NAME

ldap.conf, .ldaprc - LDAP configuration file/environment variables

SYNOPSIS

/etc/ldap/ldap.conf, ldaprc, .ldaprc, $LDAP<option-name>

DESCRIPTION

If the environment variable **LDAPNOINIT** is defined, all defaulting is disabled.

The *ldap.conf* configuration file is used to set system-wide defaults to be applied when running *ldap* clients.

Users may create an optional configuration file, *ldaprc* or *.ldaprc*, in their home directory which will be used to override the system-wide defaults file. The file *ldaprc* in the current working directory is also used.

Additional configuration files can be specified using the **LDAPCONF** and **LDAPRC** environment variables. **LDAPCONF** may be set to the path of a configuration file. This path can be absolute or relative to the current working directory. The**LD APRC**, if defined, should be the basename of a file in the current working directory or in the user's home directory.

Environmental variables may also be used to augment the file based defaults. The name of the variable is the option name with an added prefix of **LDAP**. For example, to define **BASE** via the environment, set the variable **LDAPBASE** to the desired value.

Some options are user-only. Such options are ignored if present in the *ldap.conf* (or file specified by **LDAPCONF**).

Thus the following files and variables are read, in order:
variable $LDAPNOINIT, and if that is not set:
system file /etc/ldap/ldap.conf,
user files $HOME/ldaprc, $HOME/.ldaprc, ./ldaprc,
system file $LDAPCONF,
user files $HOME/$LDAPRC, $HOME/.$LDAPRC, ./$LDAPRC,
variables $LDAP<uppercase option name>.
Settings late in the list override earlier ones.

SYNTAX

The configuration options are case-insensitive; their value, on a case by case basis, may be case-sensitive.

Blank lines are ignored.
Lines beginning with a hash mark ('#') are comments, and ignored.

Valid lines are made of an option's name (a sequence of non-blanks, conventionally written in uppercase, although not required), followed by a value. The value starts with the first non-blank character after the option's name, and terminates at the end of the line, or at the last sequence of blanks before the end of the line. The tokenization of the value, if any, is delegated to the handler(s) for that option, if any. Quoting values that contain blanks may be incorrect, as the quotes would become part of the value. For example,

Wrong - erroneous quotes:
URI "ldap:// ldaps://"

Right - space-separated list of URIs, without quotes:
URI ldap:// ldaps://

Right - DN syntax needs quoting for Example, Inc:
BASE ou=IT staff,o="Example, Inc",c=US
or:
BASE ou=IT staff,o=Example2C Inc,c=US

Wrong - comment on same line as option:
DEREF never # Never follow aliases

A line cannot be longer than LINE_MAX, which should be more than 2000 bytes on all platforms. There is no mechanism to split a long line on multiple lines, either for beautification or to overcome the above

limit.

OPTIONS

The different configuration options are:

URI <ldap[si]://[name[:port]] ...>

Specifies the URI(s) of an LDAP server(s) to which the *LDAP* library should connect. The URI scheme may be any of **ldap**, **ldaps** or **ldapi**, which refer to LDAP over TCP, LDAP over SSL (TLS) and LDAP over IPC (UNIX domain sockets), respectively. Each server's name can be specified as a domain-style name or an IP address literal. Optionally, the server's name can followed by a ':' and the port number the LDAP server is listening on. If no port number is provided, the default port for the scheme is used (389 for ldap://, 636 for ldaps://). For LDAP over IPC, **name** is the name of the socket, and no **port** is required, nor allowed; note that directory separators must be URL-encoded, like any other characters that are special to URLs; so the socket

/usr/local/var/ldapi

must be specified as

ldapi://%2Fusr%2Flocal%2Fvar%2Fldapi

A space separated list of URIs may be provided.

BASE <base>

Specifies the default base DN to use when performing ldap operations. The base must be specified as a Distinguished Name in LDAP format.

BINDDN <dn>

Specifies the default bind DN to use when performing ldap operations. The bind DN must be specified as a Distinguished Name in LDAP format. **This is a user-only option.**

DEREF <when>

Specifies how alias dereferencing is done when performing a search. The **<when>** can be specified as one of the following keywords:

never Aliases are never dereferenced. This is the default.

searching

Aliases are dereferenced in subordinates of the base object, but not in locating the base object of the search.

finding Aliases are only dereferenced when locating the base object of the search.

always Aliases are dereferenced both in searching and in locating the base object of the search.

HOST <name[:port] ...>

Specifies the name(s) of an LDAP server(s) to which the *LDAP* library should connect. Each server's name can be specified as a domain-style name or an IP address and optionally followed by a ':' and the port number the ldap server is listening on. A space separated list of hosts may be provided. **HOST** is deprecated in favor of **URI**.

NETWORK_TIMEOUT <integer>

Specifies the timeout (in seconds) after which the poll(2) following a connect(2) returns in case of no activity.

PORT <port>

Specifies the default port used when connecting to LDAP servers(s). The port may be specified as a number. **PORT** is deprecated in favor of **URI.**

REFERRALS <on/true/yes/off/false/no>

Specifies if the client should automatically follow referrals returned by LDAP servers. The default is on. Note that the command line tools **ldapsearch(1)** &co always override this option.

SIZELIMIT <integer>
Specifies a size limit (number of entries) to use when performing searches. The number should be a non-negative integer. *SIZELIMIT* of zero (0) specifies a request for unlimited search size. Please note that the server may still apply any server-side limit on the amount of entries that can be returned by a search operation.

TIMELIMIT <integer>
Specifies a time limit (in seconds) to use when performing searches. The number should be a non-negative integer. *TIMELIMIT* of zero (0) specifies unlimited search time to be used. Please note that the server may still apply any server-side limit on the duration of a search operation. **VERSION {2|3}** Specifies what version of the LDAP protocol should be used.

TIMEOUT <integer>
Specifies a timeout (in seconds) after which calls to synchronous LDAP APIs will abort if no response is received. Also used for any **ldap_result(3)** calls where a NULL timeout parameter is supplied.

SASL OPTIONS

If OpenLDAP is built with Simple Authentication and Security Layer support, there are more options you can specify.

SASL_MECH <mechanism>
Specifies the SASL mechanism to use. **This is a user-only option.**

SASL_REALM <realm>
Specifies the SASL realm. **This is a user-only option.**

SASL_AUTHCID <authcid>
Specifies the authentication identity. **This is a user-only option.**

SASL_AUTHZID <authcid>
Specifies the proxy authorization identity. **This is a user-only option.**

SASL_SECPROPS <properties>
Specifies Cyrus SASL security properties. The **<properties>** can be specified as a comma-separated list of the following:

none (without any other properties) causes the properties defaults ("noanonymous,noplain") to be cleared.

noplain
disables mechanisms susceptible to simple passive attacks.

noactive
disables mechanisms susceptible to active attacks.

nodict disables mechanisms susceptible to passive dictionary attacks.

noanonymous
disables mechanisms which support anonymous login.

forwardsec
requires forward secrecy between sessions.

passcred
requires mechanisms which pass client credentials (and allows mechanisms which can pass credentials to do so).

minssf=<factor>
specifies the minimum acceptable *security strength factor* as an integer approximating the effective key length used for encryption. 0 (zero) implies no protection, 1 implies integrity protection only, 56 allows DES or other weak ciphers, 112 allows triple DES and other strong ciphers, 128 allows RC4, Blowfish and other modern strong ciphers. The default is 0.

maxssf=<factor>
specifies the maximum acceptable *security strength factor* as an integer (see **minssf** description). The default is **INT_MAX**.

maxbufsize=<factor>
specifies the maximum security layer receive buffer size allowed. 0 disables security layers. The default is 65536.

GSSAPI OPTIONS

If OpenLDAP is built with Generic Security Services Application Programming Interface support, there are more options you can specify.

GSSAPI_SIGN <on/true/yes/off/false/no>
Specifies if GSSAPI signing (GSS_C_INTEG_FLAG) should be used. The default is off.

GSSAPI_ENCRYPT <on/true/yes/off/false/no>
Specifies if GSSAPI encryption (GSS_C_INTEG_FLAG and GSS_C_CONF_FLAG) should be used. The default is off.

GSSAPI_ALLOW_REMOTE_PRINCIPAL <on/true/yes/off/false/no>
Specifies if GSSAPI based authentication should try to form the target principal name out of the ldapServiceName or dnsHostName attribute of the targets RootDSE entry. The default is off.

TLS OPTIONS

If OpenLDAP is built with Transport Layer Security support, there are more options you can specify. These options are used when an **ldaps:// URI** is selected (by default or otherwise) or when the application negotiates TLS by issuing the LDAP StartTLS operation.

TLS_CACERT <filename>
Specifies the file that contains certificates for all of the Certificate Authorities the client will recognize.

TLS_CACERTDIR <path>
Specifies the path of a directory that contains Certificate Authority certificates in separate individual files. The **TLS_CACERT** is always used before **TLS_CACERTDIR.** This parameter is ignored with GnuTLS. On Debian openldap is linked against GnuTLS.

When using Mozilla NSS, <path> may contain a Mozilla NSS cert/key database. If <path> contains a Mozilla NSS cert/key database and CA cert files, OpenLDAP will use the cert/key database and will ignore the CA cert files.

TLS_CERT <filename>
Specifies the file that contains the client certificate. **This is a user-only option.**

When using Mozilla NSS, if using a cert/key database (specified with TLS_CACERTDIR), TLS_CERT specifies the name of the certificate to use:
TLS_CERT Certificate for Sam Carter
If using a token other than the internal built in token, specify the token name first, followed by a colon:
TLS_CERT my hardware device:Certificate for Sam Carter
Use certutil -L to list the certificates by name:
certutil -d /path/to/certdbdir -L

TLS_KEY <filename>
Specifies the file that contains the private key that matches the certificate stored in the **TLS_CERT** file. Currently, the private key must not be protected with a password, so it is of critical importance that the key file is protected carefully. **This is a user-only option.**

When using Mozilla NSS, TLS_KEY specifies the name of a file that contains the password for the key for the certificate specified with TLS_CERT. The modutil command can be used to turn off password protection for the cert/key database. For example, if TLS_CACERTDIR specifies /home/scarter/.moznss as the location of the cert/key database, use modutil to change the password

to the empty string:
modutil -dbdir ~/.moznss -changepw 'NSS Certificate DB'
You must have the old password, if any. Ignore the WARNING about the running browser. Press 'Enter' for the new password.

TLS_CIPHER_SUITE <cipher-suite-spec>

Specifies acceptable cipher suite and preference order. <cipher-suite-spec> should be a cipher specification for the TLS library in use (OpenSSL, GnuTLS, or Mozilla NSS). Example:

OpenSSL:

TLS_CIPHER_SUITE HIGH:MEDIUM:+SSLv2

GnuTLS:

TLS_CIPHER_SUITE SECURE256:!AES-128-CBC

To check what ciphers a given spec selects in OpenSSL, use:

openssl ciphers -v <cipher-suite-spec>

With GnuTLS the available specs can be found in the manual page of **gnutls-cli(1)** (see the description of the option **--priority**).

In older versions of GnuTLS, where gnutls-cli does not support the option --priority, you can obtain the — more limited — list of ciphers by calling:

gnutls-cli -l

When using Mozilla NSS, the OpenSSL cipher suite specifications are used and translated into the format used internally by Mozilla NSS. There isn't an easy way to list the cipher suites from the command line. The authoritative list is in the source code for Mozilla NSS in the file sslinfo.c in the structure
static const SSLCipherSuiteInfo suiteInfo[]

TLS_PROTOCOL_MIN <major>[.<minor>]

Specifies minimum SSL/TLS protocol version that will be negotiated. If the server doesn't support at least that version, the SSL handshake will fail. To require TLS 1.x or higher, set this option to 3.(x+1), e.g.,

TLS_PROTOCOL_MIN 3.2

would require TLS 1.1. Specifying a minimum that is higher than that supported by the OpenLDAP implementation will result in it requiring the highest level that it does support. This parameter is ignored with GnuTLS.

TLS_RANDFILE <filename>

Specifies the file to obtain random bits from when /dev/[u]random is not available. Generally set to the name of the EGD/PRNGD socket. The environment variable RANDFILE can also be used to specify the filename. This parameter is ignored with GnuTLS and Mozilla NSS. On Debian openldap is linked against GnuTLS.

TLS_REQCERT <level>

Specifies what checks to perform on server certificates in a TLS session, if any. The **<level>** can be specified as one of the following keywords:

never The client will not request or check any server certificate.

allow The server certificate is requested. If no certificate is provided, the session proceeds normally. If a bad certificate is provided, it will be ignored and the session proceeds normally.

try The server certificate is requested. If no certificate is provided, the session proceeds normally. If a bad certificate is provided, the session is immediately terminated.

demand | hard
These keywords are equivalent. The server certificate is requested. If no certificate is provided, or a bad certificate is provided, the session is immediately terminated. This is the default setting.

TLS_CRLCHECK <level>
Specifies if the Certificate Revocation List (CRL) of the CA should be used to verify if the server certificates have not been revoked. This requires **TLS_CACERTDIR** parameter to be set. This parameter is ignored with GnuTLS and Mozilla NSS. On Debian openldap is linked against GnuTLS. **<level>** can be specified as one of the following keywords:

none No CRL checks are performed

peer Check the CRL of the peer certificate

all Check the CRL for a whole certificate chain

TLS_CRLFILE <filename>
Specifies the file containing a Certificate Revocation List to be used to verify if the server certificates have not been revoked. This parameter is only supported with GnuTLS and Mozilla NSS.

ENVIRONMENT VARIABLES

LDAPNOINIT
disable all defaulting

LDAPCONF
path of a configuration file

LDAPRC
basename of ldaprc file in $HOME or $CWD

LDAP<option-name>
Set <option-name> as from ldap.conf

FILES

/etc/ldap/ldap.conf
system-wide ldap configuration file

$HOME/ldaprc, $HOME/.ldaprc
user ldap configuration file

$CWD/ldaprc
local ldap configuration file

SEE ALSO

ldap(3), ldap_set_option(3), ldap_result(3), openssl(1), **sasl(3)**

AUTHOR

Kurt Zeilenga, The OpenLDAP Project

ACKNOWLEDGEMENTS

OpenLDAP Software is developed and maintained by The OpenLDAP Project <http://www.openldap.org/>. **OpenLDAP Software** is derived from University of Michigan LDAP 3.3 Release.