

NAME

ldap - OpenLDAP Lightweight Directory Access Protocol API

LIBRARY

OpenLDAP LDAP (libldap, -lldap)

SYNOPSIS

```
#include <ldap.h>
```

DESCRIPTION

The Lightweight Directory Access Protocol (LDAP) (RFC 4510) provides access to X.500 directory services. These services may be stand-alone or part of a distributed directory service. This client API supports LDAP over TCP (RFC 4511), LDAP over TLS/SSL, and LDAP over IPC (UNIX domain sockets). This API supports SASL (RFC 4513) and Start TLS (RFC 4513) as well as a number of protocol extensions. This API is loosely based upon IETF/LDAPEXT C LDAP API draft specification, a (orphaned) work in progress.

The OpenLDAP Software package includes a stand-alone server in **slapd(8)**, various LDAP clients, and an LDAP client library used to provide programmatic access to the LDAP protocol. This man page gives an overview of the LDAP library routines.

Both synchronous and asynchronous APIs are provided. Also included are various routines to parse the results returned from these routines. These routines are found in the -lldap library.

The basic interaction is as follows. A session handle is created using **ldap_initialize(3)** and set the protocol version to 3 by calling **ldap_set_option(3)**. The underlying session is established first operation is issued. This would generally be a Start TLS or Bind operation, or a Search operation to read attributes of the Root DSE. A Start TLS operation is performed by calling **ldap_start_tls_s(3)**. A LDAP bind operation is performed by calling **ldap_sasl_bind(3)** or one of its friends. A Search operation is performed by calling **ldap_search_ext_s(3)** or one of its friends.

Subsequently, additional operations are performed by calling one of the synchronous or asynchronous routines (e.g., **ldap_compare_ext_s(3)** or **ldap_compare_ext(3)** followed by **ldap_result(3)**). Results returned from these routines are interpreted by calling the LDAP parsing routines such as **ldap_parse_result(3)**. The LDAP association and underlying connection is terminated by calling **ldap_unbind_ext(3)**. Errors can be interpreted by calling **ldap_err2string(3)**.

LDAP versions

This library supports version 3 of the Lightweight Directory Access Protocol (LDAPv3) as defined in RFC 4510. It also supports a variant of version 2 of LDAP as defined by U-Mich LDAP and, to some

degree, RFC 1777. Version 2 (all variants) are considered obsolete. Version 3 should be used instead.

For backwards compatibility reasons, the library defaults to version 2. Hence, all new applications (and all actively maintained applications) should use **ldap_set_option(3)** to select version 3. The library manual pages assume version 3 has been selected.

INPUT and OUTPUT PARAMETERS

All character string input/output is expected to be/is UTF-8 encoded Unicode (version 3.2).

Distinguished names (DN) (and relative distinguished names (RDN) to be passed to the LDAP routines should conform to RFC 4514 UTF-8 string representation.

Search filters to be passed to the search routines are to be constructed by hand and should conform to RFC 4515 UTF-8 string representation.

LDAP URLs to be passed to routines are expected to conform to RFC 4516 format. The **ldap_url(3)** routines can be used to work with LDAP URLs.

LDAP controls to be passed to routines can be manipulated using the **ldap_controls(3)** routines.

DISPLAYING RESULTS

Results obtained from the search routines can be output by hand, by calling **ldap_first_entry(3)** and **ldap_next_entry(3)** to step through the entries returned, **ldap_first_attribute(3)** and **ldap_next_attribute(3)** to step through an entry's attributes, and **ldap_get_values(3)** to retrieve a given attribute's values. Attribute values may or may not be displayable.

UTILITY ROUTINES

Also provided are various utility routines. The **ldap_sort(3)** routines are used to sort the entries and values returned via the ldap search routines.

DEPRECATED INTERFACES

A number of interfaces are now considered deprecated. For instance, **ldap_add(3)** is deprecated in favor of **ldap_add_ext(3)**. Deprecated interfaces generally remain in the library. The macro **LDAP_DEPRECATED** can be defined to a non-zero value (e.g., **-DLLDAP_DEPRECATED=1**) when compiling program designed to use deprecated interfaces. It is recommended that developers writing new programs, or updating old programs, avoid use of deprecated interfaces. Over time, it is expected that documentation (and, eventually, support) for deprecated interfaces to be eliminated.

BER LIBRARY

Also included in the distribution is a set of lightweight Basic Encoding Rules routines. These routines

are used by the LDAP library routines to encode and decode LDAP protocol elements using the (slightly simplified) Basic Encoding Rules defined by LDAP. They are not normally used directly by an LDAP application program except in the handling of controls and extended operations. The routines provide a printf and scanf-like interface, as well as lower-level access. These routines are discussed in **lber-decode(3)**, **lber-encode(3)**, **lber-memory(3)**, and **lber-types(3)**.

INDEX

- ldap_initialize(3) initialize the LDAP library without opening a connection to a server
- ldap_result(3) wait for the result from an asynchronous operation
- ldap_abandon_ext(3) abandon (abort) an asynchronous operation
- ldap_add_ext(3) asynchronously add an entry
- ldap_add_ext_s(3) synchronously add an entry
- ldap_sasl_bind(3) asynchronously bind to the directory
- ldap_sasl_bind_s(3) synchronously bind to the directory
- ldap_unbind_ext(3) synchronously unbind from the LDAP server and close the connection
- ldap_unbind(3) and ldap_unbind_s(3) are equivalent to **ldap_unbind_ext(3)**
- ldap_memfree(3) dispose of memory allocated by LDAP routines.
- ldap_compare_ext(3) asynchronously compare to a directory entry
- ldap_compare_ext_s(3) synchronously compare to a directory entry
- ldap_delete_ext(3) asynchronously delete an entry
- ldap_delete_ext_s(3) synchronously delete an entry
- ld_errno(3) LDAP error indication
- ldap_errlist(3) list of LDAP errors and their meanings

- `ldap_err2string(3)` convert LDAP error indication to a string
- `ldap_extended_operation(3)`
asynchronously perform an arbitrary extended operation
- `ldap_extended_operation_s(3)`
synchronously perform an arbitrary extended operation
- `ldap_first_attribute(3)` return first attribute name in an entry
- `ldap_next_attribute(3)` return next attribute name in an entry
- `ldap_first_entry(3)` return first entry in a chain of search results
- `ldap_next_entry(3)` return next entry in a chain of search results
- `ldap_count_entries(3)` return number of entries in a search result
- `ldap_get_dn(3)` extract the DN from an entry
- `ldap_get_values_len(3)` return an attribute's values with lengths
- `ldap_value_free_len(3)` free memory allocated by `ldap_get_values_len(3)`
- `ldap_count_values_len(3)`
return number of values
- `ldap_modify_ext(3)` asynchronously modify an entry
- `ldap_modify_ext_s(3)` synchronously modify an entry
- `ldap_mods_free(3)` free array of pointers to mod structures used by `ldap_modify_ext(3)`
- `ldap_rename(3)` asynchronously rename an entry
- `ldap_rename_s(3)` synchronously rename an entry
- `ldap_msgfree(3)` free results allocated by `ldap_result(3)`
- `ldap_msgtype(3)` return the message type of a message from `ldap_result(3)`

<code>ldap_msgid(3)</code>	return the message id of a message from <code>ldap_result(3)</code>
<code>ldap_search_ext(3)</code>	asynchronously search the directory
<code>ldap_search_ext_s(3)</code>	synchronously search the directory
<code>ldap_is_ldap_url(3)</code>	check a URL string to see if it is an LDAP URL
<code>ldap_url_parse(3)</code>	break up an LDAP URL string into its components
<code>ldap_sort_entries(3)</code>	sort a list of search results
<code>ldap_sort_values(3)</code>	sort a list of attribute values
<code>ldap_sort_strcasecmp(3)</code>	case insensitive string comparison

SEE ALSO

ldap.conf(5), **slapd(8)**, **draft-ietf-ldapext-ldap-c-api-xx.txt** <<http://www.ietf.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.

These API manual pages are loosely based upon descriptions provided in the IETF/LDAPEXT C LDAP API Internet Draft, a (orphaned) work in progress.