamp

Synopsis

```
( 0.9.2342.19200300.100.1.23
 NAME 'lastModifiedTime'
 DESC 'old variant of modifyTimestamp'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 1274' )
```

Description Defines the last time, in UTC format, that a change was made to the entry.

Syntax Directory String, multi-valued.

Examples

```
lastModifiedTime: Thu Sep 21 17:23:09 MEST 2006
```

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name ldapSyntaxes – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.4.1.1466.101.120.16
 NAME 'ldapSyntaxes'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 USAGE directoryOperation
 X-ORIGIN 'RFC 2252' )
```

Description This attribute identifies the syntaxes implemented, with each value corresponding to one syntax.

Syntax Directory String, multi-valued.

Usage Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2252

Name loginShell – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.4
 NAME 'loginShell'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 SINGLE-VALUE
 X-ORIGIN 'RFC 2307' )
```

Description The path to the login shell.

Syntax IA5 String, single-valued.

Examples `loginShell: /bin/bash`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name macAddress – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.22
 NAME 'macAddress'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2307' )
```

Description The MAC address in maximal, colon separated hex notation.

Syntax Directory String, multi-valued.

Examples

```
macAddress: 8:0:20:c0:5c:96
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name

mail, rfc822mailbox – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.3
 NAME ( 'mail' 'rfc822mailbox' )
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 1274' )
```

Description

Identifies a user's primary email address (the email address retrieved and displayed by white pages lookup applications).

Syntax

Directory String, multi-valued.

Examples

```
mail: banderson@example.com
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name mailPreferenceOption – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.47
 NAME 'mailPreferenceOption'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 X-ORIGIN 'RFC 1274' )
```

Description Indicates a preference for the inclusion of user names on mailing lists (electronic or physical). Accepted values include:

- 0: user does not want to be included in mailing lists.
- 1: user consents to be added to any mailing list.
- 2: user only wants to be added to mailing lists that the list provider views as relevant to the user's professional interests.

The absence of this attribute for a person should be interpreted as if the attribute were present with the value `no-list-inclusion`. This attribute should be interpreted by anyone using the directory to derive mailing lists, and its value respected.

Syntax Integer, single-valued.

Examples `mailPreferenceOption:0`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name manager – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.10
 NAME 'manager'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-ORIGIN 'RFC 1274' )
```

Description Identifies the distinguished name of the entry's manager.

Syntax DN, multi-valued.

Examples `manager:cn=Jane Doe, ou=Quality Control, dc=example, dc=com`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name matchingRules – Standard LDAP attribute type

Synopsis

```
( 2.5.21.4
 NAME 'matchingRules'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2252' )
```

Description Multi-valued attribute that defines the matching rules used within a subschema. Each value defines one matching rule.

Syntax Directory String, multi-valued.

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2252

Name matchingRuleUse – Standard LDAP attribute type

Synopsis

```
( 2.5.21.8
 NAME 'matchingRuleUse'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2252' )
```

Description Used to indicate the attribute types to which a matching rule applies in a subschema.

Syntax Directory String, multi-valued.

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2252

Name member – Standard LDAP attribute type

Synopsis

```
( 2.5.4.31
 NAME 'member'
 DESC 'Standard LDAP attribute type'
 SUP distinguishedName
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-ORIGIN 'RFC 2256' )
```

Description Identifies the distinguished names for each member of the group.

Syntax DN, multi-valued.

Examples

```
member: cn=John Doe, dc=example, dc=com
```

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name memberCertificateDescription – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.199
 NAME 'memberCertificateDescription'
 DESC 'Sun ONE defined attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description A multi-valued attribute, for which each value is a description, a pattern, or a filter matching the subject DN of a certificate (usually certificates used for SSL client authentication).

memberCertificateDescription matches any certificate that contains a subject DN with the same AVAs as the description. The description may contain multiple `ou=` AVAs. A matching DN must contain those same `ou=` AVAs, in the same order, although it may contain other AVAs (including other `ou=` AVAs) interspersed. For any other attribute type (not `ou`), there should be at most one AVA of that type in the description. If there are several, all but the last are ignored.

A matching DN must contain that same AVA, but no other AVA of the same type nearer the root (later, syntactically).

AVAs are considered the same if they contain the same attribute description (case-insensitive comparison) and the same attribute value (case-insensitive comparison, leading and trailing whitespace ignored, and consecutive whitespace characters treated as a single SP).

In order to be considered a member of a group with the following memberCertificateDescription, a certificate would need to include `ou=x`, `ou=A`, and `o=example`, but not `o=company`.

Syntax IA5 String, multi-valued.

Examples

```
memberCertificateDescription: {ou=x, ou=A, o=company, o=example}
```

In order to match the group's requirements, a certificate's subject DNs must contain the same `ou` attribute types in the same order as defined in the memberCertificateDescription attribute.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name memberNisNetgroup – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.13
 NAME ’memberNisNetgroup’
 DESC ’Standard LDAP attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 X-ORIGIN ’RFC 2307’ )
```

Description The name of a netgroup.

Syntax IA5 String, multi-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name memberUid – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.12
 NAME 'memberUid'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 X-ORIGIN 'RFC 2307' )
```

Description The user id of the member.

Syntax IA5 String, multi-valued.

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name memberURL – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.198
 NAME 'memberURL'
 DESC 'Sun ONE defined attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Identifies a URL associated with each member of a group. Any type of labeled URL can be used.

Syntax IA5 String, multi-valued.

Examples

```
memberURL: ldap:///cn=jdoe,dc=example,dc=com
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name mobile, mobileTelephoneNumber – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.41
 NAME ( 'mobile' 'mobileTelephoneNumber' )
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.50
 X-ORIGIN 'RFC 1274' )
```

Description Identifies the entry's mobile or cellular phone number. Abbreviation: mobile.

Syntax Telephone Number, multi-valued.

Examples

```
mobileTelephoneNumber: 415-555-4321
```

```
mobile: 415-555-4321
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name multiLineDescription – Pilot attribute type

Synopsis

```
( 1.3.6.1.4.1.250.1.2
 NAME ’multiLineDescription’
 DESC ’Pilot attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’Internet White Pages Pilot’ )
```

Description Provides descriptive text for a mail user. When represented in LDIF format, each line should be separated by a dollar sign ($). Directory Server expects 0 or 1 occurrences of this attribute per mail account.

Syntax Directory String, multi-valued.

Examples

```
multiLineDescription: Account Administrator and$directory manager.
```

To represent an actual dollar sign ($) or backslash (\) within this text, use the escaped hex values \24 and \5c respectively. For example, to represent the string:

```
The dollar ($) value can be found
in the c:\cost file.
```

provide the string:

```
The dollar (\24) value can be found$in the c:\5ccost file.
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Internet White Pages Pilot

Name

name – LDAP attribute type

Synopsis

```
( 2.5.4.41
 NAME 'name'
 EQUALITY caseIgnoreMatch
 SUBSTR caseIgnoreSubstringsMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32768}
 X-ORIGIN 'RFC 2256' )
```

Description

Identifies the attribute supertype from which string attribute types used for naming may be formed. It is unlikely that values of this type will occur in an entry. LDAP server implementations that do not support attribute subtyping do not need to recognize this attribute in requests. Client implementations should not assume that LDAP servers are capable of performing attribute subtyping.

Syntax

Directory String, multi-valued.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name nameForms – Standard LDAP attribute type

Synopsis

```
( 2.5.21.7
 NAME ’nameForms’
 DESC ’Standard LDAP attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’RFC 2252’ )
```

Description Multi-valued attribute that defines the name forms used in a subschema. Each value defines one name form.

Note that Directory Server does not support or use this attribute.

Syntax Directory String, multi-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2252

Name namingContexts – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.4.1.1466.101.120.5
 NAME 'namingContexts'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 USAGE dsaOperation
 X-ORIGIN 'RFC 2252' )
```

Description Corresponds to a naming context the server is mastering or shadowing. When Directory Server does not master any information (for example, it is an LDAP gateway to a public X.500 directory), this attribute is absent. When Directory Server believes it contains the entire directory, the attribute has a single value, and that value is the empty string (indicating the null DN of the root).This attribute permits a client contacting a server to choose suitable base objects for searching.

Syntax DN, multi-valued.

Usage Operational attribute used by a Directory Server instance; returned in `ldapsearch` only when specifically requested.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2252

Name newRdn – Changelog attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.9
 NAME 'newRdn'
 DESC 'Changelog attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-ORIGIN 'Changelog Internet Draft' )
```

Description In the case of `modrdn` operations, specifies the new RDN of the entry.

Syntax DN, multi-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Changelog Internet Draft

Name newSuperior – Changelog attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.11
 NAME 'newSuperior'
 DESC 'Changelog attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-ORIGIN 'Changelog Internet Draft' )
```

Description In the case of `modrdn` operations, specifies the `newSuperior` attribute of the entry.

Syntax DN, multi-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Changelog Internet Draft

Name nisMapEntry – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.27
 NAME 'nisMapEntry'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 SINGLE-VALUE
 X-ORIGIN 'RFC 2307' )
```

Description The NIS map entry ID.

Syntax IA5 String, single-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name nisMapName – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.26
 NAME 'nisMapName'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2307' )
```

Description The name of the NIS map.

Syntax Directory String, multi-valued.

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name nisNetgroupTriple – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.14
 NAME ’nisNetgroupTriple’
 DESC ’Standard LDAP attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 X-ORIGIN ’RFC 2307’ )
```

Description Defines a NIS netgroup with the syntax *hostname*, *username*, *domainname*.

Syntax IA5 String, multi-valued.

Examples

```
nisNetgroupTriple: (myserver,jsmith,example.com)
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name nsds5ReplConflict – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.973
 NAME 'nsds5ReplConflict'
 DESC 'Sun ONE defined attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 USAGE directoryOperation
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description This attribute is a conflict marker attribute. It is included on entries that have a change conflict that cannot be resolved automatically by the replication process.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name nsIdleTimeout – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.573
NAME 'nsIdleTimeout'
DESC 'Binder-based connection idle timeout (seconds)'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation
X-DS-USE 'internal'
X-ORIGIN 'Sun ONE Directory Server' )
```

Description This attribute specifies the maximum time a client connection can remain idle before the connection is dropped.

Syntax Integer, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema. Operational attribute used by the directory service.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name

nsLicensedFor – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.36
 NAME 'nsLicensedFor'
 DESC 'Sun ONE defined attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'Sun ONE Administration Services' )
```

Description

Identifies the server the user is licensed to use. The Administration Server expects each `nsLicenseUser` entry to contain zero or more instances of this attribute. Valid keywords for this attribute are currently:

- mail: the user is a licensed client of the Messaging Server.
- new: the user is a licensed client of the Collabra Server.
- slapd: the user is a licensed client of Directory Server.
- cal: the user is a licensed client of the Calendar Server.

Syntax

Directory String, multi-valued.

Examples

```
nsLicensedFor: slapd
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name nsLicenseEndTime – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.38
 NAME 'nsLicenseEndTime'
 DESC 'Sun ONE defined attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'Sun ONE Administration Services' )
```

Description Reserved for future use.

Syntax Directory String, multi-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name nsLicenseStartTime – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.37
 NAME 'nsLicenseStartTime'
 DESC 'Sun ONE defined attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'Sun ONE Administration Services' )
```

Description Reserved for future use.

Syntax Directory String, multi-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name nsLookThroughLimit – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.570
NAME 'nsLookThroughLimit'
DESC 'Binder-based search operation look through limit (candidate entries)'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation
X-DS-USE 'internal'
X-ORIGIN 'Sun ONE Directory Server' )
```

Description This attribute specifies the maximum number of entries examined for a search operation.

Syntax Integer, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema. Operational attribute used by the directory service.

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name nsRole – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.574
 NAME 'nsRole'
 DESC 'Sun ONE defined attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 NO-USER-MODIFICATION
 USAGE directoryOperation
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description This attribute is a computed attribute that is not stored with the entry itself. It identifies which roles an entry belongs to.

Syntax DN, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

The value of this attribute may only be modified by the server.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name nsRoleDN – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.575
 NAME ’nsRoleDN’
 DESC ’Sun ONE defined attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 USAGE directoryOperation
 X-DS-USE ’internal’
 X-ORIGIN ’Sun ONE Directory Server’ )
```

Description This attribute contains the distinguished name of each managed role to which the entry belongs. Membership of a managed role is conferred upon an entry by adding the role's DN to the entry's `nsRoleDN` attribute.

This attribute is not to be confused with the generated `nsRole` attribute that contains the DN of *all* roles to which the entry belongs, as computed by Directory Server. Use `nsRoleDN` to set managed role membership, and use `nsRole` to evaluate role membership.

Syntax DN, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

Examples

```
dn: cn=staff,ou=People,dc=example,dc=com
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsSimpleRoleDefinition
objectclass: nsManagedRoleDefinition

dn: uid=bjensen,ou=People,dc=example,dc=com
objectclass: top
objectclass: person
sn: Jensen
cn: Babs Jensen
uid: bjensen
nsroledn: cn=staff,ou=People,dc=example,dc=com
```

A nested role specifies containment of one or more roles of any type. In that case, nsRoleDN defines the DN of the contained roles.

```
dn: cn=everybody,o=example.com
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsNestedRoleDefinition
nsroledn: cn=manager,ou=People,dc=example,dc=com
nsroledn: cn=staff,ou=People,dc=example,dc=com
```

Attributes

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name nsRoleFilter – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.576
 NAME ’nsRoleFilter’
 DESC ’Sun ONE defined attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 SINGLE-VALUE
 X-DS-USE ’internal’
 X-ORIGIN ’Sun ONE Directory Server’ )
```

Description Specifies a search filter to select entries having the role.

Syntax IA5 String, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name

nsRoleScopeDn – Sun ONE defined attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.10
 NAME 'nsRoleScopeDn'
 DESC 'Sun ONE defined attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 SINGLE-VALUE
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description

Determines the scope of a role entry. If this attribute is not present, the scope of the role is defined by the LDAPsubentry. Otherwise, the scope is the union of the scope defined by the LDAPsubentry and the scope defined in this attribute.

Syntax

DN, single-valued.

Usage

Attribute specific to this Directory Server instance and version of the schema.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name nsSizeLimit – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.571
NAME 'nsSizeLimit'
DESC 'Binder-based search operation size limit (entries)'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation
X-DS-USE 'internal'
X-ORIGIN 'Sun ONE Directory Server' )
```

Description This attribute specifies the maximum number of entries returned in response to a search operation.

Syntax Integer, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema. Operational attribute used by the directory service.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name nsTimeLimit – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.572
NAME 'nsTimeLimit'
DESC 'Binder-based search operation time limit (seconds)'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE
USAGE directoryOperation
X-DS-USE 'internal'
X-ORIGIN 'Sun ONE Directory Server' )
```

Description This attribute specifies the maximum time spent processing a search operation.

Syntax Integer, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema. Operational attribute used by the directory service.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name numSubordinates – count of immediate subordinates

Synopsis

```
( 1.3.1.1.4.1.453.16.2.103
 NAME ’numSubordinates’
 DESC ’count of immediate subordinates’
 EQUALITY integerMatch
 ORDERING integerOrderingMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 NO-USER-MODIFICATION
 USAGE directoryOperation
 X-ORIGIN ’numSubordinates Internet Draft’ )
```

Description Indicates how many immediate subordinates an entry has.

For example, `numSubordinates=0` in a leaf entry.

Syntax Integer, single-valued.

Usage Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

The value of this attribute may only be modified by the server.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, numSubordinates Internet Draft

Name

o, organizationname – Standard LDAP attribute type

Synopsis

```
( 2.5.4.10
 NAME ( 'o' 'organizationname' )
 DESC 'Standard LDAP attribute type'
 SUP name
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2256' )
```

Description

Identifies the name of the organization.

Syntax

Directory String, multi-valued.

Examples

```
organizationName: Example, Inc.
```

or

```
o: Example, Inc.
```

Attributes

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name objectClass – Standard LDAP attribute type

Synopsis

```
( 2.5.4.0
 NAME ’objectClass’
 DESC ’Standard LDAP attribute type’
 EQUALITY objectIdentifierMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’RFC 2256 (XXX: syntax should be ...38)’ )
```

Description Specifies the object classes of the object. Must include the object.

Syntax Directory String, multi-valued.

Examples

```
objectClass: person
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name

objectClasses – Standard LDAP attribute type

Synopsis

```
( 2.5.21.6
 NAME 'objectClasses'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2252' )
```

Description

Multi-valued attribute that defines the object classes used in a subschema. Each value defines one object class.

Syntax

Directory String, multi-valued.

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2252

Name obsoletedByDocument – Pilot attribute type

Synopsis

```
( 0.9.2342.19200300.102.1.4
 NAME 'obsoletedByDocument'
 DESC 'Pilot attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-ORIGIN 'Internet White Pages Pilot' )
```

Description Contains the distinguished name of a document that obsoletes the document entry.

Syntax DN, multi-valued.

Examples

```
obsoletedbyDocument: cn=Doc Version 2, ou=Document Library,dc=example, dc=com
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Internet White Pages Pilot

Name obsoletesDocument – Pilot attribute type

Synopsis

```
( 0.9.2342.19200300.102.1.3
 NAME 'obsoletesDocument'
 DESC 'Pilot attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-ORIGIN 'Internet White Pages Pilot' )
```

Description Contains the distinguished name of a document that is obsoleted by the document entry.

Syntax DN, multi-valued.

Examples

```
obsoletesDocument: cn=Doc Version 1, ou=Document Library,dc=example, dc=com
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Internet White Pages Pilot

Name

oncRpcNumber – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.18
 NAME 'oncRpcNumber'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 X-ORIGIN 'RFC 2307' )
```

Description

The Open Network Computing (ONC) Remote Procedure Call (RPC) number.

Syntax

Integer, single-valued.

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name

organizationalStatus – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.45
 NAME 'organizationalStatus'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 1274' )
```

Description

Specifies a category by which a person is often referred to in an organization.

Syntax

Directory String, multi-valued.

Examples

```
organizationalStatus: researcher
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name otherMailbox – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.22
 NAME ’otherMailbox’
 DESC ’Standard LDAP attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’RFC 1274’ )
```

Description Specifies values for electronic mailbox types other than X.400 and rfc822.

Syntax Directory String, multi-valued.

Examples

```
otherMailbox: Telemail: x378: Joe
```

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name

ou, organizationalUnitName – Standard LDAP attribute type

Synopsis

```
( 2.5.4.11
 NAME ( 'ou' 'organizationalUnitName' )
 DESC 'Standard LDAP attribute type'
 SUP name
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2256' )
```

Description

Identifies the name of an organizational unit.

Syntax

Directory String, multi-valued.

Examples

```
organizationalUnitName: Marketing
```

or

```
ou: Marketing
```

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name

owner – Standard LDAP attribute type

Synopsis

```
( 2.5.4.32
 NAME ’owner’
 DESC ’Standard LDAP attribute type’
 SUP distinguishedName
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-ORIGIN ’RFC 2256’ )
```

Description

Identifies the distinguished name of the person responsible for the entry.

Syntax

DN, multi-valued.

Examples

```
owner: cn=Babs Jensen, dc=example, dc=com
```

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name pager, pagerTelephoneNumber – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.42
 NAME ( ’pager’ ’pagerTelephoneNumber’ )
 DESC ’Standard LDAP attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.50
 X-ORIGIN ’RFC 1274’ )
```

Description Identifies the entry's pager phone number.

Syntax Telephone Number, multi-valued.

Examples

```
pagerTelephoneNumber: 415-555-6789
```

or

```
pager: 415-555-6789
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name passwordAllowChangeTime – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.214
 NAME ’passwordAllowChangeTime’
 DESC ’Sun ONE defined password policy attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
 SINGLE-VALUE
 USAGE directoryOperation
 X-DS-USE ’internal’
 X-ORIGIN ’Sun ONE Directory Server’ )
```

Description Indicates the exact time after which the user can change their password.

Syntax Generalized Time, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name passwordChange – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.102
 NAME 'passwordChange'
 DESC 'Sun ONE defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Indicates whether users may change their passwords.

This attribute may be on or off. The default value is on. If this attribute is not present, a value of on is assumed.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name passwordCheckSyntax – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.103
 NAME ’passwordCheckSyntax’
 DESC ’Sun ONE defined password policy attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE ’internal’
 X-ORIGIN ’Sun ONE Directory Server’ )
```

Description Indicates whether the password syntax will be checked before the password is saved. The password syntax checking mechanism verifies that the password meets the password minimum length requirement. The password syntax checking mechanism also verifies that the password does not contain any attribute value stored in the uid, cn, sn, givenName, ou, or mail attributes of the user entry.

This attribute may be `on` or `off`. The default value is `off`.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name passwordExp – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.98
 NAME 'passwordExp'
 DESC 'Sun ONE defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Indicates whether user passwords will expire after a specified number of seconds. By default, passwords do not expire. When password expiration is enabled, you can set the number of seconds after which the password will expire with the `passwordMaxAge` attribute.

This attribute may be `on` or `off`. The default value is `off`.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name passwordExpirationTime – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.91
 NAME 'passwordExpirationTime'
 DESC 'Sun ONE defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
 SINGLE-VALUE
 USAGE directoryOperation
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Indicates the exact time after which the user's password expires.

Syntax Generalized Time, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name passwordExpireWithoutWarning – Sun ONE defined password policy attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.86
 NAME 'passwordExpireWithoutWarning'
 DESC 'Sun ONE defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Indicates whether a password can expire regardless of whether the user was warned about the expiration date.

This attribute may be on or `off`. The default value is `off`.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name passwordExpWarned – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.92
 NAME ’passwordExpWarned’
 DESC ’Sun ONE defined password policy attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 USAGE directoryOperation
 X-DS-USE ’internal’
 X-ORIGIN ’Sun ONE Directory Server’ )
```

Description Indicates that a password expiration warning has been sent to the user.

Syntax Directory String, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name passwordHistory – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.96
 NAME ’passwordHistory’
 DESC ’Sun ONE defined password policy attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
 USAGE directoryOperation
 X-DS-USE ’internal’
 X-ORIGIN ’Sun ONE Directory Server’ )
```

Description Contains the history of the user's previous passwords.

Syntax Binary, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name passwordInHistory – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.101
 NAME 'passwordInHistory'
 DESC 'Sun ONE defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Indicates the number of passwords Directory Server stores in history. The valid range of this attribute is 0 to 24. Passwords that are stored in history cannot be reused.

The password history is disabled by default. The default value of this attribute is 0. This implies that the server does not store any old passwords. By default, users can reuse old passwords.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name passwordLockout – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.105
 NAME 'passwordLockout'
 DESC 'Sun ONE defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Enables the account lockout mechanism. If this attribute is set to `on`, users are locked out of the directory once the maximum number of consecutive failed bind attempts has been reached. The maximum number of consecutive bind attempts is specified by the `passwordMaxFailure` attribute. Users remain locked out for the length of time specified by the `passwordLockoutDuration` attribute.

This attribute may be `on` or `off`. The default value is `off`.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name passwordLockoutDuration – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.109
 NAME 'passwordLockoutDuration'
 DESC 'Sun ONE defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Specifies the length of time in seconds during which users will be locked out of the directory. The lockout duration is enabled when `passwordLockout` is set to `on`. If this attribute is not present, or is set to `0`, the account remains locked until it is reset by an administrator.

The valid range of this attribute is 0 to the maximum 32-bit integer value (2147483647) in seconds, with a default of 3600 seconds, or one hour.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name passwordMaxAge – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.97
 NAME 'passwordMaxAge'
 DESC 'Sun ONE defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Indicates the number of seconds after which user passwords will expire. The valid range of this attribute is 1 to the maximum 32-bit integer value (2147483647) in seconds. To use this attribute, you must enable password expiration with the `passwordExp` attribute.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name passwordMaxFailure – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.106
 NAME 'passwordMaxFailure'
 DESC 'Sun ONE defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Specifies the number of consecutive failed bind attempts after which a user is locked out of the directory when `passwordLockout` is set to `on`. Each time an invalid password is used to bind, the password failure counter is incremented. The value of the counter is stored on the operational attribute, `passwordRetryCount`.

The valid range of this attribute is 0 to 32767, with a default value of 3. When set to `0`, this attribute disables lockout.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name passwordMinAge – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.222
 NAME 'passwordMinAge'
 DESC 'Sun ONE defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Specifies the number of seconds that must elapse betweenpassword modifications. Use this attribute with the `passwordInHistory` attribute to prevent users from quickly cycling through passwords so they can use their old passwords again. The default value, 0, indicates that the user can change the password again immediately after updating the password. The valid range of this attribute is from 0 to 2147472000 seconds, which is 24855 days.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name passwordMinLength – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.99
 NAME 'passwordMinLength'
 DESC 'Sun ONE defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Specifies the minimum number of characters that must be used in a password. Syntax checking is performed based on this attribute when `passwordCheckSyntax` is set to `on`. The valid range of this attribute is 2 to 512 characters, with the default value being 6 characters.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name passwordMustChange – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.220
 NAME 'passwordMustChange'
 DESC 'Sun ONE defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Indicates whether users must change their passwords when they first bind to Directory Server, or when the password has been reset by the administrator.

When this attribute is set to `on`, attempts to bind result in a DSA is unwilling to perform error (53), with additional information, Password was reset and must be changed. For users to change their password, the `passwordChange` attribute must be set to `on`.

This attribute may be `on` or `off`. The default value is `off`.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name passwordNonRootMayResetUserpwd – LDAP attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.782
 NAME ’passwordNonRootMayResetUserpwd’
 DESC ’Sun ONE defined password policy attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE ’internal’
 X-ORIGIN ’Sun ONE Directory Server’ )
```

Description Whether a user other than Directory Manager may reset user passwords.

Syntax Directory String, multi-valued.

Examples

```
passwordNonRootMayResetUserpwd: on
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name passwordPolicySubentry – Sun ONE defined password policy attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.30
 NAME ('passwordPolicySubentry' 'pwdPolicySubentry')
 DESC 'Sun ONE defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 SINGLE-VALUE
 USAGE directoryOperation
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description The DN of an LDAPsubentry containing the password policy attributes that will be applied to a user entry.

Syntax DN, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name passwordResetDuration – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.107
 NAME ’passwordResetDuration’
 DESC ’Sun ONE defined password policy attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE ’internal’
 X-ORIGIN ’Sun ONE Directory Server’ )
```

Description This attribute specifies in seconds the period of time that passes before the server resets the retry count to zero.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples

```
passwordResetDuration: 600
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name passwordResetFailureCount – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.223
 NAME 'passwordResetFailureCount'
 DESC 'Sun ONE defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Specifies the length of time in seconds after which the password failure is reset to `0`, even if no successful authentication occurs. The counter is stored in the operational attribute, `passwordRetryCount`.

The valid range for this attribute is 0 to the maximum 32-bit integer value (2147483647) in seconds, with a default of 600 seconds, meaning five minutes. When this attribute is set to `0`, the failure counter is reset only when a successful bind occurs.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name passwordRetryCount – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.93
 NAME 'passwordRetryCount'
 DESC 'Sun ONE defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 USAGE directoryOperation
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Counts the number of consecutive failed attempts at entering the correct password.

Syntax Directory String, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name

passwordRootdnMayBypassModsChecks – Sun ONE defined password policy attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.468
 NAME 'passwordRootdnMayBypassModsChecks'
 DESC 'Sun ONE defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description

When set to on, this password policy attribute allows the root DN to modify passwords, even if the modification violates the password policy. This allows exceptions to the password policy. If the Directory Manager changes a password and the server detects that the new password violates the minimum length or the password history, a warning is logged, but the modification proceeds.

The default value is off, meaning the server rejects even changes to passwords by the Directory Manager if such changes violate the specified password policy.

Syntax

Directory String, single-valued.

Usage

Attribute specific to this Directory Server instance and version of the schema.

Examples

```
passwordRootdnMayBypassModsChecks: on
```

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name passwordStorageScheme – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.221
 NAME 'passwordStorageScheme'
 DESC 'Sun ONE defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Specifies the algorithm used to hash Directory Server passwords. The default password storage scheme is the Salted Secure Hash Algorithm (SSHA).

The following hash types are supported:

- SSHA (Salted Secure Hash Algorithm) is the recommended method as it is the most secure.
- SHA (Secure Hash Algorithm) a version in use before SSHA.
- CRYPT is the UNIX crypt algorithm. It is provided for compatibility with UNIX passwords and supports MD5, Blowfish, and other strong algorithms. To specify the algorithm used, give the format of the salt in the `nsslapd-pluginarg()()` argument as follows:

  ```
  nsslapd-pluginarg(): value()
  ```

  The value is in the `snprintf` format corresponding to specific salt formats. For example, some of the formats supported include `%.2s`, `$1$%.8s`, `$2a$04$%.22s`, and `$md5$%.8s$`. If the string value maps to an algorithm that is not supported by the operating system, then a warning message is logged and the hash will be made using the default UNIX algorithm with a salt made of 31 random characters.

If this attribute is set to CLEAR, passwords are not encrypted and appear in plain text.

You can extend how password attributes are stored by writing your own password storage scheme plug-in.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples `passwordStorageScheme: CLEAR`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name passwordUnlock – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.108
 NAME 'passwordUnlock'
 DESC 'Sun ONE defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Specifies whether user accounts will be unlocked after a period of time when `passwordLockout` is set to `on`. The period of time is specified with the `passwordLockoutDuration` attribute.

If this attribute is set to `on`, and the value of the `passwordMaxFailure` attribute has been reached, then the account is unlocked after the number of seconds specified in the `passwordLockoutDuration` attribute. If this attribute is set to `off`, the account remains locked until an administration resets it.

This attribute may be `on` or `off`. The default value is `on`.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name passwordWarning – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.104
 NAME 'passwordWarning'
 DESC 'Sun ONE defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Specifies the number of seconds before a user password expires that the user will receive a password expiration warning on attempting to authenticate to the directory.

The server does not send a warning directly to the end user. Instead, the server returns the warning to the client application.

Note – End users *do not automatically receive email or other notification* as a result of the `passwordWarning` attribute being set to `on`. Make sure the warning received by the client application is appropriately delivered to the end user.

The valid range for this attribute is 1 to the maximum 32-bit integer value (2147483647) in seconds, with the default value set at 86400 seconds, meaning 1 day.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name

personalSignature – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.53
 NAME 'personalSignature'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
 X-ORIGIN 'RFC 1274' )
```

Description

A signature file, in binary format, for the entry.

Syntax

Binary, multi-valued.

Examples

```
personalSignature:: AAAAAA==
```

Attributes

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name personalTitle – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.40
 NAME 'personalTitle'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 1274' )
```

Description Specifies a personal title for a person. Examples of personal titles are `Ms`, `Dr`, `Prof`, and `Rev`.

Syntax Directory String, multi-valued.

Examples

```
personalTitle: Mr
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name

photo – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.7
 NAME 'photo'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
 X-ORIGIN 'RFC 1274' )
```

Description

Contains a photo, in binary form, of the entry.

Syntax

Binary, multi-valued.

Examples

```
photo:: AAAAAA==
```

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name physicalDeliveryOfficeName – Standard LDAP attribute type

Synopsis

```
( 2.5.4.19
 NAME ’physicalDeliveryOfficeName’
 DESC ’Standard LDAP attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’RFC 2256’ )
```

Description Identifies the name of the city or village in which a physical delivery office is located.

Syntax Directory String, multi-valued.

Examples `physicalDeliveryOfficeName: Santa Clara`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name postalAddress – Standard LDAP attribute type

Synopsis

```
( 2.5.4.16
 NAME 'postalAddress'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2256' )
```

Description Identifies the mailing address for the entry. This field is intended to include multiple lines. When represented in LDIF format, each line should be separated by a dollar sign ($).

Syntax Directory String, multi-valued.

Examples

```
postalAddress: P.O. Box 3541$Santa Clara, CA$99555
```

To represent an actual dollar sign ($) or backslash (\) within the text, use the escaped hex values \24 and \5c respectively.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name postalCode – Standard LDAP attribute type

Synopsis

```
( 2.5.4.17
 NAME 'postalCode'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2256' )
```

Description Identifies the entry's zip code in the United States.

Syntax Directory String, multi-valued.

Examples

```
postalCode: 44224
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name

postOfficeBox – Standard LDAP attribute type

Synopsis

```
( 2.5.4.18
 NAME 'postOfficeBox'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2256' )
```

Description

Specifies a postal mailing address.

Syntax

Directory String, multi-valued.

Examples

```
postOfficeBox: P.O. Box 1234
```

Attributes

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name preferredDeliveryMethod – Standard LDAP attribute type

Synopsis

```
( 2.5.4.28
 NAME 'preferredDeliveryMethod'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-ORIGIN 'RFC 2256' )
```

Description Identifies the entry's preferred contact or delivery method.

Syntax Directory String, single-valued.

Examples

```
preferredDeliveryMethod: telephone
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name preferredLanguage – inetOrgPerson attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.39
 NAME 'preferredLanguage'
 DESC 'inetOrgPerson attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-ORIGIN 'inetOrgPerson Internet Draft' )
```

Description Defines a person's preferred written or spoken language. The value for this attribute should conform to the syntax for HTTP Accept-Language header values.

Syntax Directory String, single-valued.

Examples

```
preferredLanguage: en-us
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, RFC 2798

Name presentationAddress – Standard LDAP attribute type

Synopsis

```
( 2.5.4.29
 NAME 'presentationAddress'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 SINGLE-VALUE
 X-ORIGIN 'RFC 2256' )
```

Description Contains an OSI presentation address for the entry. The presentation address consists of an OSI Network Address and up to three selectors, one each for use by the transport, session, and presentation entities.

Syntax IA5 String, single-valued.

Examples

```
presentationAddress: TELEX+00726322+RFC-1006+02+130.59.2.1
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name

protocolInformation – Standard LDAP attribute type

Synopsis

```
( 2.5.4.48
 NAME 'protocolInformation'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2256' )
```

Description

Used in conjunction with the presentationAddress attribute to provide additional information to the OSI network service.

Syntax

Directory String, multi-valued.

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name pwdAccountLockedTime – Directory Server defined password policy attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.17
 NAME ’pwdAccountLockedTime’
 DESC ’Directory Server defined password policy attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
 SINGLE-VALUE
 NO-USER-MODIFICATION
 USAGE directoryOperation
 X-DS-USE ’internal’
 X-ORIGIN ’Password Policy for LDAP Directories Internet Draft’ )
```

Description Holds the time that the user's account was locked. A locked account means that the password may no longer be used to authenticate.

A value of `000001010000Z` means the account has been locked permanently, and that only a password administrator can unlock the account.

Syntax Generalized Time, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

The value of this attribute may only be modified by the server.

Examples `pwdAccountLockedTime: 20050103121520Z`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name pwdAllowUserChange – LDAP attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.14
 NAME 'pwdAllowUserChange'
 DESC 'Password Allow User Change'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
 X-DS-USE 'internal'
 X-ORIGIN 'Password Policy for LDAP Directories Internet Draft' )
```

Description Indicates whether users can change their own passwords, although the change operation is still subject to access control.

If this attribute is not present, a value of `TRUE` is assumed. This attribute is intended to be used in the absense of an access control mechanism.

Syntax Boolean, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples `pwdAllowUserChange: TRUE`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name pwdAttribute – LDAP attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.1
 NAME ’pwdAttribute’
 EQUALITY objectIdentifierMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 )
```

Description Holds the name of the attribute to which the password policy is applied. Currently only `userPassword` can be used.

Syntax Object Identifier, multi-valued.

Examples

```
pwdAttribute: userPassword
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name

pwdChangedTime – Directory Server defined password policy attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.16
 NAME 'pwdChangedTime'
 DESC 'Directory Server defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
 SINGLE-VALUE
 NO-USER-MODIFICATION
 USAGE directoryOperation
 X-DS-USE 'internal'
 X-ORIGIN 'Password Policy for LDAP Directories Internet Draft' )
```

Description

Specifies the last time the entry's password was changed. This is used by the password expiration policy.

If this attribute is not present, the password will never expire.

Syntax

Generalized Time, single-valued.

Usage

Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

The value of this attribute may only be modified by the server.

Examples

```
pwdChangedTime: 20050103121520Z
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name pwdCheckQuality – LDAP attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.5
 NAME 'pwdCheckQuality'
 DESC 'Level of required quality'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 X-DS-USE 'internal'
 X-ORIGIN 'Password Policy for LDAP Directories Internet Draft' )
```

Description Indicates how the password quality will be verified while being modified or added. This attribute can take the following values:

`0` Default. Quality checking is not enforced.

`1` Directory Server checks the quality of the password. If Directory Server cannot determine the quality of the password, because the password is hashed for example, it accepts the password and logs a warning message.

`2` Directory Server checks the quality of the password. If Directory Server cannot determine the quality of the password, it returns `LDAP_CONSTRAINT_VIOLATION` and refuses the operation.

If the password value is already hashed, and prefixed by a tag other than `{CLEAR}` setting this attribute to `1` means the server does not check quality and logs a warning. Setting this attribute to `2` in this case causes the server to reject the modification because it cannot check the password quality.

Syntax Integer, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples

```
pwdCheckQuality: 1
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name pwdExpireWarning – Password Warning Expiration

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.7
 NAME ’pwdExpireWarning’
 DESC ’Password Warning Expiration’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 X-DS-USE ’internal’
 X-ORIGIN ’Password Policy for LDAP Directories Internet Draft’ )
```

Description Specifies the maximum number of seconds before a password is due to expire that expiration warning messages will be returned to an authenticating user.

If this attribute is not present, or if the value is `0` no warnings will be returned. If not `0`, the value must be smaller than the value of the pwdMaxAge attribute.

Syntax Integer, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples `pwdExpireWarning: 604800`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name pwdFailureCountInterval – Password Failure Count Interval

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.12
 NAME ’pwdFailureCountInterval’
 DESC ’Password Failure Count Interval’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 X-DS-USE ’internal’
 X-ORIGIN ’Password Policy for LDAP Directories Internet Draft’ )
```

Description Holds the number of seconds after which the password failures are purged from the failure counter, even though no successful authentication occurred.

If this attribute is not present, or if the value is `0` the failure counter is only reset by a successful authentication.

Syntax Integer, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples

```
pwdFailureCountInterval: 600
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name pwdFailureTime – Directory Server defined password policy attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.19
 NAME 'pwdFailureTime'
 DESC 'Directory Server defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
 NO-USER-MODIFICATION
 USAGE directoryOperation
 X-DS-USE 'internal'
 X-ORIGIN 'Password Policy for LDAP Directories Internet Draft' )
```

Description Holds the timestamps of consecutive authentication failures.

Syntax Generalized Time, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

The value of this attribute may only be modified by the server.

Examples

```
pwdFailureTime: 20050103121520Z
pwdFailureTime: 20050103121742Z
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name pwdGraceAuthNLimit – Password Grace Login

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.8
 NAME ’pwdGraceAuthNLimit’
 DESC ’Password Grace Login’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 X-DS-USE ’internal’
 X-ORIGIN ’Password Policy for LDAP Directories Internet Draft’ )
```

Description Specifies the number of times an expired password can be used to authenticate.

If this attribute is not present, or if the value is `0` authentication will fail.

Syntax Integer, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples

```
pwdGraceAuthNLimit: 3
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name pwdGraceUseTime – Directory Server defined password policy attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.21
 NAME ’pwdGraceUseTime’
 DESC ’Directory Server defined password policy attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
 NO-USER-MODIFICATION
 USAGE directoryOperation
 X-DS-USE ’internal’
 X-ORIGIN ’Password Policy for LDAP Directories Internet Draft’ )
```

Description Holds the timestamps of the grace authentications allowed after the password expired.

Syntax Generalized Time, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

The value of this attribute may only be modified by the server.

Examples

```
pwdGraceUseTime: 20050103121520Z
```

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name pwdHistory – Directory Server defined password policy attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.20
 NAME 'pwdHistory'
 DESC 'Directory Server defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
 NO-USER-MODIFICATION
 USAGE directoryOperation
 X-DS-USE 'internal'
 X-ORIGIN 'Password Policy for LDAP Directories Internet Draft' )
```

Description Holds a history of previously used passwords.

Syntax Binary, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

The value of this attribute may only be modified by the server.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name pwdInHistory – Number of Passwords in history

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.4
 NAME ’pwdInHistory’
 DESC ’Number of Passwords in history’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 X-DS-USE ’internal’
 X-ORIGIN ’Password Policy for LDAP Directories Internet Draft’ )
```

Description Specifies the maximum number of used passwords stored in the pwdHistory attribute.

If this attribute is not present, or if the value is `0`, used passwords are not stored in the pwdHistory attribute and thus may be reused.

Syntax Integer, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples

```
pwdInHistory: 3
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name pwdIsLockoutPrioritized – LDAP attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.794
 NAME 'pwdIsLockoutPrioritized'
 DESC 'Password Lockout Replication Priority'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
 SINGLE-VALUE
 X-DS-USE 'internal'
 X-ORIGIN 'Sun Directory Server' )
```

Description This attribute specifies whether prioritized replication is used to copy account lockout attribute values.

Syntax Boolean, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples

```
pwdIsLockoutPrioritized: TRUE
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name pwdKeepLastAuthTime – Enable last authentication time recording

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.798
 NAME 'pwdKeepLastAuthTime'
 DESC 'Enable last authentication time recording'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
 SINGLE-VALUE
 X-DS-USE 'internal'
 X-ORIGIN 'Sun Directory Server' )
```

Description Whether the timestamp of the last successful authentication should be stored in the operational attribute `pwdLastAuthTime` on the entry.

Note – Using this feature can affect performance. When you configure Directory Server to save `pwdLastAuthTime` timestamps, the server must perform an internal modify operation for each successful bind.

Syntax Boolean, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples

```
pwdKeepLastAuthTime: TRUE
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name pwdLastAuthTime – Last authentication time

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.797
 NAME ’pwdLastAuthTime’
 DESC ’Last authentication time’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
 SINGLE-VALUE
 NO-USER-MODIFICATION
 USAGE directoryOperation
 X-DS-USE ’internal’
 X-ORIGIN ’Sun Directory Server’ )
```

Description The timestamp of the last successful authentication involving the entry. Activate this attribute by setting `pwdKeepLastAuthTime` in the password policy entry to `on`.

Syntax Generalized Time, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

The value of this attribute may only be modified by the server.

Examples

```
pwdLastAuthTime: 20060103121520Z
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name pwdLockout – Password Lockout

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.9
 NAME 'pwdLockout'
 DESC 'Password Lockout'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
 X-DS-USE 'internal'
 X-ORIGIN 'Password Policy for LDAP Directories Internet Draft' )
```

Description Indicates, when its value is TRUE, that the password may not be used to authenticate after a specified number of consecutive failed bind attempts. The maximum number of consecutive failed bind attempts is specified in pwdMaxFailure.

If this attribute is not present, or if the value is FALSE the password may be used to authenticate when the number of failed bind attempts has been reached.

Syntax Boolean, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples

```
pwdLockout: TRUE
```

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name pwdLockoutDuration – Password Lockout Duration

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.10
 NAME 'pwdLockoutDuration'
 DESC 'Password Lockout Duration'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 X-DS-USE 'internal'
 X-ORIGIN 'Password Policy for LDAP Directories Internet Draft' )
```

Description Holds the number of seconds that the password cannot be used to authenticate due to too many failed bind attempts.

If this attribute is not present, or if the value is `0` the password cannot be used to authenticate until reset by a password administrator.

Syntax Integer, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples

```
pwdLockoutDuration: 300
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name pwdMaxAge – Password Max Age

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.3
 NAME ’pwdMaxAge’
 DESC ’Password Max Age’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 X-DS-USE ’internal’
 X-ORIGIN ’Password Policy for LDAP Directories Internet Draft’ )
```

Description Holds the number of seconds after which a modified password will expire.

If this attribute is not present, or if the value is `0` the password does not expire. If not 0, the value must be greater than or equal to the value of pwdMinAge.

Syntax Integer, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples

```
pwdMaxAge: 8640000
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name pwdMaxFailure – Password Max Failure

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.11
 NAME 'pwdMaxFailure'
 DESC 'Password Max Failure'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 X-DS-USE 'internal'
 X-ORIGIN 'Password Policy for LDAP Directories Internet Draft' )
```

Description Specifies the number of consecutive failed bind attempts after which the password may not be used to authenticate.

If this attribute is not present, or if the value is `0` this policy is not checked, and the value of pwdLockout is ignored.

Syntax Integer, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples

```
pwdMaxFailure: 3
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name pwdMinAge – Password Min Age

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.2
 NAME ’pwdMinAge’
 DESC ’Password Min Age’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 X-DS-USE ’internal’
 X-ORIGIN ’Password Policy for LDAP Directories Internet Draft’ )
```

Description Holds the number of seconds that must elapse between modifications to the password.

If this attribute is not present, 0 seconds is assumed.

Syntax Integer, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples

```
pwdMinAge: 604800
```

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name pwdMinLength – Password Min Length

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.6
 NAME ’pwdMinLength’
 DESC ’Password Min Length’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 X-DS-USE ’internal’
 X-ORIGIN ’Password Policy for LDAP Directories Internet Draft’ )
```

Description When quality checking is enabled, this attribute holds the minimum number of characters that must be used in a password. If this attribute is not present, no minimum password length will be enforced. If Directory Server is unable to check the length, because the password is hashed for example, Directory Server will, depending on the value of the pwdCheckQuality attribute, either accept the password without checking it (when pwdCheckQuality is `0` or `1`) or return `LDAP_CONSTRAINT_VIOLATION` and refuse to add or modify the password.

Syntax Integer, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples

```
pwdMinLength: 6
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name pwdMustChange – Password Must Change

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.13
 NAME 'pwdMustChange'
 DESC 'Password Must Change'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
 X-DS-USE 'internal'
 X-ORIGIN 'Password Policy for LDAP Directories Internet Draft' )
```

Description Specifies with a value of `TRUE` that users must change their passwords when they first bind to the directory after a password is set or reset by any other user, such as a password administrator, who has the access rights to modify the password.

If this attribute is not present, or if the value is `FALSE` users are not required to change their password upon binding after the password administrator sets or resets the password. This attribute is typically set by a password administrator after resetting a user's password.

When this attribute is set to `TRUE`, attempts to bind result in a DSA is unwilling to perform error (53), with additional information, Password was reset and must be changed.

Syntax Boolean, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples `pwdMustChange: TRUE`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name pwdPolicySubentry – Directory Server defined password policy attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.23
 NAME 'pwdPolicySubEntry'
 DESC 'Directory Server defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
 SINGLE-VALUE
 USAGE directoryOperation
 X-DS-USE 'internal'
 X-ORIGIN 'Password Policy for LDAP Directories Internet Draft' )
```

Description Points to the pwdPolicy subentry in effect for this object.

Syntax DN, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

Examples

```
pwdPolicySubentry: cn=myPwdPolicy,cn=pwp,cn=config
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name pwdReset – Directory Server defined password policy attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.22
 NAME ’pwdReset’
 DESC ’Directory Server defined password policy attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
 SINGLE-VALUE
 NO-USER-MODIFICATION
 USAGE directoryOperation
 X-DS-USE ’internal’
 X-ORIGIN ’Password Policy for LDAP Directories Internet Draft’ )
```

Description Holds a flag to indicate, when TRUE, that the password has been updated by the password administrator and must be changed by the user.

Syntax Boolean, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in ldapsearch only when specifically requested.

The value of this attribute may only be modified by the server.

Examples

```
pwdReset: TRUE
```

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name pwdSafeModify – Password Safe Modify

Synopsis

```
( 1.3.6.1.4.1.42.2.27.8.1.15
 NAME ’pwdSafeModify’
 DESC ’Password Safe Modify’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
 X-DS-USE ’internal’
 X-ORIGIN ’Password Policy for LDAP Directories Internet Draft’ )
```

Description Specifies whether or not the existing password must be sent along with the new password when being changed.

If this attribute is not present, a value of `FALSE` is assumed.

Syntax Boolean, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Examples `pwdSafeModify: TRUE`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name

ref – Standard LDAP referral attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.34
 NAME 'ref'
 DESC 'Standard LDAP referral attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 X-ORIGIN 'RFC 3296' )
```

Description

Used in LDAPv3 to support smart referrals. Contains an LDAP URL in the format:

`ldap://`*servername*`:`*portnumber*`/`*DN*

The port number is optional.

Syntax

IA5 String, multi-valued.

Examples

```
ref: ldap://server.example.com:389/ou=People, o=example.com
```

Note that DN special characters must be escaped. For example:

```
ref: ldap://server.example.com:389/ou=People, o=example%Inc
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 3296

Name registeredAddress – Standard LDAP attribute type

Synopsis

```
( 2.5.4.26
 NAME 'registeredAddress'
 DESC 'Standard LDAP attribute type'
 SUP postalAddress
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2256' )
```

Description This attribute contains a postal address for receiving telegrams or expedited documents. The recipient's signature is usually required on delivery.

Syntax Directory String, multi-valued.

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name replicaIdentifier – RetroChangelog attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.724
 NAME ’replicaIdentifier’
 DESC ’RetroChangelog attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 NO-USER-MODIFICATION
 USAGE directoryOperation
 X-ORIGIN ’Sun Directory Server’ )
```

Syntax Directory String, single-valued.

Usage Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

The value of this attribute may only be modified by the server.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name replicationCSN – RetroChangelog attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.725
 NAME 'replicationCSN'
 DESC 'RetroChangelog attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 NO-USER-MODIFICATION
 USAGE directoryOperation
 X-ORIGIN 'Sun Directory Server' )
```

Description This attribute is used for the retro change log. When the retro change log is enabled, this attribute specifies a change sequence number (CSN) for each record in the retro change log corresponding to a replicated operation. The CSN uniquely identifies each change made to the replicated data.

The CSN contains a timestamp, sequence number, replica ID, and subsequence number.

Syntax Directory String, single-valued.

Usage Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

The value of this attribute may only be modified by the server.

Examples

```
replicationCSN: 451d2c6d000200010000
```

In this example, the change sequence number is concatenated from the following values.

`451d2c6d` This represents time as the number of seconds since January 1, 1970.

`0002` This is the sequence number, which is used to distinguish between operations that happened during the same second.

`0001` This is the replica ID, which in this example is 1.

`0000` This is the subsequence number, which is not always used, but helps the server to manage information about the state of replication.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name retryCountResetTime – Sun ONE defined password policy attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.94
 NAME 'retryCountResetTime'
 DESC 'Sun ONE defined password policy attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
 SINGLE-VALUE
 USAGE directoryOperation
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Specifies the exact time after which the `passwordRetryCount` is reset.

Syntax Generalized Time, single-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name roleOccupant – Standard LDAP attribute type

Synopsis

```
( 2.5.4.33
 NAME 'roleOccupant'
 DESC 'Standard LDAP attribute type'
 SUP distinguishedName
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-ORIGIN 'RFC 2256' )
```

Description Contains the distinguished name of the person acting in the role defined in the `organizationalRole` entry.

Syntax DN, multi-valued.

Examples

```
roleOccupant: uid=jdoe, dc=example, dc=com
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name

roomNumber – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.6
 NAME 'roomNumber'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 1274' )
```

Description

Specifies the room number of an object. Note that the commonName attribute should be used for naming room objects.

Syntax

Directory String, multi-valued.

Examples

```
roomNumber: 230
```

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name searchGuide – Standard LDAP attribute type

Synopsis

```
( 2.5.4.14
 NAME 'searchGuide'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 X-ORIGIN 'RFC 2256' )
```

Description Specifies information for a suggested search criteria when using the entry as the base object in the directory tree for a search operation. When constructing search filters, use `enhancedSearchGuide` instead.

Syntax IA5 String, multi-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name secretary – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.21
 NAME 'secretary'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-ORIGIN 'RFC 1274' )
```

Description Identifies the entry's secretary or administrative assistant.

Syntax DN, multi-valued.

Examples

```
secretary: cn=John Doe, dc=example, dc=com
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name seeAlso – Standard LDAP attribute type

Synopsis

```
( 2.5.4.34
 NAME ’seeAlso’
 DESC ’Standard LDAP attribute type’
 SUP distinguishedName
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-ORIGIN ’RFC 2256’ )
```

Description Identifies another Directory Server entry that may contain information related to this entry.

Syntax DN, multi-valued.

Examples

```
seeAlso: cn=Quality Control Inspectors,ou=manufacturing,dc=example, dc=com
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name serialNumber – Standard LDAP attribute type

Synopsis

```
( 2.5.4.5
 NAME 'serialNumber'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2256' )
```

Description Specifies the serial number of a device.

Syntax Directory String, multi-valued.

Examples

```
serialNumber: 555-1234-AZ
```

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name shadowExpire – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.10
 NAME 'shadowExpire'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 X-ORIGIN 'RFC 2307' )
```

Description Related to the /etc/shadow file, this attribute contains an absolute date specifying when the login may no longer be used.

Syntax Integer, single-valued.

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name shadowFlag – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.11
 NAME 'shadowFlag'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 X-ORIGIN 'RFC 2307' )
```

Description Related to the `/etc/shadow` file, this attribute is currently not used and is reserved for future use.

Syntax Integer, single-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name shadowInactive – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.9
 NAME 'shadowInactive'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 X-ORIGIN 'RFC 2307' )
```

Description Related to the `/etc/shadow` file, this attribute specifies the number of days of inactivity allowed for the specified user.

Syntax Integer, single-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name

shadowLastChange – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.5
 NAME 'shadowLastChange'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 X-ORIGIN 'RFC 2307' )
```

Description

Related to the `/etc/shadow` file, this attribute specifies number of days between January 1, 1970, and the date that the password was last modified.

Syntax

Integer, single-valued.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name shadowMax – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.7
 NAME 'shadowMax'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 X-ORIGIN 'RFC 2307' )
```

Description Related to the `/etc/shadow` file, this attribute specifies the maximum number of days the password is valid.

Syntax Integer, single-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name

shadowMin – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.6
 NAME 'shadowMin'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 X-ORIGIN 'RFC 2307' )
```

Description

Related to the /etc/shadow file, this attribute specifies the minimum number of days required between password changes.

Syntax

Integer, single-valued.

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name shadowWarning – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.8
 NAME 'shadowWarning'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 X-ORIGIN 'RFC 2307' )
```

Description Related to the `/etc/shadow` file, this attribute specifies the number of days before the password expires that the user is warned.

Syntax Integer, single-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name singleLevelQuality – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.50
 NAME 'singleLevelQuality'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-ORIGIN 'RFC 1274' )
```

Description Specifies the purported data quality at the level immediately below in the DIT.

Syntax Directory String, single-valued.

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name sn, surName – Standard LDAP attribute type

Synopsis

```
( 2.5.4.4
 NAME ( ’sn’ ’surName’ )
 DESC ’Standard LDAP attribute type’
 SUP name
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’RFC 2256’ )
```

Description Identifies the entry's surname, also referred to as last name or family name.

Syntax Directory String, multi-valued.

Examples

```
surname: Anderson
```

or

```
sn: Anderson
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name

st, stateOrProvinceName – Standard LDAP attribute type

Synopsis

```
( 2.5.4.8
 NAME ( 'st' 'stateOrProvinceName' )
 DESC 'Standard LDAP attribute type'
 SUP name
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2256' )
```

Description

Identifies the state or province in which the entry resides. Abbreviation: st.

Syntax

Directory String, multi-valued.

Examples

```
stateOrProvinceName: California
```

or

```
st: California
```

Attributes

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name street, streetaddress – Standard LDAP attribute type

Synopsis

```
( 2.5.4.9
 NAME ( ’street’ ’streetaddress’ )
 DESC ’Standard LDAP attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’RFC 2256’ )
```

Description Identifies the entry's house number and street name.

Syntax Directory String, multi-valued.

Examples

```
streetAddress: 1234 Ridgeway Drive
```

or

```
street: 1234 Ridgeway Drive
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name subject – Pilot attribute type

Synopsis

```
( 0.9.2342.19200300.102.1.8
 NAME 'subject'
 DESC 'Pilot attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'Internet White Pages Pilot' )
```

Description Contains information about the subject matter of the document entry.

Syntax Directory String, multi-valued.

Examples

```
subject: employee option grants
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Internet White Pages Pilot

Name subschemaSubentry – Standard LDAP attribute type

Synopsis

```
( 2.5.18.10
 NAME 'subschemaSubentry'
 DESC 'Standard LDAP attribute type'
 EQUALITY distinguishedNameMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 SINGLE-VALUE
 NO-USER-MODIFICATION
 USAGE directoryOperation
 X-ORIGIN 'RFC 2252' )
```

Description DN of the entry that contains schema information for this entry. This attribute is present for every entry in the directory.

Syntax DN, single-valued.

Usage Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

The value of this attribute may only be modified by the server.

Examples

```
subschemaSubentry: cn=schema
```

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2252

Name subtreeMaximumQuality – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.52
 NAME 'subtreeMaximumQuality'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-ORIGIN 'RFC 1274' )
```

Description Specifies the purported maximum data quality for a DIT subtree.

Syntax Directory String, single-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name subtreeMinimumQuality – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.51
 NAME ’subtreeMinimumQuality’
 DESC ’Standard LDAP attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 X-ORIGIN ’RFC 1274’ )
```

Description Specifies the purported minimum data quality for a DIT subtree.

Syntax Directory String, single-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name

supportedAlgorithms – Standard LDAP attribute type

Synopsis

```
( 2.5.4.52
 NAME 'supportedAlgorithms'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
 X-ORIGIN 'RFC 2256' )
```

Description

This attribute is to be stored and requested in the binary form, as `supportedAlgorithms;binary`.

Syntax

Binary, multi-valued.

Examples

```
supportedAlgorithms;binary: AAAAAA==
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name supportedApplicationContext – Standard LDAP attribute type

Synopsis

```
( 2.5.4.30
 NAME 'supportedApplicationContext'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2256' )
```

Description This attribute contains the identifiers of OSI application contexts.

Syntax Directory String, multi-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name supportedControl – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.4.1.1466.101.120.13
 NAME 'supportedControl'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 USAGE dsaOperation
 X-ORIGIN 'RFC 2252' )
```

Description The values of this attribute are the object identifiers (OIDs) that identify the controls supported by the server. When the server does not support controls, this attribute is absent.

Syntax Directory String, multi-valued.

Usage Operational attribute used by a Directory Server instance; returned in `ldapsearch` only when specifically requested.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2252

Name supportedExtension – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.4.1.1466.101.120.7
 NAME 'supportedExtension'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 USAGE dsaOperation
 X-ORIGIN 'RFC 2252' )
```

Description The values of this attribute are the object identifiers (OIDs) that identify the supported extended operations supported by the server. When the server does not support extensions, this attribute is absent.

Syntax Directory String, multi-valued.

Usage Operational attribute used by a Directory Server instance; returned in `ldapsearch` only when specifically requested.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2252

Name supportedLDAPVersion – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.4.1.1466.101.120.15
 NAME 'supportedLDAPVersion'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 USAGE dsaOperation
 X-ORIGIN 'RFC 2252' )
```

Description Identifies the versions of the LDAP protocol implemented by the server. This attribute is defined in RFC 2252.

Syntax Integer, multi-valued.

Usage Operational attribute used by a Directory Server instance; returned in `ldapsearch` only when specifically requested.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2252

Name supportedSASLMechanisms – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.4.1.1466.101.120.14
 NAME 'supportedSASLMechanisms'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 USAGE dsaOperation
 X-ORIGIN 'RFC 2252' )
```

Description Identifies the names of supported SASL mechanisms supported by the server. When the server does not support SASL attributes, this attribute is absent.

Syntax Directory String, multi-valued.

Usage Operational attribute used by a Directory Server instance; returned in `ldapsearch` only when specifically requested.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2252

Name supportedSSLCiphers – List of ciphers supported by SSL lib

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.800
 NAME ’supportedSSLCiphers’
 DESC ’List of ciphers supported by SSL lib’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 USAGE dSAOperation
 X-ORIGIN ’Sun Directory Server’ )
```

Description This attribute contains the list of SSL ciphers supported by Directory Server, and can be read from the root DSE. The content of this attribute is dynamically loaded from the library providing SSL support. The ciphers listed here can be enabled by adding the values to the list contained in `nsSSL3Ciphers` on `cn=encryption,cn=config`.

Syntax Directory String, multi-valued.

Usage Operational attribute used by a Directory Server instance; returned in `ldapsearch` only when specifically requested.

Examples

```
supportedSSLCiphers: TLS_DHE_RSA_WITH_AES_256_CBC_SHA
supportedSSLCiphers: TLS_DHE_DSS_WITH_AES_256_CBC_SHA
supportedSSLCiphers: TLS_RSA_WITH_AES_256_CBC_SHA
supportedSSLCiphers: TLS_DHE_DSS_WITH_RC4_128_SHA
supportedSSLCiphers: TLS_DHE_RSA_WITH_AES_128_CBC_SHA
supportedSSLCiphers: TLS_DHE_DSS_WITH_AES_128_CBC_SHA
supportedSSLCiphers: SSL_RSA_WITH_RC4_128_MD5
supportedSSLCiphers: SSL_RSA_WITH_RC4_128_SHA
supportedSSLCiphers: TLS_RSA_WITH_AES_128_CBC_SHA
supportedSSLCiphers: SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
supportedSSLCiphers: SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA
supportedSSLCiphers: SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA
supportedSSLCiphers: SSL_RSA_WITH_3DES_EDE_CBC_SHA
supportedSSLCiphers: SSL_DHE_RSA_WITH_DES_CBC_SHA
supportedSSLCiphers: SSL_DHE_DSS_WITH_DES_CBC_SHA
supportedSSLCiphers: SSL_RSA_FIPS_WITH_DES_CBC_SHA
supportedSSLCiphers: SSL_RSA_WITH_DES_CBC_SHA
supportedSSLCiphers: TLS_RSA_EXPORT1024_WITH_RC4_56_SHA
supportedSSLCiphers: TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA
supportedSSLCiphers: SSL_RSA_EXPORT_WITH_RC4_40_MD5
supportedSSLCiphers: SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5
supportedSSLCiphers: SSL_RSA_WITH_NULL_SHA
supportedSSLCiphers: SSL_RSA_WITH_NULL_MD5
supportedSSLCiphers: SSL_CK_RC4_128_WITH_MD5
supportedSSLCiphers: SSL_CK_RC2_128_CBC_WITH_MD5
supportedSSLCiphers: SSL_CK_DES_192_EDE3_CBC_WITH_MD5
supportedSSLCiphers: SSL_CK_DES_64_CBC_WITH_MD5
supportedSSLCiphers: SSL_CK_RC4_128_EXPORT40_WITH_MD5
supportedSSLCiphers: SSL_CK_RC2_128_CBC_EXPORT40_WITH_MD5
```

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name targetDn – Changelog attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.6
 NAME 'targetDn'
 DESC 'Changelog attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-ORIGIN 'Changelog Internet Draft' )
```

Description Contains the DN of the entry that was affected by the LDAP operation. In the case of a `modrdn` operation, the `targetDn` attribute contains the DN of the entry before it was modified or moved.

Syntax DN, multi-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Changelog Internet Draft

Name targetUniqueId – RetroChangelog attribute type

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.1.596
 NAME 'targetUniqueId'
 DESC 'RetroChangelog attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 NO-USER-MODIFICATION
 USAGE directoryOperation
 X-ORIGIN 'Sun Directory Server' )
```

Description This attribute is used for the retro change log. When the retro change log is enabled, this attribute provides the unique ID of the target entry for each record in the retro change log.

Syntax Directory String, single-valued.

Usage Operational attribute used by the directory service; returned in `ldapsearch` only when specifically requested.

The value of this attribute may only be modified by the server.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name telephoneNumber – Standard LDAP attribute type

Synopsis

```
( 2.5.4.20
 NAME 'telephoneNumber'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.50
 X-ORIGIN 'RFC 2256' )
```

Description Identifies the entry's phone number.

Syntax Telephone Number, multi-valued.

Examples `telephoneNumber: 415-555-2233`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name teletexTerminalIdentifier – Standard LDAP attribute type

Synopsis

```
( 2.5.4.22
 NAME 'teletexTerminalIdentifier'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2256' )
```

Description Identifies the entry's teletex terminal identifier. The format of the attribute is as follows:

teletex-id = *ttx-term* 0*("$" *ttx-param*)

ttx-term = *printablestring*

ttx-param = *ttx-key* ":" *ttx-value*

ttx-key = "graphic" / "control" / "misc" / "page" / "private"

ttx-value = *octetstring*

The first printable string is the encoding of the first portion of the teletex terminal identifier to be encoded, and the subsequent 0 or more octet strings are subsequent portions of the teletex terminal identifier.

Syntax Directory String, multi-valued.

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name

telexNumber – Standard LDAP attribute type

Synopsis

```
( 2.5.4.21
 NAME ’telexNumber’
 DESC ’Standard LDAP attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’RFC 2256’ )
```

Description

Defines the telex number of the entry. The format of the telex number is as follows:

```
actual-number "$" country "$" answerback
```

where:

- actual-number: the syntactic representation of the number portion of the TELEX number being encoded.
- country: the TELEX country code.
- answerback: the answerback code of a TELEX terminal.

Syntax

Directory String, multi-valued.

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name textEncodedORAddress – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.2
 NAME ’textEncodedORAddress’
 DESC ’Standard LDAP attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN ’RFC 1274’ )
```

Description Defines the text-encoded Originator/Recipient (X.400) address of the entry as defined in RFC987.

Syntax Directory String, multi-valued.

Examples

```
textEncodedORAddress: /S=doe/OU=eng/O=example/ADMD=telemail/C=us/
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name title – Standard LDAP attribute type

Synopsis

```
( 2.5.4.12
 NAME 'title'
 DESC 'Standard LDAP attribute type'
 SUP name
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 2256' )
```

Description Identifies the title of a person in the organization.

Syntax Directory String, multi-valued.

Examples

```
title: Senior QC Inspector
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name uid, userid – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.1
 NAME ( 'uid' 'userid' )
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 1274' )
```

Description Identifies the entry's userid (usually the logon ID). Abbreviation: uid.

Syntax Directory String, multi-valued.

Examples

```
userid: banderson
```

or

```
uid: banderson
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name uidNumber – Standard LDAP attribute type

Synopsis

```
( 1.3.6.1.1.1.1.0
 NAME 'uidNumber'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 SINGLE-VALUE
 X-ORIGIN 'RFC 2307' )
```

Description Related to the `/etc/shadow` file, this attribute specifies the user's login ID.

Syntax Integer, single-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name uniqueIdentifier – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.44
 NAME 'uniqueIdentifier'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 1274' )
```

Description Identifies a specific item used to distinguish between two entries when a distinguished name has been reused. This attribute is intended to detect an instance of a reference to a distinguished name that has been deleted. This attribute is assigned by the server.

Syntax Directory String, multi-valued.

Examples

```
uniqueIdentifier: 17B
```

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name uniqueMember – Standard LDAP attribute type

Synopsis

```
( 2.5.4.50
 NAME 'uniqueMember'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-ORIGIN 'RFC 2256' )
```

Description Identifies a group of names associated with an entry where each name was given a uniqueIdentifier to ensure its uniqueness. A value for the uniqueMember attribute is a DN followed by an optional hash (#) and uniqueIdentifier.

Syntax DN, multi-valued.

Examples

```
uniqueMember: cn=John Doe, dc=example, dc=com #17
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name updatedByDocument – Pilot attribute type

Synopsis

```
( 0.9.2342.19200300.102.1.6
 NAME 'updatedByDocument'
 DESC 'Pilot attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-ORIGIN 'Internet White Pages Pilot' )
```

Description Contains the distinguished name of a document that is an updated version of the document entry.

Syntax DN, multi-valued.

Examples

```
updatedByDocument: cn=Doc Version 2, ou=Document Library,dc=example, dc=com
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Internet White Pages Pilot

Name

updatesDocument – Pilot attribute type

Synopsis

```
( 0.9.2342.19200300.102.1.5
 NAME ’updatesDocument’
 DESC ’Pilot attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-ORIGIN ’Internet White Pages Pilot’ )
```

Description

Contains the distinguished name of a document for which this document is an updated version.

Syntax

DN, multi-valued.

Examples

```
updatesDocument: cn=Doc Version 1, ou=Document Library,dc=example, dc=com
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Internet White Pages Pilot

Name userCertificate – Standard LDAP attribute type

Synopsis

```
( 2.5.4.36
 NAME 'userCertificate'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
 X-ORIGIN 'RFC 2256' )
```

Description This attribute contains a certificate. It is to be stored and requested in the binary form, as `userCertificate;binary`.

Syntax Binary, multi-valued.

Examples

```
userCertificate;binary:: AAAAAA==
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name userClass – Standard LDAP attribute type

Synopsis

```
( 0.9.2342.19200300.100.1.8
 NAME 'userClass'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-ORIGIN 'RFC 1274' )
```

Description Specifies a category of computer user. The semantics of this attribute are arbitrary. The organizationalStatus attribute makes no distinction between computer users and others users and may be more applicable.

Syntax Directory String, multi-valued.

Examples `userClass: intern`

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name userPassword – Standard LDAP attribute type

Synopsis

```
( 2.5.4.35
 NAME ’userPassword’
 DESC ’Standard LDAP attribute type’
 EQUALITY octetStringMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.40{128}
 X-ORIGIN ’RFC 2256’ )
```

Description Identifies the entry's password and encryption method in the following format:

```
{encryption method}encrypted password
```

Transfer of clear text passwords is strongly discouraged where the underlying transport service cannot guarantee confidentiality. Transfer of clear text may result in disclosure of the password to unauthorized parties.

Syntax Octet String, multi-valued.

Examples

```
userPassword: {ssha}9LsFG7RT+dFnPErwSfxDlaQTn6dbIFGklMNFRr==
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name userPKCS12 – inetOrgPerson attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.216
 NAME 'userPKCS12'
 DESC 'inetOrgPerson attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
 X-ORIGIN 'inetOrgPerson Internet Draft' )
```

Description This attribute provides a format for the exchange of personal identity information. The attribute is to be stored and requested in binary form, as `userPKCS12;binary`. The attribute values are PFX PDUs stored as binary data.

Syntax Binary, multi-valued.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, RFC 2798

Name userSMIMECertificate – inetOrgPerson attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.40
 NAME ’userSMIMECertificate’
 DESC ’inetOrgPerson attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
 X-ORIGIN ’inetOrgPerson Internet Draft’ )
```

Description Used by Netscape Communicator for S/MIME. This attribute is to be stored and requested in the binary form, as `userSMIMECertificate;binary`.

Syntax Binary, multi-valued.

Examples

```
userSMIMECertificate;binary:: AAAAAA==
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, RFC 2798

Name vendorName – LDAP attribute type

Synopsis

```
( 1.3.6.1.1.4
 NAME 'vendorName'
 EQUALITY 1.3.6.1.4.1.1466.109.114.1
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 NO-USER-MODIFICATION
 USAGE dsaOperation
 X-ORIGIN 'RFC 3045' )
```

Description Represents the name of the LDAP server implementer. This attribute must not be used by client applications to gather information related to supported features of the LDAP implementation.

Syntax Directory String, single-valued.

Usage Operational attribute used by a Directory Server instance; returned in `ldapsearch` only when specifically requested.

The value of this attribute may only be modified by the server.

Examples

```
vendorName: Sun Microsystems, Inc.
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 3045

Name vendorVersion – LDAP attribute type

Synopsis

```
( 1.3.6.1.1.5
 NAME ’vendorVersion’
 EQUALITY 1.3.6.1.4.1.1466.109.114.1
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE
 NO-USER-MODIFICATION
 USAGE dsaOperation
 X-ORIGIN ’RFC 3045’ )
```

Description Represents the version of the LDAP server implementation. This attribute must not be used by client applications to gather information related to supported features of the LDAP implementation.

Syntax Directory String, single-valued.

Usage Operational attribute used by a Directory Server instance; returned in `ldapsearch` only when specifically requested.

The value of this attribute may only be modified by the server.

Examples

```
vendorVersion: v6
```

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 3045

Name vlvBase – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.207
 NAME 'vlvBase'
 DESC 'Sun ONE defined attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Defines the base DN of a VLV search.

Syntax DN, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name vlvEnabled – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.213
 NAME 'vlvEnabled'
 DESC 'Sun ONE defined attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Used by the server to signal whether the index is available or unavailable. When VLV indexes are created offline, new `vlvSearch` entries are enabled when the indexes are rebuilt. VLV indexes can also be created while the server is running in read-only mode. This attribute is read-only and single-valued.

Syntax Integer, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name vlvFilter – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.209
 NAME ’vlvFilter’
 DESC ’Sun ONE defined attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 X-DS-USE ’internal’
 X-ORIGIN ’Sun ONE Directory Server’ )
```

Description Defines the filter for a VLV search.

Syntax IA5 String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name vlvScope – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.208
 NAME ’vlvScope’
 DESC ’Sun ONE defined attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 X-DS-USE ’internal’
 X-ORIGIN ’Sun ONE Directory Server’ )
```

Description Defines the scope of a VLV search.

Syntax Integer, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name vlvSort – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.210
 NAME 'vlvSort'
 DESC 'Sun ONE defined attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Defines the sort specification for a VLV search. Consists of a list of comma-delimited attribute names. A minus sign is used to denote a reverse sort. The example below will result in a sort by `uid`, then by reverse common name.

Syntax Directory String, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name vlvUses – Sun ONE defined attribute type

Synopsis

```
( 2.16.840.1.113730.3.1.219
 NAME ’vlvUses’
 DESC ’Sun ONE defined attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 X-DS-USE ’internal’
 X-ORIGIN ’Sun ONE Directory Server’ )
```

Description This read-only attribute displays the number of times the VLV index was used. The value is reset when the server is restarted.

Syntax Integer, multi-valued.

Usage Attribute specific to this Directory Server instance and version of the schema.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name x121Address – Standard LDAP attribute type

Synopsis

```
( 2.5.4.24
 NAME 'x121Address'
 DESC 'Standard LDAP attribute type'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
 X-ORIGIN 'RFC 2256' )
```

Description Defines the X.121 address of a person.

Syntax IA5 String, multi-valued.

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name

x500UniqueIdentifier – Standard LDAP attribute type

Synopsis

```
( 2.5.4.45
 NAME ’x500UniqueIdentifier’
 DESC ’Standard LDAP attribute type’
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
 X-ORIGIN ’RFC 2256’ )
```

Description

Reserved for future use. A binary method of identification useful for differentiating objects when a distinguished name has been reused.

Syntax

Binary, multi-valued.

Examples

```
x500UniqueIdentifier: 17B
```

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

REFERENCE

LDAP Schema Object Classes

Name account – Standard LDAP objectclass

Synopsis

```
( 0.9.2342.19200300.100.4.5
 NAME 'account'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST uid
 MAY ( description $
  host $
  l $
  o $
  ou $
  seeAlso )
 X-ORIGIN 'RFC 1274' )
```

Description Used to define entries representing computer accounts.

Origin This object class is defined by RFC 1274.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

uid(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat), host(5dsat), l(5dsat), o(5dsat), ou(5dsat), seeAlso(5dsat)

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name alias – Standard LDAP objectclass

Synopsis

```
( 2.5.6.1
 NAME 'alias'
 DESC 'Standard LDAP objectclass'
 SUP top
 ABSTRACT
 MUST aliasedObjectName
 X-ORIGIN 'RFC 2256' )
```

Description Abstract object class, used to point to other entries in the directory tree.

Note that alias dereferencing is not supported in Sun Java System Directory Server.

Origin This object class is defined by RFC 2256.

Type Abstract object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

aliasedObjectName(5dsat)

Allowed Attributes Entries of this object class have no optional attribute types other than those inherited from the superior(s).

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name applicationEntity – Standard LDAP objectclass

Synopsis

```
( 2.5.6.12
 NAME 'applicationEntity'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST ( presentationAddress $
  cn )
 MAY ( description $
  l $
  o $
  ou $
  seeAlso $
  supportedApplicationContext )
 X-ORIGIN 'RFC 2256' )
```

Description Used to describe entries representing applications.

Origin This object class is defined by RFC 2256.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat), presentationAddress(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat), l(5dsat), o(5dsat), ou(5dsat), seeAlso(5dsat), supportedApplicationContext(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name bootableDevice – Standard LDAP objectclass

Synopsis

```
( 1.3.6.1.1.1.2.12
 NAME 'bootableDevice'
 DESC 'Standard LDAP objectclass'
 SUP top
 AUXILIARY
 MAY ( bootFile $
  bootParameter $
  cn )
 X-ORIGIN 'RFC 2307' )
```

Description Auxiliary object class that specifies a device with boot parameters.

Origin This object class is defined by RFC 2307.

Type Auxiliary object class

Superior top(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

bootFile(5dsat), bootParameter(5dsat), cn(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name

changeLogEntry – LDAP changelog objectclass

Synopsis

```
( 2.16.840.1.113730.3.2.1
 NAME 'changeLogEntry'
 DESC 'LDAP changelog objectclass'
 SUP top
 STRUCTURAL
 MUST ( targetDn $
  changeTime $
  changeNumber $
  changeType )
 MAY ( changes $
  newRdn $
  deleteOldRdn $
  newSuperior )
 X-ORIGIN 'Changelog Internet Draft' )
```

Description

Internal object class, used to represent changes made to Directory Server. You can configure Directory Server to maintain a change log that is compatible with the change log implemented in earlier versions of Directory Server by enabling the Retro Changelog plug-in. Each entry in the change log has the object class changeLogEntry.

Origin

This object class is defined by Changelog Internet Draft.

Type

Structural object class

Superior

top(5dsoc)

Required Attributes

Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

changeNumber(5dsat), changeTime(5dsat), changeType(5dsat), targetDn(5dsat)

Allowed Attributes

Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

changeHasReplFixupOp(5dsat), changeIsReplFixupOp(5dsat), changes(5dsat), deleteOldRdn(5dsat), deletedEntryAttrs(5dsat), newRdn(5dsat), newSuperior(5dsat), replicaIdentifier(5dsat), replicationCSN(5dsat), targetUniqueId(5dsat)

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Changelog Internet Draft

Name cosClassicDefinition – Sun ONE defined objectclass

Synopsis

```
( 2.16.840.1.113730.3.2.100
 NAME 'cosClassicDefinition'
 DESC 'Sun ONE defined objectclass'
 SUP cosSuperDefinition
 STRUCTURAL
 MAY ( costemplatedn $
  cosspecifier )
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Identifies the template entry using both the template entry's DN (as specified in the `cosTemplateDn` attribute) and the value of one of the target entry's attributes (as specified in the `cosSpecifier` attribute).

Origin This object class is defined by Sun Java System Directory Server.

Type Structural object class

Superior cosSuperDefinition(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

cosspecifier(5dsat), costemplatedn(5dsat)

Usage Configuration object specific to this Directory Server instance, not replicated.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name cosDefinition – Sun ONE defined objectclass

Synopsis

```
( 2.16.840.1.113730.3.2.84
 NAME 'cosDefinition'
 DESC 'Sun ONE defined objectclass'
 SUP top
 STRUCTURAL
 MAY ( costargettree $
  costemplatedn $
  cosspecifier $
  cosAttribute $
  aci $
  cn $
  uid )
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Defines the Class of Service you are using. This object class is supported for compatibility with an earlier version of the Directory Server CoS Plugin. Its use is deprecated.

Origin This object class is defined by Sun Java System Directory Server.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

aci(5dsat), cn(5dsat), cosAttribute(5dsat), cosspecifier(5dsat), costargettree(5dsat), costemplatedn(5dsat), uid(5dsat)

Usage Configuration object specific to this Directory Server instance, not replicated.

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name cosIndirectDefinition – Sun ONE defined objectclass

Synopsis

```
( 2.16.840.1.113730.3.2.102
 NAME 'cosIndirectDefinition'
 DESC 'Sun ONE defined objectclass'
 SUP cosSuperDefinition
 STRUCTURAL
 MAY cosIndirectSpecifier
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Identifies the template entry using the value of one of the target entry's attributes. The attribute of the target entry is specified in the `cosIndirectSpecifier` attribute.

Origin This object class is defined by Sun Java System Directory Server.

Type Structural object class

Superior cosSuperDefinition(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

cosIndirectSpecifier(5dsat)

Usage Configuration object specific to this Directory Server instance, not replicated.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name cosPointerDefinition – Sun ONE defined objectclass

Synopsis

```
( 2.16.840.1.113730.3.2.101
 NAME 'cosPointerDefinition'
 DESC 'Sun ONE defined objectclass'
 SUP cosSuperDefinition
 STRUCTURAL
 MAY costemplatedn
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Identifies the template entry associated with the CoS definition using the template entry's DN value. The DN of the template entry is specified in the `cosTemplateDn` attribute.

Origin This object class is defined by Sun Java System Directory Server.

Type Structural object class

Superior cosSuperDefinition(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

costemplatedn(5dsat)

Usage Configuration object specific to this Directory Server instance, not replicated.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name cosSuperDefinition – Sun ONE defined objectclass

Synopsis

```
( 2.16.840.1.113730.3.2.99
 NAME 'cosSuperDefinition'
 DESC 'Sun ONE defined objectclass'
 SUP ldapSubEntry
 STRUCTURAL
 MUST cosAttribute
 MAY description
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description All CoS definition object classes inherit from the `cosSuperDefinition` object class.

Origin This object class is defined by Sun Java System Directory Server.

Type Structural object class

Superior ldapSubEntry(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cosAttribute(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat)

Usage Configuration object specific to this Directory Server instance, not replicated.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name costemplate – Sun ONE defined objectclass

Synopsis

```
( 2.16.840.1.113730.3.2.128
 NAME 'costemplate'
 DESC 'Sun ONE defined objectclass'
 SUP top
 STRUCTURAL
 MAY ( cn $
  cosPriority )
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Contains a list of the shared attribute values.

Origin This object class is defined by Sun Java System Directory Server.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

cn(5dsat), cosPriority(5dsat)

Usage Configuration object specific to this Directory Server instance, not replicated.

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name country – Standard LDAP objectclass

Synopsis

```
( 2.5.6.2
 NAME 'country'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST c
 MAY ( searchGuide $
  description )
 X-ORIGIN 'RFC 2256' )
```

Description Contains the two-character code representing country names, as defined in ISO-3166.

Origin This object class is defined by RFC 2256.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

c(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat), searchGuide(5dsat)

Examples

```
countryName: IE
```

or

```
c: IE
```

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name

dcObject – Standard LDAP objectclass

Synopsis

```
( 1.3.6.1.4.1.1466.344
 NAME 'dcObject'
 DESC 'Standard LDAP objectclass'
 SUP top
 AUXILIARY
 MUST dc
 X-ORIGIN 'RFC 2247' )
```

Description

This auxiliary object class defines a domain component, such as a network domain that is associated with the entry. This object class is defined as auxiliary because it is commonly used in combination with another object class, such as `organization`, `organizationUnit`, or `locality`.

Origin

This object class is defined by RFC 2247.

Type

Auxiliary object class

Superior

top(5dsoc)

Required Attributes

Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

dc(5dsat)

Allowed Attributes

Entries of this object class have no optional attribute types other than those inherited from the superior(s).

Examples

```
dn: ou=Engineering,dc=example,dc=com
objectClass: top
objectClass: organizationalUnit
objectClass: dcObject
ou: Engineering
dc: example
```

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2247

Name device – Standard LDAP objectclass

Synopsis

```
( 2.5.6.14
 NAME 'device'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST cn
 MAY ( description $
  l $
  o $
  ou $
  owner $
  seeAlso $
  serialNumber )
 X-ORIGIN 'RFC 2256' )
```

Description Used to store information about network devices, such as printers, in the directory.

Origin This object class is defined by RFC 2256.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat), l(5dsat), o(5dsat), ou(5dsat), owner(5dsat), seeAlso(5dsat), serialNumber(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name document – Standard LDAP objectclass

Synopsis

```
( 0.9.2342.19200300.100.4.6
 NAME 'document'
 DESC 'Standard LDAP objectclass'
 SUP pilotObject
 STRUCTURAL
 MUST documentIdentifier
 MAY ( abstract $
  authorCn $
  authorSn $
  cn $
  description $
  documentAuthor $
  documentLocation $
  documentPublisher $
  documentStore $
  documentTitle $
  documentVersion $
  keyWords $
  l $
  o $
  obsoletedByDocument $
  obsoletesDocument $
  ou $
  seeAlso $
  subject $
  updatedByDocument $
  updatesDocument )
 X-ORIGIN 'RFC 1274' )
```

Description Used to define entries that represent documents in the directory.

Origin This object class is defined by RFC 1274.

Type Structural object class

Superior pilotObject(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

documentIdentifier(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

abstract(5dsat), authorCn(5dsat), authorSn(5dsat), cn(5dsat), description(5dsat), documentAuthor(5dsat), documentLocation(5dsat), documentPublisher(5dsat), documentStore(5dsat), documentTitle(5dsat), documentVersion(5dsat), keyWords(5dsat), l(5dsat), o(5dsat), obsoletedByDocument(5dsat), obsoletesDocument(5dsat), ou(5dsat), seeAlso(5dsat), subject(5dsat), updatedByDocument(5dsat), updatesDocument(5dsat)

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name documentSeries – Standard LDAP objectclass

Synopsis

```
( 0.9.2342.19200300.100.4.9
 NAME 'documentSeries'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST cn
 MAY ( description $
  l $
  o $
  ou $
  seeAlso $
  telephoneNumber )
 X-ORIGIN 'RFC 1274' )
```

Description Used to define an entry that represents a series of documents.

Origin This object class is defined by RFC 1274.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat), l(5dsat), o(5dsat), ou(5dsat), seeAlso(5dsat), telephoneNumber(5dsat)

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name domain – Standard LDAP objectclass

Synopsis

```
( 0.9.2342.19200300.100.4.13
 NAME 'domain'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST dc
 MAY ( associatedName $
  businessCategory $
  description $
  destinationIndicator $
  facsimileTelephoneNumber $
  internationaliSDNNumber $
  l $
  o $
  physicalDeliveryOfficeName $
  postOfficeBox $
  postalAddress $
  postalCode $
  preferredDeliveryMethod $
  registeredAddress $
  searchGuide $
  seeAlso $
  st $
  street $
  telephoneNumber $
  teletexTerminalIdentifier $
  telexNumber $
  userPassword $
  x121Address )
 X-ORIGIN 'RFC 2247' )
```

Description Used to represent Internet Domains (for example, `example.com`). The `domainComponent` attribute should be used for naming entries of this object class.

The `domain` object class can only be used with an entry that does not correspond to an organization, organizational unit, or other type of object for which an object class has been defined. The domain object class requires that the `domainComponent` attribute be present, and allows several other attributes to be present in the entry. These allowed attributes are used to describe the object represented by the domain, and may also be useful when searching.

Origin This object class is defined by RFC 2247.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

dc(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

associatedName(5dsat), businessCategory(5dsat), description(5dsat), destinationIndicator(5dsat), facsimileTelephoneNumber(5dsat), internationaliSDNNumber(5dsat), l(5dsat), o(5dsat), physicalDeliveryOfficeName(5dsat), postOfficeBox(5dsat), postalAddress(5dsat), postalCode(5dsat), preferredDeliveryMethod(5dsat), registeredAddress(5dsat), searchGuide(5dsat), seeAlso(5dsat), st(5dsat), street(5dsat), telephoneNumber(5dsat), teletexTerminalIdentifier(5dsat), telexNumber(5dsat), userPassword(5dsat), x121Address(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2247

Name domainRelatedObject – Standard LDAP objectclass

Synopsis

```
( 0.9.2342.19200300.100.4.17
 NAME 'domainRelatedObject'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST associatedDomain
 X-ORIGIN 'RFC 1274' )
```

Description Used to define entries that represent DNS/NRS domains that are equivalent to an X.500 domain, for example, an organization or organizational unit.

Origin This object class is defined by RFC 1274.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

associatedDomain(5dsat)

Allowed Attributes Entries of this object class have no optional attribute types other than those inherited from the superior(s).

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name dSA – Standard LDAP objectclass

Synopsis

```
( 2.5.6.13
 NAME 'dSA'
 DESC 'Standard LDAP objectclass'
 SUP applicationEntity
 STRUCTURAL
 MAY knowledgeInformation
 X-ORIGIN 'RFC 2256' )
```

Description Used to define entries representing Directory Server Agents.

Origin This object class is defined by RFC 2256.

Type Structural object class

Superior applicationEntity(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

knowledgeInformation(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name dsSaslConfig – Sun ONE defined objectclass

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.2.116
 NAME 'dsSaslConfig'
 DESC 'Sun ONE defined objectclass'
 SUP top
 STRUCTURAL
 MUST ( dsSaslPluginsEnable $
  dsSaslPluginsPath )
 MAY ( dsSaslMinSSF $
  dsSaslMaxSSF $
  dsSaslMaxBufSize )
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Used as the object class for the SASL configuration entry. The `dse.ldif` entry governing SASL configuration has DN `cn=SASL, cn=security, cn=config`.

Origin This object class is defined by Sun Java System Directory Server.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

dsSaslPluginsEnable(5dsat), dsSaslPluginsPath(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

dsSaslMaxBufSize(5dsat), dsSaslMaxSSF(5dsat), dsSaslMinSSF(5dsat)

Usage Configuration object specific to this Directory Server instance, not replicated.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name

extensibleObject – LDAPv3 extensible object

Synopsis

```
( 1.3.6.1.4.1.1466.101.120.111
 NAME 'extensibleObject'
 DESC 'LDAPv3 extensible object'
 SUP top
 AUXILIARY
 X-ORIGIN 'RFC 2252' )
```

Description

Auxiliary object class which, when present in an entry, permits the entry to optionally hold any attribute. The allowed attribute list of this class is implicitly the set of all attributes known to the server.

In general it is better to use a more restrictive object class when designing schema for your deployment, as the server can do very little checking for `extensibleObject`. In particular, a good practice for new applications is to add specific auxiliary object classes where specific new attribute sets are needed, rather than to leave everything wide open and undecided with `extensibleObject`.

Origin

This object class is defined by RFC 2252.

Type

Auxiliary object class

Superior

top(5dsoc)

Required Attributes

Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes

Entries of this object class have no optional attribute types other than those inherited from the superior(s).

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2252

Name friendlyCountry – Standard LDAP objectclass

Synopsis

```
( 0.9.2342.19200300.100.4.18
 NAME 'friendlyCountry'
 DESC 'Standard LDAP objectclass'
 SUP country
 STRUCTURAL
 MUST co
 X-ORIGIN 'RFC 1274' )
```

Description Used to define country entries in the directory tree. This object class is used to allow more user-friendly country names than those allowed by the country object class.

Origin This object class is defined by RFC 1274.

Type Structural object class

Superior country(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

co(5dsat)

Allowed Attributes Entries of this object class have no optional attribute types other than those inherited from the superior(s).

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name groupOfCertificates – Sun ONE defined objectclass

Synopsis

```
( 2.16.840.1.113730.3.2.31
 NAME 'groupOfCertificates'
 DESC 'Sun ONE defined objectclass'
 SUP top
 STRUCTURAL
 MUST cn
 MAY ( memberCertificateDescription $
  businessCategory $
  description $
  o $
  ou $
  owner $
  seeAlso )
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Used to describe a set of X.509 certificates. Any certificate that matches one of the memberCertificateDescription values is considered a member of the group.

Origin This object class is defined by Sun Java System Directory Server.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

businessCategory(5dsat), description(5dsat), memberCertificateDescription(5dsat), o(5dsat), ou(5dsat), owner(5dsat), seeAlso(5dsat)

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name groupOfNames – Standard LDAP objectclass

Synopsis

```
( 2.5.6.9
 NAME 'groupOfNames'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST cn
 MAY ( member $
  businessCategory $
  description $
  o $
  ou $
  owner $
  seeAlso )
 X-ORIGIN 'RFC 2256' )
```

Description Used to define entries for a group of names.

Origin This object class is defined by RFC 2256.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

businessCategory(5dsat), description(5dsat), member(5dsat), o(5dsat), ou(5dsat), owner(5dsat), seeAlso(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name groupOfUniqueNames – Standard LDAP objectclass

Synopsis

```
( 2.5.6.17
 NAME 'groupOfUniqueNames'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST cn
 MAY ( uniqueMember $
  businessCategory $
  description $
  o $
  ou $
  owner $
  seeAlso )
 X-ORIGIN 'RFC 2256' )
```

Description Used to define entries for a group of unique names.

Origin This object class is defined by RFC 2256.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

businessCategory(5dsat), description(5dsat), o(5dsat), ou(5dsat), owner(5dsat), seeAlso(5dsat), uniqueMember(5dsat)

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name groupOfURLs – Sun ONE defined objectclass

Synopsis

```
( 2.16.840.1.113730.3.2.33
 NAME 'groupOfURLs'
 DESC 'Sun ONE defined objectclass'
 SUP top
 STRUCTURAL
 MUST cn
 MAY ( memberURL $
  businessCategory $
  description $
  o $
  ou $
  owner $
  seeAlso )
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description An auxiliary object class of groupOfUniqueNames or groupOfNames. The group consists of a list of labeled URLs.

Origin This object class is defined by Sun Java System Directory Server.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

businessCategory(5dsat), description(5dsat), memberURL(5dsat), o(5dsat), ou(5dsat), owner(5dsat), seeAlso(5dsat)

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name ieee802Device – Standard LDAP objectclass

Synopsis

```
( 1.3.6.1.1.1.2.11
 NAME 'ieee802Device'
 DESC 'Standard LDAP objectclass'
 SUP top
 AUXILIARY
 MAY ( macAddress $
  cn )
 X-ORIGIN 'RFC 2307' )
```

Description Auxiliary object class, specifying a device with a MAC address.

Origin This object class is defined by RFC 2307.

Type Auxiliary object class

Superior top(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

cn(5dsat), macAddress(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name inetOrgPerson – Internet extended organizational person objectclass

Synopsis

```
( 2.16.840.1.113730.3.2.2
 NAME 'inetOrgPerson'
 DESC 'Internet extended organizational person objectclass'
 SUP organizationalPerson
 STRUCTURAL
 MAY ( audio $
  businessCategory $
  carLicense $
  departmentNumber $
  displayName $
  employeeType $
  employeeNumber $
  givenName $
  homePhone $
  homePostalAddress $
  initials $
  jpegPhoto $
  labeledUri $
  manager $
  mobile $
  pager $
  photo $
  preferredLanguage $
  mail $
  o $
  roomNumber $
  secretary $
  uid $
  x500UniqueIdentifier $
  userCertificate $
  userSMIMECertificate $
  userPKCS12 )
 X-ORIGIN 'inetOrgPerson Internet Draft' )
```

Description Used to define entries representing people in an organization's enterprise network.

Origin This object class is defined by RFC 2798.

Type Structural object class

Superior organizationalPerson(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

audio(5dsat), businessCategory(5dsat), carLicense(5dsat), departmentNumber(5dsat), displayName(5dsat), employeeNumber(5dsat), employeeType(5dsat), givenName(5dsat), homePhone(5dsat), homePostalAddress(5dsat), initials(5dsat), jpegPhoto(5dsat), labeledUri(5dsat), mail(5dsat), manager(5dsat), mobile(5dsat), o(5dsat), pager(5dsat), photo(5dsat), preferredLanguage(5dsat), roomNumber(5dsat), secretary(5dsat), uid(5dsat), userCertificate(5dsat), userPKCS12(5dsat), userSMIMECertificate(5dsat), x500UniqueIdentifier(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, RFC 2798

Name

ipHost – Standard LDAP objectclass

Synopsis

```
( 1.3.6.1.1.1.2.6
 NAME 'ipHost'
 DESC 'Standard LDAP objectclass'
 SUP top
 AUXILIARY
 MUST ( ipHostNumber $
  cn )
 MAY ( manager $
  description $
  l )
 X-ORIGIN 'RFC 2307' )
```

Description

Auxiliary object class, specifying an abstraction of a host, an IP device. The distinguished value of the cn attribute denotes the canonical name of the host.

Origin

This object class is defined by RFC 2307.

Type

Auxiliary object class

Superior

top(5dsoc)

Required Attributes

Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat), ipHostNumber(5dsat)

Allowed Attributes

Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat), l(5dsat), manager(5dsat)

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name

ipNetwork – Standard LDAP objectclass

Synopsis

```
( 1.3.6.1.1.1.2.7
 NAME 'ipNetwork'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST ( ipNetworkNumber $
  cn )
 MAY ( ipNetmaskNumber $
  manager $
  l $
  description )
 X-ORIGIN 'RFC 2307' )
```

Description

Auxiliary object class, specifying an abstraction of a host, an IP device. The distinguished value of the cn attribute denotes the canonical name of the host.

Origin

This object class is defined by RFC 2307.

Type

Structural object class

Superior

top(5dsoc)

Required Attributes

Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat), ipNetworkNumber(5dsat)

Allowed Attributes

Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat), ipNetmaskNumber(5dsat), l(5dsat), manager(5dsat)

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name ipProtocol – Standard LDAP objectclass

Synopsis

```
( 1.3.6.1.1.1.2.4
 NAME 'ipProtocol'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST ( cn $
  ipProtocolNumber )
 MAY description
 X-ORIGIN 'RFC 2307' )
```

Description Abstraction of an IP protocol. This object class maps a protocol number to one or more names. The distinguished value of the cn attribute denotes the protocol's canonical name.

Origin This object class is defined by RFC 2307.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat), ipProtocolNumber(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat)

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name ipService – Standard LDAP objectclass

Synopsis

```
( 1.3.6.1.1.1.2.3
 NAME 'ipService'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST ( cn $
  ipServicePort $
  ipServiceProtocol )
 MAY description
 X-ORIGIN 'RFC 2307' )
```

Description Abstraction of an Internet Protocol service. This object class maps an IP port and protocol (such as TCP or UDP) to one or more names. The distinguished value of the cn attribute denotes the service's canonical name.

Origin This object class is defined by RFC 2307.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat), ipServicePort(5dsat), ipServiceProtocol(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat)

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name javaContainer – Container for a Java object

Synopsis

```
( 1.3.6.1.4.1.42.2.27.4.2.1
 NAME 'javaContainer'
 DESC 'Container for a Java object'
 SUP top
 STRUCTURAL
 MUST cn
 X-ORIGIN 'RFC 2713' )
```

Description Represents a container for a Java object.

Origin This object class is defined by RFC 2713.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat)

Allowed Attributes Entries of this object class have no optional attribute types other than those inherited from the superior(s).

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2713

Name javaMarshalledObject – Java marshalled object

Synopsis

```
( 1.3.6.1.4.1.42.2.27.4.2.8
 NAME 'javaMarshalledObject'
 DESC 'Java marshalled object'
 SUP javaObject
 AUXILIARY
 MUST javaSerializedData
 X-ORIGIN 'RFC 2713' )
```

Description Auxiliary object class that represents a Java marshalled object. It must be mixed with a structural object class.

Origin This object class is defined by RFC 2713.

Type Auxiliary object class

Superior javaObject(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

javaSerializedData(5dsat)

Allowed Attributes Entries of this object class have no optional attribute types other than those inherited from the superior(s).

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2713

Name javaNamingReference – JNDI reference

Synopsis

```
( 1.3.6.1.4.1.42.2.27.4.2.7
 NAME 'javaNamingReference'
 DESC 'JNDI reference'
 SUP javaObject
 AUXILIARY
 MAY ( javaReferenceAddress $
  javaFactory )
 X-ORIGIN 'RFC 2713' )
```

Description Auxiliary object class that represents a JNDI reference. It must be mixed in with a structural object class.

Origin This object class is defined by RFC 2713.

Type Auxiliary object class

Superior javaObject(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

javaFactory(5dsat), javaReferenceAddress(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2713

Name javaObject – Java object representation

Synopsis

```
( 1.3.6.1.4.1.42.2.27.4.2.4
 NAME ’javaObject’
 DESC ’Java object representation’
 SUP top
 ABSTRACT
 MUST javaClassName
 MAY ( javaClassNames $
  javaCodebase $
  javaDoc $
  description )
 X-ORIGIN ’RFC 2713’ )
```

Description Abstract object class that represents a Java object.

Origin This object class is defined by RFC 2713.

Type Abstract object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

javaClassName(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat), javaClassNames(5dsat), javaCodebase(5dsat), javaDoc(5dsat)

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2713

Name javaSerializedObject – Java serialized object

Synopsis

```
( 1.3.6.1.4.1.42.2.27.4.2.5
 NAME 'javaSerializedObject'
 DESC 'Java serialized object'
 SUP javaObject
 AUXILIARY
 MUST javaSerializedData
 X-ORIGIN 'RFC 2713' )
```

Description Auxiliary object class that represents a Java serialized object. It must be mixed in with a structural object class.

Origin This object class is defined by RFC 2713.

Type Auxiliary object class

Superior javaObject(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

javaSerializedData(5dsat)

Allowed Attributes Entries of this object class have no optional attribute types other than those inherited from the superior(s).

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2713

Name labeledURIObject – object that contains the URI attribute type

Synopsis

```
( 1.3.6.1.4.1.250.3.15
 NAME 'labeledURIObject'
 DESC 'object that contains the URI attribute type'
 SUP top
 AUXILIARY
 MAY labeledUri
 X-ORIGIN 'RFC 2079' )
```

Description Auxiliary object class that can be added to existing directory objects to allow for inclusion of URI values. This approach does not preclude including the `labeledURI` attribute type directly in other object classes as appropriate.

Origin This object class is defined by RFC 2079.

Type Auxiliary object class

Superior top(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

labeledUri(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2079

Name ldapSubEntry – LDAP Subentry class, version 1

Synopsis

```
( 2.16.840.1.113719.2.142.6.1.1
 NAME 'ldapSubEntry'
 DESC 'LDAP Subentry class, version 1'
 SUP top
 STRUCTURAL
 MAY cn
 X-DS-USE 'internal'
 X-ORIGIN 'LDAP Subentry Internet Draft' )
```

Description This structural object class may be used to indicate operations and management related entries in the directory, called LDAP Subentries.

Origin This object class is defined by LDAP Subentry Internet Draft.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

cn(5dsat)

Usage Configuration object specific to this Directory Server instance, not replicated.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, LDAP Subentry Internet Draft

Name

locality – Standard LDAP attribute type

Synopsis

```
( 2.5.6.3
 NAME 'locality'
 DESC 'Standard LDAP attribute type'
 SUP top
 STRUCTURAL
 MAY ( description $
  l $
  searchGuide $
  seeAlso $
  st $
  street )
 X-ORIGIN 'RFC 2256' )
```

Description

Used to define entries that represent localities or geographic areas.

Origin

This object class is defined by RFC 2256.

Type

Structural object class

Superior

top(5dsoc)

Required Attributes

Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes

Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat), l(5dsat), searchGuide(5dsat), seeAlso(5dsat), st(5dsat), street(5dsat)

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name newPilotPerson – Pilot objectclass

Synopsis

```
( 0.9.2342.19200300.100.4.4
 NAME 'newPilotPerson'
 DESC 'Pilot objectclass'
 SUP person
 STRUCTURAL
 MAY ( businessCategory $
  drink $
  homePhone $
  homePostalAddress $
  janetMailbox $
  mail $
  mailPreferenceOption $
  mobile $
  organizationalStatus $
  otherMailbox $
  pager $
  personalSignature $
  personalTitle $
  preferredDeliveryMethod $
  roomNumber $
  secretary $
  textEncodedORAddress $
  uid $
  userClass )
 X-ORIGIN 'Internet White Pages Pilot' )
```

Description Used as a subclass of person, to allow the use of a number of additional attributes to be assigned to entries of the person object class. Inherits cn and sn from the person object class.

Origin This object class is defined by Internet White Pages Pilot.

Type Structural object class

Superior person(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

businessCategory(5dsat), drink(5dsat), homePhone(5dsat), homePostalAddress(5dsat), janetMailbox(5dsat), mail(5dsat), mailPreferenceOption(5dsat), mobile(5dsat), organizationalStatus(5dsat), otherMailbox(5dsat), pager(5dsat), personalSignature(5dsat), personalTitle(5dsat), preferredDeliveryMethod(5dsat), roomNumber(5dsat), secretary(5dsat), textEncodedORAddress(5dsat), uid(5dsat), userClass(5dsat)

Attributes

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Internet White Pages Pilot

Name nisMap – Standard LDAP objectclass

Synopsis

```
( 1.3.6.1.1.1.2.9
 NAME 'nisMap'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST nisMapName
 MAY description
 X-ORIGIN 'RFC 2307' )
```

Description A generic abstraction of a NIS map.

Origin This object class is defined by RFC 2307.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

nisMapName(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name nisNetgroup – Standard LDAP objectclass

Synopsis

```
( 1.3.6.1.1.1.2.8
 NAME 'nisNetgroup'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST cn
 MAY ( nisNetgroupTriple $
  memberNisNetgroup $
  description )
 X-ORIGIN 'RFC 2307' )
```

Description An abstraction of a netgroup. May refer to other netgroups.

Origin This object class is defined by RFC 2307.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat), memberNisNetgroup(5dsat), nisNetgroupTriple(5dsat)

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name nisObject – Standard LDAP objectclass

Synopsis

```
( 1.3.6.1.1.1.2.10
 NAME 'nisObject'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST ( cn $
  nisMapEntry $
  nisMapName )
 MAY description
 X-ORIGIN 'RFC 2307' )
```

Description Defines an entry in a NIS map.

Origin This object class is defined by RFC 2307.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat), nisMapEntry(5dsat), nisMapName(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name nsComplexRoleDefinition – Sun ONE defined objectclass

Synopsis

```
( 2.16.840.1.113730.3.2.95
 NAME 'nsComplexRoleDefinition'
 DESC 'Sun ONE defined objectclass'
 SUP nsRoleDefinition
 STRUCTURAL
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Any role that is not a simple role is, by definition, a complex role.

Origin This object class is defined by Sun Java System Directory Server.

Type Structural object class

Superior nsRoleDefinition(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class have no optional attribute types other than those inherited from the superior(s).

Usage Configuration object specific to this Directory Server instance, not replicated.

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name nsFilteredRoleDefinition – Sun ONE defined objectclass

Synopsis

```
( 2.16.840.1.113730.3.2.97
 NAME 'nsFilteredRoleDefinition'
 DESC 'Sun ONE defined objectclass'
 SUP nsComplexRoleDefinition
 STRUCTURAL
 MUST nsRoleFilter
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Specifies assignment of entries to the role, depending upon the attributes contained by each entry.

Origin This object class is defined by Sun Java System Directory Server.

Type Structural object class

Superior nsComplexRoleDefinition(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

nsRoleFilter(5dsat)

Allowed Attributes Entries of this object class have no optional attribute types other than those inherited from the superior(s).

Usage Configuration object specific to this Directory Server instance, not replicated.

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name nsLicenseUser – Sun ONE defined objectclass

Synopsis

```
( 2.16.840.1.113730.3.2.7
 NAME 'nsLicenseUser'
 DESC 'Sun ONE defined objectclass'
 SUP top
 STRUCTURAL
 MAY ( nsLicensedFor $
  nsLicenseStartTime $
  nsLicenseEndTime )
 X-ORIGIN 'Sun ONE Administration Services' )
```

Description Used to track licenses for servers that are licensed on a per-client basis. `nsLicenseUser` is intended to be used with the `inetOrgPerson` object class. You can manage the contents of this object class through the Users and Groups area of the Administration Server.

Origin This object class is defined by Sun Java System Administration Services.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

nsLicenseEndTime(5dsat), nsLicenseStartTime(5dsat), nsLicensedFor(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name nsManagedRoleDefinition – Sun ONE defined objectclass

Synopsis

```
( 2.16.840.1.113730.3.2.96
 NAME 'nsManagedRoleDefinition'
 DESC 'Sun ONE defined objectclass'
 SUP nsSimpleRoleDefinition
 STRUCTURAL
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Specifies assignment of a role to an explicit, enumerated list of members.

Origin This object class is defined by Sun Java System Directory Server.

Type Structural object class

Superior nsSimpleRoleDefinition(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class have no optional attribute types other than those inherited from the superior(s).

Usage Configuration object specific to this Directory Server instance, not replicated.

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name nsNestedRoleDefinition – Sun ONE defined objectclass

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.2.9
 NAME 'nsNestedRoleDefinition'
 DESC 'Sun ONE defined objectclass'
 SUP nsComplexRoleDefinition
 STRUCTURAL
 MUST nsRoleDN
 MAY nsRoleScopeDn
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Specifies containment of one or more roles of any type within the role.

Origin This object class is defined by Sun Java System Directory Server.

Type Structural object class

Superior nsComplexRoleDefinition(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

nsRoleDN(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

nsRoleScopeDn(5dsat)

Usage Configuration object specific to this Directory Server instance, not replicated.

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name nsRoleDefinition – Sun ONE defined objectclass

Synopsis

```
( 2.16.840.1.113730.3.2.93
 NAME 'nsRoleDefinition'
 DESC 'Sun ONE defined objectclass'
 SUP ldapSubEntry
 STRUCTURAL
 MAY description
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description All role definition object classes inherit from the `nsRoleDefinition` object class.

Origin This object class is defined by Sun Java System Directory Server.

Type Structural object class

Superior ldapSubEntry(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat)

Usage Configuration object specific to this Directory Server instance, not replicated.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name

nsSimpleRoleDefinition – Sun ONE defined objectclass

Synopsis

```
( 2.16.840.1.113730.3.2.94
 NAME 'nsSimpleRoleDefinition'
 DESC 'Sun ONE defined objectclass'
 SUP nsRoleDefinition
 STRUCTURAL
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description

Roles containing this object class are called simple roles because they have a deliberately limited flexibility, which makes it easy to:

- Enumerate the members of a role.
- Determine whether a given entry possesses a particular role.
- Enumerate all the roles possessed by a given entry.
- Assign a particular role to a given entry.
- Remove a particular role from a given entry.

Origin

This object class is defined by Sun Java System Directory Server.

Type

Structural object class

Superior

nsRoleDefinition(5dsoc)

Required Attributes

Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes

Entries of this object class have no optional attribute types other than those inherited from the superior(s).

Usage

Configuration object specific to this Directory Server instance, not replicated.

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name oncRpc – Standard LDAP objectclass

Synopsis

```
( 1.3.6.1.1.1.2.5
 NAME 'oncRpc'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST ( cn $
  oncRpcNumber )
 MAY description
 X-ORIGIN 'RFC 2307' )
```

Description An abstraction of an Open Network Computing (ONC) Remote Procedure Call (RPC) binding. This class maps an ONC RPC number to a name. The distinguished value of the cn attribute denotes the RPC service's canonical name.

Origin This object class is defined by RFC 2307.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat), oncRpcNumber(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat)

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name organization – Standard LDAP objectclass

Synopsis

```
( 2.5.6.4
 NAME 'organization'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST o
 MAY ( businessCategory $
  description $
  destinationIndicator $
  facsimileTelephoneNumber $
  internationaliSDNNumber $
  l $
  physicalDeliveryOfficeName $
  postOfficeBox $
  postalAddress $
  postalCode $
  preferredDeliveryMethod $
  registeredAddress $
  searchGuide $
  seeAlso $
  st $
  street $
  telephoneNumber $
  teletexTerminalIdentifier $
  telexNumber $
  userPassword $
  x121Address )
 X-ORIGIN 'RFC 2256' )
```

Description Used to define entries that represent organizations. An organization is generally assumed to be a large, relatively static grouping within a larger corporation or enterprise.

Origin This object class is defined by RFC 2256.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

o(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

businessCategory(5dsat), description(5dsat), destinationIndicator(5dsat), facsimileTelephoneNumber(5dsat), internationaliSDNNumber(5dsat), l(5dsat), physicalDeliveryOfficeName(5dsat), postOfficeBox(5dsat), postalAddress(5dsat), postalCode(5dsat), preferredDeliveryMethod(5dsat), registeredAddress(5dsat), searchGuide(5dsat), seeAlso(5dsat), st(5dsat), street(5dsat), telephoneNumber(5dsat), teletexTerminalIdentifier(5dsat), telexNumber(5dsat), userPassword(5dsat), x121Address(5dsat)

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name organizationalPerson – Standard LDAP objectclass

Synopsis

```
( 2.5.6.7
 NAME 'organizationalPerson'
 DESC 'Standard LDAP objectclass'
 SUP person
 STRUCTURAL
 MAY ( destinationIndicator $
  facsimileTelephoneNumber $
  internationaliSDNNumber $
  l $
  ou $
  physicalDeliveryOfficeName $
  postOfficeBox $
  postalAddress $
  postalCode $
  preferredDeliveryMethod $
  registeredAddress $
  st $
  street $
  teletexTerminalIdentifier $
  telexNumber $
  title $
  x121Address )
 X-ORIGIN 'RFC 2256' )
```

Description Used to define entries for people employed by or associated with an organization.

Origin This object class is defined by RFC 2256.

Type Structural object class

Superior person(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

destinationIndicator(5dsat), facsimileTelephoneNumber(5dsat), internationaliSDNNumber(5dsat), l(5dsat), ou(5dsat), physicalDeliveryOfficeName(5dsat), postOfficeBox(5dsat), postalAddress(5dsat), postalCode(5dsat), preferredDeliveryMethod(5dsat), registeredAddress(5dsat), st(5dsat), street(5dsat), teletexTerminalIdentifier(5dsat), telexNumber(5dsat), title(5dsat), x121Address(5dsat)

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name organizationalRole – Standard LDAP objectclass

Synopsis

```
( 2.5.6.8
 NAME 'organizationalRole'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST cn
 MAY ( description $
  destinationIndicator $
  facsimileTelephoneNumber $
  internationaliSDNNumber $
  l $
  ou $
  physicalDeliveryOfficeName $
  postOfficeBox $
  postalAddress $
  postalCode $
  preferredDeliveryMethod $
  registeredAddress $
  roleOccupant $
  seeAlso $
  st $
  street $
  telephoneNumber $
  teletexTerminalIdentifier $
  telexNumber $
  x121Address )
 X-ORIGIN 'RFC 2256' )
```

Description Used to define entries that represent roles held by people within an organization.

Origin This object class is defined by RFC 2256.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat), destinationIndicator(5dsat), facsimileTelephoneNumber(5dsat), internationaliSDNNumber(5dsat), l(5dsat), ou(5dsat), physicalDeliveryOfficeName(5dsat), postOfficeBox(5dsat), postalAddress(5dsat), postalCode(5dsat), preferredDeliveryMethod(5dsat), registeredAddress(5dsat), roleOccupant(5dsat), seeAlso(5dsat), st(5dsat), street(5dsat), telephoneNumber(5dsat), teletexTerminalIdentifier(5dsat), telexNumber(5dsat), x121Address(5dsat)

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name organizationalUnit – Standard LDAP objectclass

Synopsis

```
( 2.5.6.5
 NAME 'organizationalUnit'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST ou
 MAY ( businessCategory $
  description $
  destinationIndicator $
  facsimileTelephoneNumber $
  internationaliSDNNumber $
  l $
  physicalDeliveryOfficeName $
  postOfficeBox $
  postalAddress $
  postalCode $
  preferredDeliveryMethod $
  registeredAddress $
  searchGuide $
  seeAlso $
  st $
  street $
  telephoneNumber $
  teletexTerminalIdentifier $
  telexNumber $
  userPassword $
  x121Address )
 X-ORIGIN 'RFC 2256' )
```

Description Used to define entries that represent organizational units. An organizational unit is generally assumed to be a relatively static grouping within a larger organization.

Origin This object class is defined by RFC 2256.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

ou(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

businessCategory(5dsat), description(5dsat), destinationIndicator(5dsat), facsimileTelephoneNumber(5dsat), internationaliSDNNumber(5dsat), l(5dsat), physicalDeliveryOfficeName(5dsat), postOfficeBox(5dsat), postalAddress(5dsat), postalCode(5dsat), preferredDeliveryMethod(5dsat), registeredAddress(5dsat), searchGuide(5dsat), seeAlso(5dsat), st(5dsat), street(5dsat), telephoneNumber(5dsat), teletexTerminalIdentifier(5dsat), telexNumber(5dsat), userPassword(5dsat), x121Address(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name passwordPolicy – Sun ONE defined password policy objectclass

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.2.6
 NAME 'passwordPolicy'
 DESC 'Sun ONE defined password policy objectclass'
 SUP top
 STRUCTURAL
 MUST cn
 MAY ( description $
  passwordMaxAge $
  passwordExp $
  passwordMinLength $
  passwordInHistory $
  passwordChange $
  passwordWarning $
  passwordLockout $
  passwordMaxFailure $
  passwordResetDuration $
  passwordUnlock $
  passwordLockoutDuration $
  passwordCheckSyntax $
  passwordMustChange $
  passwordStorageScheme $
  passwordMinAge $
  passwordResetFailureCount $
  passwordExpireWithoutWarning $
  passwordRootdnMayBypassModsChecks $
  passwordNonRootMayResetUserpwd )
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Entries of this object class defines a 5.x password policy entry that holds configurable password policy attributes.

The 5.x password policy configuration object classes and attribute types are deprecated. Use `pwdPolicy` and `sunPwdPolicy` objects instead.

Origin This object class is defined by Sun Java System Directory Server.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat), passwordChange(5dsat), passwordCheckSyntax(5dsat), passwordExp(5dsat), passwordExpireWithoutWarning(5dsat), passwordInHistory(5dsat), passwordLockout(5dsat), passwordLockoutDuration(5dsat), passwordMaxAge(5dsat), passwordMaxFailure(5dsat), passwordMinAge(5dsat), passwordMinLength(5dsat), passwordMustChange(5dsat), passwordNonRootMayResetUserpwd(5dsat), passwordResetDuration(5dsat), passwordResetFailureCount(5dsat), passwordRootdnMayBypassModsChecks(5dsat), passwordStorageScheme(5dsat), passwordUnlock(5dsat), passwordWarning(5dsat)

Usage Configuration object specific to this Directory Server instance, not replicated.

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Obsolete: Scheduled for removal after this release

Name person – Standard LDAP objectclass

Synopsis

```
( 2.5.6.6
 NAME 'person'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST ( sn $
  cn )
 MAY ( description $
  seeAlso $
  telephoneNumber $
  userPassword )
 X-ORIGIN 'RFC 2256' )
```

Description Used to define entries that generically represent people. This object class is the base class for the organizationalPerson object class.

Origin This object class is defined by RFC 2256.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat), sn(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat), seeAlso(5dsat), telephoneNumber(5dsat), userPassword(5dsat)

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name pilotObject – Standard LDAP objectclass

Synopsis

```
( 0.9.2342.19200300.100.4.3
 NAME 'pilotObject'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MAY ( audio $
  ditRedirect $
  info $
  jpegPhoto $
  lastModifiedBy $
  lastModifiedTime $
  manager $
  photo $
  uniqueIdentifier )
 X-ORIGIN 'RFC 1274' )
```

Description Used as a subclass to allow additional attributes to be assigned to entries of all other object classes.

Origin This object class is defined by RFC 1274.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

audio(5dsat), ditRedirect(5dsat), info(5dsat), jpegPhoto(5dsat), lastModifiedBy(5dsat), lastModifiedTime(5dsat), manager(5dsat), photo(5dsat), uniqueIdentifier(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name pilotOrganization – Standard LDAP objectclass

Synopsis

```
( 0.9.2342.19200300.100.4.20
 NAME 'pilotOrganization'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST ( ou $
  o )
 MAY ( buildingName $
  businessCategory $
  description $
  destinationIndicator $
  facsimileTelephoneNumber $
  internationaliSDNNumber $
  l $
  physicalDeliveryOfficeName $
  postOfficeBox $
  postalAddress $
  postalCode $
  preferredDeliveryMethod $
  registeredAddress $
  searchGuide $
  seeAlso $
  st $
  street $
  telephoneNumber $
  teletexTerminalIdentifier $
  telexNumber $
  userPassword $
  x121Address )
 X-ORIGIN 'RFC 1274' )
```

Description Used as a subclass to allow additional attributes to be assigned to organization and organizationalUnit object class entries.

Origin This object class is defined by RFC 1274.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

o(5dsat), ou(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

buildingName(5dsat), businessCategory(5dsat), description(5dsat), destinationIndicator(5dsat), facsimileTelephoneNumber(5dsat), internationaliSDNNumber(5dsat), l(5dsat), physicalDeliveryOfficeName(5dsat), postOfficeBox(5dsat), postalAddress(5dsat), postalCode(5dsat), preferredDeliveryMethod(5dsat), registeredAddress(5dsat), searchGuide(5dsat), seeAlso(5dsat), st(5dsat), street(5dsat), telephoneNumber(5dsat), teletexTerminalIdentifier(5dsat), telexNumber(5dsat), userPassword(5dsat), x121Address(5dsat)

Attributes

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name posixAccount – Standard LDAP objectclass

Synopsis

```
( 1.3.6.1.1.1.2.0
 NAME 'posixAccount'
 DESC 'Standard LDAP objectclass'
 SUP top
 AUXILIARY
 MUST ( cn $
  uid $
  uidNumber $
  gidNumber $
  homeDirectory )
 MAY ( userPassword $
  loginShell $
  gecos $
  description )
 X-ORIGIN 'RFC 2307' )
```

Description Auxiliary object class.

Origin This object class is defined by RFC 2307.

Type Auxiliary object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat), gidNumber(5dsat), homeDirectory(5dsat), uid(5dsat), uidNumber(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat), gecos(5dsat), loginShell(5dsat), userPassword(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name posixGroup – Standard LDAP objectclass

Synopsis

```
( 1.3.6.1.1.1.2.2
 NAME 'posixGroup'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST ( cn $
  gidNumber )
 MAY ( userPassword $
  memberUid $
  description )
 X-ORIGIN 'RFC 2307' )
```

Description Structural object class.

Origin This object class is defined by RFC 2307.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat), gidNumber(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat), memberUid(5dsat), userPassword(5dsat)

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name

pwdPolicy – Password Policy objectclass

Synopsis

```
(  1.3.6.1.4.1.42.2.27.8.2.1
 NAME 'pwdPolicy'
 DESC 'Password Policy objectclass'
 SUP top
 AUXILIARY
 MUST ( pwdAttribute )
 MAY ( pwdMinAge $
  pwdMaxAge $
  pwdInHistory $
  pwdCheckQuality $
  pwdMinLength $
  pwdExpireWarning $
  pwdGraceAuthNLimit $
  pwdLockout $
  pwdLockoutDuration  $
  pwdMaxFailure $
  pwdFailureCountInterval $
  pwdMustChange $
  pwdAllowUserChange $
  pwdSafeModify)
 X-DS-USE 'internal'
 X-ORIGIN 'Password Policy for LDAP Directories Internet Draft' )
```

Description

Contains the attributes defining a password policy in effect for a set of users. A password policy is defined for a particular subtree of the DIT by adding to an LDAP subentry whose immediate superior is the root of the subtree, the pwdPolicy auxiliary object class. The scope of the password policy is defined by the SubtreeSpecification attribute of the LDAP subentry as specified in RFC 3672.

Each object that is controlled by password policy advertises the subentry that is being used to control its policy in its `pwdPolicySubentry` attribute. Clients wishing to examine or manage password policy for an object may interrogate the `pwdPolicySubentry` for that object in order to arrive at the proper `pwdPolicy` subentry.

Origin

This object class is defined by Password Policy Internet-Draft.

Type

Auxiliary object class

Superior

top(5dsoc)

Required Attributes

Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

pwdAttribute(5dsat)

Allowed Attributes

Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

pwdAllowUserChange(5dsat), pwdCheckQuality(5dsat), pwdExpireWarning(5dsat), pwdFailureCountInterval(5dsat), pwdGraceAuthNLimit(5dsat), pwdInHistory(5dsat), pwdLockout(5dsat), pwdLockoutDuration(5dsat), pwdMaxAge(5dsat), pwdMaxFailure(5dsat), pwdMinAge(5dsat), pwdMinLength(5dsat), pwdMustChange(5dsat), pwdSafeModify(5dsat)

Usage

Configuration object specific to this Directory Server instance, not replicated.

Attributes

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Password Policy Internet-Draft

Name

referral – Standard LDAP referral objectclass

Synopsis

```
( 2.16.840.1.113730.3.2.6
 NAME 'referral'
 DESC 'Standard LDAP referral objectclass'
 SUP top
 STRUCTURAL
 MUST ref
 X-ORIGIN 'RFC 3296' )
```

Description

Used to represent a subordinate reference information in the directory. These referral objects hold one or more URIs contained in values of the `ref` attribute type and are used to generate protocol referrals and continuations.

Note – To use this object class, you must either make it a subclass, or use it with the `extensibleObject` object class. This ensures that you have an attribute for naming the entry.

Origin

This object class is defined by RFC 3296.

Type

Structural object class

Superior

top(5dsoc)

Required Attributes

Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

ref(5dsat)

Allowed Attributes

Entries of this object class have no optional attribute types other than those inherited from the superior(s).

Attributes

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 3296

Name residentialPerson – Standard LDAP objectclass

Synopsis

```
( 2.5.6.10
 NAME ’residentialPerson’
 DESC ’Standard LDAP objectclass’
 SUP person
 STRUCTURAL
 MUST l
 MAY ( businessCategory $
  destinationIndicator $
  facsimileTelephoneNumber $
  internationaliSDNNumber $
  physicalDeliveryOfficeName $
  postOfficeBox $
  postalAddress $
  postalCode $
  preferredDeliveryMethod $
  registeredAddress $
  st $
  street $
  teletexTerminalIdentifier $
  telexNumber $
  x121Address )
 X-ORIGIN ’RFC 2256’ )
```

Description Used by Directory Server to contain a person's residential information.

Origin This object class is defined by RFC 2256.

Type Structural object class

Superior person(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

l(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

businessCategory(5dsat), destinationIndicator(5dsat), facsimileTelephoneNumber(5dsat), internationaliSDNNumber(5dsat), physicalDeliveryOfficeName(5dsat), postOfficeBox(5dsat), postalAddress(5dsat), postalCode(5dsat), preferredDeliveryMethod(5dsat), registeredAddress(5dsat), st(5dsat), street(5dsat), teletexTerminalIdentifier(5dsat), telexNumber(5dsat), x121Address(5dsat)

Attributes

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name RFC822localPart – Pilot objectclass

Synopsis

```
( 0.9.2342.19200300.100.4.14
 NAME 'RFC822localPart'
 DESC 'Pilot objectclass'
 SUP domain
 STRUCTURAL
 MAY ( cn $
  sn )
 X-ORIGIN 'Internet directory pilot' )
```

Description Used to define entries that represent the local part of RFC822 mail addresses. The directory treats this part of an RFC822 address as a domain.

Origin This object class is defined by Internet directory pilot.

Type Structural object class

Superior domain(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

cn(5dsat), sn(5dsat)

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	External: IETF, Internet directory pilot

Name room – Standard LDAP objectclass

Synopsis

```
( 0.9.2342.19200300.100.4.7
 NAME 'room'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST cn
 MAY ( description $
  roomNumber $
  seeAlso $
  telephoneNumber )
 X-ORIGIN 'RFC 1274' )
```

Description Used to store information in the directory about a room.

Origin This object class is defined by RFC 1274.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat), roomNumber(5dsat), seeAlso(5dsat), telephoneNumber(5dsat)

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name shadowAccount – Standard LDAP objectclass

Synopsis

```
( 1.3.6.1.1.1.2.1
 NAME 'shadowAccount'
 DESC 'Standard LDAP objectclass'
 SUP top
 AUXILIARY
 MUST uid
 MAY ( userPassword $
  shadowLastChange $
  shadowMin $
  shadowMax $
  shadowWarning $
  shadowInactive $
  shadowExpire $
  shadowFlag $
  description )
 X-ORIGIN 'RFC 2307' )
```

Description Auxiliary object class. Related to the `/etc/shadow` file.

Origin This object class is defined by RFC 2307.

Type Auxiliary object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

uid(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat), shadowExpire(5dsat), shadowFlag(5dsat), shadowInactive(5dsat), shadowLastChange(5dsat), shadowMax(5dsat), shadowMin(5dsat), shadowWarning(5dsat), userPassword(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2307

Name simpleSecurityObject – Standard LDAP objectclass

Synopsis

```
( 0.9.2342.19200300.100.4.19
 NAME 'simpleSecurityObject'
 DESC 'Standard LDAP objectclass'
 SUP top
 STRUCTURAL
 MUST userPassword
 X-ORIGIN 'RFC 1274' )
```

Description Used to allow an entry to contain the `userPassword` attribute when an entry's principal object classes do not allow `userPassword` as an attribute type. Reserved for future use.

Origin This object class is defined by RFC 1274.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

userPassword(5dsat)

Allowed Attributes Entries of this object class have no optional attribute types other than those inherited from the superior(s).

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 1274

Name strongAuthenticationUser – Standard LDAP objectclass

Synopsis

```
( 2.5.6.15
 NAME 'strongAuthenticationUser'
 DESC 'Standard LDAP objectclass'
 SUP top
 AUXILIARY
 MUST userCertificate
 X-ORIGIN 'RFC 2256' )
```

Description Auxiliary object class, used to store a user's certificate entry in the directory. This object class is used with other object classes, such as the `person` and `organization` object classes.

Origin This object class is defined by RFC 2256.

Type Auxiliary object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

userCertificate(5dsat)

Allowed Attributes Entries of this object class have no optional attribute types other than those inherited from the superior(s).

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name subschema – Standard LDAP objectclass

Synopsis

```
( 2.5.20.1
 NAME 'subschema'
 DESC 'Standard LDAP objectclass'
 SUP top
 AUXILIARY
 MAY ( dITStructureRules $
  nameForms $
  dITContentRules $
  objectClasses $
  attributeTypes $
  matchingRules $
  matchingRuleUse )
 X-ORIGIN 'RFC 2252' )
```

Description Internal object class. An auxiliary object class subentry used to administer the subschema for the subschema administrative area. It holds the operational attributes representing the policy parameters used to express the subschema.

Origin This object class is defined by RFC 2252.

Type Auxiliary object class

Superior top(5dsoc)

Required Attributes Entries of this object class require no attribute types other than those inherited from the superior(s).

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

attributeTypes(5dsat), dITContentRules(5dsat), dITStructureRules(5dsat), matchingRuleUse(5dsat), matchingRules(5dsat), nameForms(5dsat), objectClasses(5dsat)

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2252

Name sunPwdPolicy – Sun Directory Server Password Policy objectclass

Synopsis

```
( 1.3.6.1.4.1.42.2.27.9.2.119
 NAME 'sunPwdPolicy'
 DESC 'Sun Directory Server Password Policy objectclass'
 SUP pwdPolicy
 AUXILIARY
 MUST ( cn )
 MAY ( description $
  passwordRootdnMayBypassModsChecks $
  passwordStorageScheme $
  passwordExpireWithoutWarning $
  pwdIsLockoutPrioritized $
  pwdKeepLastAuthTime )
 X-DS-USE 'internal'
 X-ORIGIN 'Sun Directory Server' )
```

Description Contains attributes used in conjunction with the `pwdPolicy` object attributes to define a password policy in effect for a set of users.

Origin This object class is defined by Sun Java System Directory Server.

Type Auxiliary object class

Superior pwdPolicy(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

description(5dsat), passwordExpireWithoutWarning(5dsat), passwordRootdnMayBypassModsChecks(5dsat), passwordStorageScheme(5dsat), pwdIsLockoutPrioritized(5dsat), pwdKeepLastAuthTime(5dsat)

Usage Configuration object specific to this Directory Server instance, not replicated.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name top – Standard LDAP objectclass

Synopsis

```
( 2.5.6.0
 NAME 'top'
 DESC 'Standard LDAP objectclass'
 ABSTRACT
 MUST objectClass
 X-ORIGIN 'RFC 2256' )
```

Description Abstract object class, that defines the root of the object class hierarchy.

Origin This object class is defined by RFC 2256.

Type Abstract object class

Superior This object class has no superiors.

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

objectClass(5dsat)

Allowed Attributes Entries of this object class have no optional attribute types other than those inherited from the superior(s).

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Standard: IETF, RFC 2256

Name vlvIndex – Sun ONE defined objectclass

Synopsis

```
( 2.16.840.1.113730.3.2.42
 NAME 'vlvIndex'
 DESC 'Sun ONE defined objectclass'
 SUP top
 STRUCTURAL
 MUST ( cn $
  vlvSort )
 MAY ( vlvEnabled $
  vlvUses )
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Used to define the sort criteria of a Virtual List View index. Each VLV index specification defines the sort order to be imposed on the result set defined in the VLV search entry. A set of VLV index entries may appear below the VLV search entry. The common name, `CN`, attribute is used as the naming component for the entry.

Origin This object class is defined by Sun Java System Directory Server.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat), vlvSort(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

vlvEnabled(5dsat), vlvUses(5dsat)

Usage Configuration object specific to this Directory Server instance, not replicated.

Attributes See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Name vlvSearch – Sun ONE defined objectclass

Synopsis

```
( 2.16.840.1.113730.3.2.38
 NAME 'vlvSearch'
 DESC 'Sun ONE defined objectclass'
 SUP top
 STRUCTURAL
 MUST ( cn $
  vlvBase $
  vlvScope $
  vlvFilter )
 MAY multiLineDescription
 X-DS-USE 'internal'
 X-ORIGIN 'Sun ONE Directory Server' )
```

Description Used to define a VLV search. Specifies the entry result set to be VLV indexed.

Origin This object class is defined by Sun Java System Directory Server.

Type Structural object class

Superior top(5dsoc)

Required Attributes Entries of this object class require the following attribute types in addition to those inherited from the superior(s):

cn(5dsat), vlvBase(5dsat), vlvFilter(5dsat), vlvScope(5dsat)

Allowed Attributes Entries of this object class may have the following optional attribute types in addition to those inherited from the superior(s):

multiLineDescription(5dsat)

Usage Configuration object specific to this Directory Server instance, not replicated.

Attributes See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWldap-directory
Stability Level	Evolving

Index