

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 **LDAPRC**, 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=Example\2C 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 be 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**.

KEEPA_LIVE_IDLE

Sets/gets the number of seconds a connection needs to remain idle before TCP starts sending keepalive probes. Linux only.

KEEPA_LIVE_PROBES

Sets/gets the maximum number of keepalive probes TCP should send before dropping the connection. Linux only.

KEEPALIVE_INTERVAL

Sets/gets the interval in seconds between individual keepalive probes. Linux only.

NETWORK_TIMEOUT <integer>

Specifies the timeout (in seconds) after which the poll(2)/select(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 **URL**.

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.

SOCKET_BIND_ADDRESSES <IP>

Specifies the source bind IP to be used for connecting to target LDAP server. Multiple IP addresses must be space separated. Only one valid IPv4 address and/or one valid IPv6 address are allowed in the list.

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.

SASL_REALM <realm>

Specifies the SASL realm.

SASL_AUTHCID <authcid>

Specifies the authentication identity. **This is a user-only option.**

SASL_AUTHZID <authzid>

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 approximate to effective key length used for encryption. 0 (zero) implies no protection, 1 implies integrity protection only, 128 allows RC4, Blowfish and other similar ciphers, 256 will require modern 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.

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

Do not perform reverse DNS lookups to canonicalize SASL host names. The default is off.

SASL_CBINDING <none/tls-unique/tls-endpoint>

The channel-binding type to use, see also LDAP_OPT_X_SASL_CBINDING. The default is none.

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**.

TLS_CERT <filename>

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

TLS_ECNAME <name>

Specify the name of the curve(s) to use for Elliptic curve Diffie-Hellman ephemeral key exchange. This option is only used for OpenSSL. This option is not used with GnuTLS; the curves may be chosen in the GnuTLS ciphersuite specification.

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.**

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 or GnuTLS). 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
```

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. On Debian **openldap** is linked against 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. On Debian **openldap** is linked against GnuTLS.

TLS_REQCERT <level>

Specifies what checks to perform on server certificates in a TLS session. 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 a bad certificate is provided, it will be ignored and the session proceeds normally.
- try** The server certificate is requested. If a bad certificate is provided, the session is immediately terminated.
- demand | hard**
These keywords are equivalent and the same as **try**. This is the default setting.

TLS_REQSAN <level>

Specifies what checks to perform on the subjectAlternativeName (SAN) extensions in a server certificate when validating the certificate name against the specified hostname of the server. The **<level>** can be specified as one of the following keywords:

- never** The client will not check any SAN in the certificate.
- allow** The SAN is checked against the specified hostname. If a SAN is present but none match the specified hostname, the SANs are ignored and the usual check against the certificate DN is used. This is the default setting.
- try** The SAN is checked against the specified hostname. If no SAN is present in the server certificate, the usual check against the certificate DN is used. If a SAN is present but doesn't match the specified hostname, the session is immediately terminated. This setting may be preferred when a mix of certs with and without SANs are in use.
- demand | hard**
These keywords are equivalent. The SAN is checked against the specified hostname. If no SAN is present in the server certificate, or no SANs match, the session is immediately terminated. This setting should be used when only certificates with SANs are in use.

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. 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.

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 the University of Michigan LDAP 3.3 Release.