NAME

ldap_get_option, ldap_set_option - LDAP option handling routines

LIBRARY

OpenLDAP LDAP (libldap, -lldap)

SYNOPSIS

#include <ldap.h>

int ldap_get_option(LDAP ****ld*, int** *option***, void ****outvalue***);**

int ldap_set_option(LDAP ****ld*, int** *option***, const void ****invalue***);**

DESCRIPTION

These routines provide access to options stored either in a LDAP handle or as global options, where applicable. They make use of a neutral interface, where the type of the value either retrieved by **ldap_get_option**(3) or set by **ldap_set_option**(3) is cast to **void ***. The actual type is determined based on the value of the **option** argument. Global options are set/retrieved by passing a NULL LDAP handle. LDAP handles inherit their default settings from the global options in effect at the time the handle is created.

LDAP_OPT_API_FEATURE_INFO

Fills-in a **LDAPAPIFeatureInfo**; **outvalue** must be a **LDAPAPIFeatureInfo ***, pointing to an already allocated struct. The **ldapaif_info_version** field of the struct must be initialized to **LDAP_FEATURE_INFO_VERSION** before making the call. The **ldapaif_name** field must be set to the name of a feature to query. This is a read-only option.

LDAP_OPT_API_INFO

Fills-in a **LDAPAPIInfo**; **outvalue** must be a **LDAPAPIInfo ***, pointing to an already allocated struct. The **ldapai_info_version** field of the struct must be initialized to **LDAP_API_INFO_VERSION** before making the call. If the version passed in does not match the current library version, the expected version number will be stored in the struct and the call will fail. The caller is responsible for freeing the elements of the **ldapai_extensions** array and the array itself using **ldap_memfree**(3). The caller must also free the **ldapi_vendor_name**. This is a read-only option.

LDAP_OPT_CLIENT_CONTROLS

Sets/gets the client-side controls to be used for all operations. This is now deprecated as modern LDAP C API provides replacements for all main operations which accepts client-side controls as explicit arguments; see for example **ldap_search_ext**(3), **ldap_add_ext**(3), **ldap_modify_ext**(3) and so on. **outvalue** must be **LDAPControl *****, and the caller is responsible of freeing the returned controls, if any, by calling **ldap_controls_free**(3), while **invalue** must be **LDAPControl *const ***; the library duplicates the controls passed via **invalue**.

LDAP_OPT_CONNECT_ASYNC

Sets/gets the status of the asynchronous connect flag. **invalue** should either be **LDAP_OPT_OFF** or **LDAP_OPT_ON**; **outvalue** must be **int ***. When set, the library will call **connect**(2) and return, without waiting for response. This leaves the handle in a connecting state. Subsequent calls to library routines will poll for completion of the connect before performing further operations. As a consequence, library calls that need to establish a connection with a DSA do not block even for the network timeout (option **LDAP_OPT_NETWORK_TIMEOUT**). This option is OpenLDAP specific.

LDAP_OPT_CONNECT_CB

This option allows to set a connect callback. **invalue** must be a **const struct ldap_conncb ***. Callbacks are executed in last in-first served order. Handle-specific callbacks are executed first, followed by global ones. Right before freeing the callback structure, the **lc_del** callback handler is passed a **NULL Sockbuf**. Calling **ldap_get_option**(3) for this option removes the callback whose pointer matches **outvalue**. This option is OpenLDAP specific.

LDAP_OPT_DEBUG_LEVEL

Sets/gets the debug level of the client library. **invalue** must be a **const int ***; **outvalue** must be a **int ***. Valid debug levels are **LDAP_DEBUG_ANY**, **LDAP_DEBUG_ARGS**, **LDAP_DEBUG_BER**, **LDAP_DEBUG_CONNS**, **LDAP_DEBUG_NONE**, **LDAP_DEBUG_PACKETS**, **LDAP_DEBUG_PARSE**, and **LDAP_DEBUG_TRACE**. This option is OpenLDAP specific.

LDAP_OPT_DEFBASE

Sets/gets a string containing the DN to be used as default base for search operations. **outvalue** must be a **char ****, and the caller is responsible of freeing the returned string by calling **ldap_memfree**(3), while **invalue** must be a **const char ***; the library duplicates the corresponding string. This option is OpenLDAP specific.

LDAP_OPT_DEREF

Sets/gets the value that defines when alias dereferencing must occur. **invalue** must be **const int ***; **outvalue** must be **int ***. They cannot be NULL. The value of ***invalue** should be one of **LDAP_DEREF_NEVER** (the default), **LDAP_DEREF_SEARCHING**, **LDAP_DEREF_FINDING**, or **LDAP_DEREF_ALWAYS**. Note that this has ever been the only means to determine alias dereferencing within search operations.

LDAP_OPT_DESC

Returns the file descriptor associated to the socket buffer of the LDAP handle passed in as **ld**; **outvalue** must be a **int ***. This is a read-only, handle-specific option.

LDAP_OPT_DIAGNOSTIC_MESSAGE

Sets/gets a string containing the error string associated to the LDAP handle. This option was formerly known as **LDAP_OPT_ERROR_STRING**. **outvalue** must be a **char ****, and the caller is responsible of freeing the returned string by calling **ldap_memfree**(3), while **invalue** must be a **char ***; the library duplicates the corresponding string.

LDAP_OPT_HOST_NAME

Sets/gets a space-separated list of hosts to be contacted by the library when trying to establish a connection. This is now deprecated in favor of **LDAP_OPT_URI**. **outvalue** must be a **char ****, and the caller is responsible of freeing the resulting string by calling **ldap_memfree**(3), while **invalue** must be a **const char ***; the library duplicates the corresponding string.

LDAP_OPT_MATCHED_DN

Sets/gets a string containing the matched DN associated to the LDAP handle. **outvalue** must be a **char ****, and the caller is responsible of freeing the returned string by calling **ldap_memfree**(3), while **invalue** must be a **const char ***; the library duplicates the corresponding string.

LDAP_OPT_NETWORK_TIMEOUT

Sets/gets the network timeout value after which **poll**(2)/**select**(2) following a **connect**(2) returns in case of no activity. **outvalue** must be a **struct timeval **** (the caller has to free ***outvalue** using **ldap_memfree**(3)), and **invalue** must be a **const struct timeval ***. They cannot be NULL. Using a struct with seconds set to -1 results in an infinite timeout, which is the default. This option is OpenLDAP specific.

LDAP_OPT_PROTOCOL_VERSION

Sets/gets the protocol version. **outvalue** and **invalue** must be **int ***.

LDAP_OPT_REFERRAL_URLS

Sets/gets an array containing the referral URIs associated to the LDAP handle. **outvalue** must be a **char *****, and the caller is responsible of freeing the returned string by calling **ldap_memvfree**(3), while **invalue** must be a NULL-terminated **char *const ***; the library duplicates the corresponding string. This option is OpenLDAP specific.

LDAP_OPT_REFERRALS

Determines whether the library should implicitly chase referrals or not. **invalue** must be **const int ***; its value should either be **LDAP_OPT_OFF** or **LDAP_OPT_ON**. **outvalue** must be **int ***.

LDAP_OPT_RESTART

Determines whether the library should implicitly restart connections (FIXME). **invalue** must be **const int ***; its value should either be **LDAP_OPT_OFF** or **LDAP_OPT_ON**. **outvalue** must be **int ***.

LDAP_OPT_RESULT_CODE

Sets/gets the LDAP result code associated to the handle. This option was formerly known as **LDAP_OPT_ERROR_NUMBER**. **invalue** must be a **const int ***. **outvalue** must be a **int ***.

LDAP_OPT_SERVER_CONTROLS

Sets/gets the server-side controls to be used for all operations. This is now deprecated as modern LDAP C API provides replacements for all main operations which accepts server-side controls as explicit arguments; see for example **ldap_search_ext**(3), **ldap_add_ext**(3), **ldap_modify_ext**(3) and so on. **outvalue** must be **LDAPControl *****, and the caller is responsible of freeing the returned controls, if any, by calling **ldap_controls_free**(3), while **invalue** must be **LDAPControl *const ***; the library duplicates the controls passed via **invalue**.

LDAP_OPT_SESSION_REFCNT

Returns the reference count associated with the LDAP handle passed in as **ld**; **outvalue** must be a **int ***. This is a read-only, handle-specific option. This option is OpenLDAP specific.

LDAP_OPT_SIZELIMIT

Sets/gets the value that defines the maximum number of entries to be returned by a search operation. **invalue** must be **const int ***, while **outvalue** must be **int ***; They cannot be NULL.

LDAP_OPT_SOCKBUF

Returns a pointer to the socket buffer of the LDAP handle passed in as **ld**; **outvalue** must be a **Sockbuf ****. This is a read-only, handle-specific option. This option is OpenLDAP specific.

LDAP_OPT_SOCKET_BIND_ADDRESSES

Sets/gets a space-separated list of IP Addresses used as binding interface to remote server when trying to establish a connection. Only one valid IPv4 address and/or one valid IPv6 address are allowed in the list. **outvalue** must be a **char ****, and the caller is responsible of freeing the returned string by calling **ldap_memfree**(3), while **invalue** must be a **const char ***; the library duplicates the corresponding string.

LDAP_OPT_TIMELIMIT

Sets/gets the value that defines the time limit after which a search operation should be terminated by the server. **invalue** must be **const int ***, while **outvalue** must be **int ***, and they cannot be NULL.

LDAP_OPT_TIMEOUT

Sets/gets a timeout value for the synchronous API calls. **outvalue** must be a **struct timeval **** (the caller has to free ***outvalue** using **ldap_memfree**(3)), and **invalue** must be a **struct timeval ***, and they cannot be NULL. Using a struct with seconds set to -1 results in an infinite timeout, which is the default. This option is OpenLDAP specific.

LDAP_OPT_URI

Sets/gets a comma- or space-separated list of URIs to be contacted by the library when trying to establish a connection. **outvalue** must be a **char ****, and the caller is responsible of freeing the resulting string by calling **ldap_memfree**(3), while **invalue** must be a **const char ***; the library parses the string into a list of **LDAPURLDesc** structures, so the invocation of **ldap_set_option**(3) may fail if URL parsing fails. URIs may only contain the **schema**, the **host**, and the **port** fields. This option is OpenLDAP specific.

LDAP_OPT_KEEPCONN

Instructs **ldap_result**(3) to keep the connection open on read error or if Notice of Disconnection is received. In these cases, the connection should be closed by the caller. This option is OpenLDAP specific.

LDAP_OPT_TCP_USER_TIMEOUT

Allows to configure TCP_USER_TIMEOUT in milliseconds on the connection, overriding the operating system setting. This option is OpenLDAP specific and supported only on Linux 2.6.37 or higher. **invalue** must be a **const unsigned int ***; **outvalue** must be an **unsigned int ***.

SASL OPTIONS

The SASL options are OpenLDAP specific and unless otherwise noted, require an LDAP handle to be passed.

LDAP_OPT_X_SASL_AUTHCID

Gets the SASL authentication identity; **outvalue** must be a **char ****, its content needs to be freed by the caller using **ldap_memfree**(3).

LDAP_OPT_X_SASL_AUTHZID

Gets the SASL authorization identity; **outvalue** must be a **char ****, its content needs to be freed by the caller using **ldap_memfree**(3).

LDAP_OPT_X_SASL_MAXBUFSIZE

Gets/sets SASL maximum buffer size; **invalue** must be **const ber_len_t ***, while **outvalue** must be **ber_len_t ***. See also **LDAP_OPT_X_SASL_SECPROPS**.

LDAP_OPT_X_SASL_MECH

Gets the SASL mechanism; **outvalue** must be a **char ****, its content needs to be freed by the caller using **ldap_memfree**(3).

LDAP_OPT_X_SASL_MECHLIST

Gets the list of the available mechanisms, in form of a NULL-terminated array of strings; **outvalue** must be **char *****. The caller must not free or otherwise muck with it. This option can be used globally.

LDAP_OPT_X_SASL_NOCANON

Sets/gets the NOCANON flag. When unset, the hostname is canonicalized. **invalue** must be **const int ***; its value should either be **LDAP_OPT_OFF** or **LDAP_OPT_ON**. **outvalue** must be **int ***.

LDAP_OPT_X_SASL_REALM

Gets the SASL realm; **outvalue** must be a **char ****, its content needs to be freed by the caller using **ldap_memfree**(3).

LDAP_OPT_X_SASL_SECPROPS

Sets the SASL secprops; **invalue** must be a **char ***, containing a comma-separated list of properties. Legal values are: **none**, **nodict**, **noplain**, **noactive**, **passcred**, **forwardsec**, **noanonymous**, **minssf=<minssf>**, **maxssf=<maxssf>**, **maxbufsize=<maxbufsize>**.

LDAP_OPT_X_SASL_SSF

Gets the SASL SSF; **outvalue** must be a **ber_len_t ***.

LDAP_OPT_X_SASL_SSF_EXTERNAL

Sets the SASL SSF value related to an authentication performed using an EXTERNAL mechanism; **invalue** must be a **const ber_len_t ***.

LDAP_OPT_X_SASL_SSF_MAX

Gets/sets SASL maximum SSF; **invalue** must be **const ber_len_t ***, while **outvalue** must be **ber_len_t ***. See also **LDAP_OPT_X_SASL_SECPROPS**.

LDAP_OPT_X_SASL_SSF_MIN

Gets/sets SASL minimum SSF; **invalue** must be **const ber_len_t ***, while **outvalue** must be **ber_len_t ***. See also **LDAP_OPT_X_SASL_SECPROPS**.

LDAP_OPT_X_SASL_USERNAME

Gets the SASL username; **outvalue** must be a **char ****. Its content needs to be freed by the caller using **ldap_memfree**(3).

LDAP_OPT_X_SASL_CBINDING

Sets/gets the channel-binding type to use in SASL, one of
LDAP_OPT_X_SASL_CBINDING_NONE (the default),
LDAP_OPT_X_SASL_CBINDING_TLS_UNIQUE the "tls-unique" type from RFC 5929.
LDAP_OPT_X_SASL_CBINDING_TLS_ENDPOINT the "tls-server-end-point" from RFC 5929, compatible with Windows. **invalue** must be **const int ***; **outvalue** must be **int ***.

TCP OPTIONS

The TCP options are OpenLDAP specific. Mainly intended for use with Linux, they may not be portable.

LDAP_OPT_X_KEEPALIVE_IDLE

Sets/gets the number of seconds a connection needs to remain idle before TCP starts sending keepalive probes. **invalue** must be **const int ***; **outvalue** must be **int ***.

LDAP_OPT_X_KEEPALIVE_PROBES

Sets/gets the maximum number of keepalive probes TCP should send before dropping the connection. **invalue** must be **const int ***; **outvalue** must be **int ***.

LDAP_OPT_X_KEEPALIVE_INTERVAL

Sets/gets the interval in seconds between individual keepalive probes. **invalue** must be **const int ***; **outvalue** must be **int ***.

TLS OPTIONS

The TLS options are OpenLDAP specific.

LDAP_OPT_X_TLS_CACERTDIR

Sets/gets the path of the directories containing CA certificates. Multiple directories may be specified, separated by a semi-colon. **invalue** must be **const char ***; **outvalue** must be **char ****, and its contents need to be freed by the caller using **ldap_memfree**(3).

LDAP_OPT_X_TLS_CACERTFILE

Sets/gets the full-path of the CA certificate file. **invalue** must be **const char ***; **outvalue** must be **char ****, and its contents need to be freed by the caller using **ldap_memfree**(3).

LDAP_OPT_X_TLS_CERTFILE

Sets/gets the full-path of the certificate file. **invalue** must be **const char ***; **outvalue** must be **char ****, and its contents need to be freed by the caller using **ldap_memfree**(3).

LDAP_OPT_X_TLS_CIPHER

Gets the cipher being used on an established TLS session. **outvalue** must be **char ****, and its contents need to be freed by the caller using **ldap_memfree**(3).

LDAP_OPT_X_TLS_CIPHER_SUITE
Sets/gets the allowed cipher suite. **invalue** must be **const char ***; **outvalue** must be **char ****, and its contents need to be freed by the caller using **ldap_memfree**(3).

LDAP_OPT_X_TLS_CONNECT_ARG
Sets/gets the connection callback argument. **invalue** must be **const void ***; **outvalue** must be **void ****.

LDAP_OPT_X_TLS_CONNECT_CB
Sets/gets the connection callback handle. **invalue** must be **const LDAP_TLS_CONNECT_CB ***; **outvalue** must be **LDAP_TLS_CONNECT_CB ****.

LDAP_OPT_X_TLS_CRLCHECK
Sets/gets the CRL evaluation strategy, one of **LDAP_OPT_X_TLS_CRL_NONE**, **LDAP_OPT_X_TLS_CRL_PEER**, or **LDAP_OPT_X_TLS_CRL_ALL**. **invalue** must be **const int ***; **outvalue** must be **int ***. Requires OpenSSL.

LDAP_OPT_X_TLS_CRLFILE
Sets/gets the full-path of the CRL file. **invalue** must be **const char ***; **outvalue** must be **char ****, and its contents need to be freed by the caller using **ldap_memfree**(3). This option is only valid for GnuTLS.

LDAP_OPT_X_TLS_CTX
Sets/gets the TLS library context. New TLS sessions will inherit their default settings from this library context. **invalue** must be **const void ***; **outvalue** must be **void ****. When using the OpenSSL library this is an SSL_CTX*. When using other crypto libraries this is a pointer to an OpenLDAP private structure. Applications generally should not use this option or attempt to manipulate this structure.

LDAP_OPT_X_TLS_DHFILE
Gets/sets the full-path of the file containing the parameters for Diffie-Hellman ephemeral key exchange. **invalue** must be **const char ***; **outvalue** must be **char ****, and its contents need to be freed by the caller using **ldap_memfree**(3).

LDAP_OPT_X_TLS_ECNAME
Gets/sets the name of the curve(s) used for elliptic curve key exchanges. **invalue** must be **const char ***; **outvalue** must be **char ****, and its contents need to be freed by the caller using

ldap_memfree(3). Ignored by GnuTLS. In GnuTLS a curve may be selected in the cipher suite specification.

LDAP_OPT_X_TLS_KEYFILE

Sets/gets the full-path of the certificate key file. **invalue** must be **const char ***; **outvalue** must be **char ****, and its contents need to be freed by the caller using **ldap_memfree**(3).

LDAP_OPT_X_TLS_NEWCTX

Instructs the library to create a new TLS library context. **invalue** must be **const int ***. A non-zero value pointed to by **invalue** tells the library to create a context for a server.

LDAP_OPT_X_TLS_PEERCERT

Gets the peer's certificate in DER format from an established TLS session. **outvalue** must be **struct berval ***, and the data it returns needs to be freed by the caller using **ldap_memfree**(3).

LDAP_OPT_X_TLS_PROTOCOL_MAX

Sets/gets the maximum protocol version. **invalue** must be **const int ***; **outvalue** must be **int ***.

LDAP_OPT_X_TLS_PROTOCOL_MIN

Sets/gets the minimum protocol version. **invalue** must be **const int ***; **outvalue** must be **int ***.

LDAP_OPT_X_TLS_RANDOM_FILE

Sets/gets the random file when **/dev/random** and **/dev/urandom** are not available. **invalue** must be **const char ***; **outvalue** must be **char ****, and its contents need to be freed by the caller using **ldap_memfree**(3). Ignored by GnuTLS older than version 2.2.

LDAP_OPT_X_TLS_REQUIRE_CERT

Sets/gets the peer certificate checking strategy, one of **LDAP_OPT_X_TLS_NEVER**, **LDAP_OPT_X_TLS_HARD**, **LDAP_OPT_X_TLS_DEMAND**, **LDAP_OPT_X_TLS_ALLOW**, **LDAP_OPT_X_TLS_TRY**.

LDAP_OPT_X_TLS_REQUIRE_SAN

Sets/gets the peer certificate subjectAlternativeName checking strategy, one of **LDAP_OPT_X_TLS_NEVER**, **LDAP_OPT_X_TLS_HARD**, **LDAP_OPT_X_TLS_DEMAND**, **LDAP_OPT_X_TLS_ALLOW**, **LDAP_OPT_X_TLS_TRY**.

LDAP_OPT_X_TLS_SSL_CTX

Gets the TLS session context associated with this handle. **outvalue** must be **void ****. When using the OpenSSL library this is an SSL*. When using other crypto libraries this is a pointer to an OpenLDAP private structure. Applications generally should not use this option.

LDAP_OPT_X_TLS_VERSION

Gets the TLS version being used on an established TLS session. **outvalue** must be **char ****, and its contents need to be freed by the caller using **ldap_memfree**(3).

LDAP_OPT_X_TLS_PEERKEY_HASH

Sets the (public) key that the application expects the peer to be using. **invalue** must be **const char *** containing the base64 encoding of the expected peer's key or in the format **<hashalg>:<peerkey hash base64 encoded>** where as a TLS session is established, the library will hash the peer's key with the provided hash algorithm and compare it with value provided and will only allow the session to continue if they match. This happens regardless of certificate checking strategy. The list of supported **hashalg** values depends on the crypto library used, check its documentation to get a list.

ERRORS

On success, the functions return **LDAP_OPT_SUCCESS**, while they may return **LDAP_OPT_ERROR** to indicate a generic option handling error. Occasionally, more specific errors can be returned, like **LDAP_NO_MEMORY** to indicate a failure in memory allocation.

NOTES

The LDAP libraries with the **LDAP_OPT_REFERRALS** option set to **LDAP_OPT_ON** (default value) automatically follow referrals using an anonymous bind. Application developers are encouraged to either implement consistent referral chasing features, or explicitly disable referral chasing by setting that option to **LDAP_OPT_OFF**.

The protocol version used by the library defaults to LDAPv2 (now historic), which corresponds to the **LDAP_VERSION2** macro. Application developers are encouraged to explicitly set **LDAP_OPT_PROTOCOL_VERSION** to LDAPv3, using the **LDAP_VERSION3** macro, or to allow users to select the protocol version.

SEE ALSO

ldap(3), **ldap_error**(3), **RFC 4422** (http://www.rfc-editor.org),

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.