NAME

ldap_search, ldap_search_s, ldap_search_st, ldap_search_ext, ldap_search_ext_s - Perform an LDAP
search operation

LIBRARY

OpenLDAP LDAP (libldap, -lldap)

SYNOPSIS

#include <sys/types.h>
#include <ldap.h>

int ldap_search_ext(
   LDAP *ld**,**
   char ***base,**
   int scope**,**
   char ***filter,**
   char ***attrs[],**
   int attrsonly**,**
   LDAPControl ****serverctrls,**
   LDAPControl ****clientctrls,**
   struct timeval ***timeout,**
   int sizelimit**,**
   int ***msgidp **);

int ldap_search_ext_s(
   LDAP *ld**,**
   char ***base,**
   int scope**,**
   char ***filter,**
   char ***attrs[],**
   int attrsonly**,**
   LDAPControl ****serverctrls,**
   LDAPControl ****clientctrls,**
   struct timeval ***timeout,**
   int sizelimit**,**
   LDAPMessage ****res **);

DESCRIPTION

These routines are used to perform LDAP search operations. The **ldap_search_ext_s()** routine does the
search synchronously (i.e., not returning until the operation completes), providing a pointer to the

resulting LDAP messages at the location pointed to by the *res* parameter.

The **ldap_search_ext()** routine is the asynchronous version, initiating the search and returning the message id of the operation it initiated in the integer pointed to by the *msgidp* parameter.

The *base* parameter is the DN of the entry at which to start the search.

The *scope* parameter is the scope of the search and should be one of LDAP_SCOPE_BASE, to search the object itself, LDAP_SCOPE_ONELEVEL, to search the object's immediate children, LDAP_SCOPE_SUBTREE, to search the object and all its descendants, or LDAP_SCOPE_CHILDREN, to search all of the descendants. Note that the latter requires the server support the LDAP Subordinates Search Scope extension.

The *filter* is a string representation of the filter to apply in the search. The string should conform to the format specified in RFC 4515 as extended by RFC 4526. For instance, "(cn=Jane Doe)". Note that use of the extension requires the server to support the LDAP Absolute True/False Filter extension. NULL may be specified to indicate the library should send the filter (objectClass=*).

The *attrs* parameter is a null-terminated array of attribute descriptions to return from matching entries. If NULL is specified, the return of all user attributes is requested. The description "*" (LDAP_ALL_USER_ATTRIBUTES) may be used to request all user attributes to be returned. The description "+"(LDAP_ALL_OPERATIONAL_ATTRIBUTES) may be used to request all operational attributes to be returned. Note that this requires the server to support the LDAP All Operational Attribute extension. To request no attributes, the description "1.1" (LDAP_NO_ATTRS) should be listed by itself.

The *attrsonly* parameter should be set to a non-zero value if only attribute descriptions are wanted. It should be set to zero (0) if both attributes descriptions and attribute values are wanted.

The *serverctrls* and *clientctrls* parameters may be used to specify server and client controls, respectively.

The **ldap_search_ext_s()** routine is the synchronous version of **ldap_search_ext().**

It also returns a code indicating success or, in the case of failure, indicating the nature of the failure of the operation. See **ldap_error**(3) for details.

NOTES
Note that both read and list functionality are subsumed by these routines, by using a filter like "(objectclass=*)" and a scope of LDAP_SCOPE_BASE (to emulate read) or

LDAP_SCOPE_ONELEVEL (to emulate list).

These routines may dynamically allocate memory. The caller is responsible for freeing such memory using supplied deallocation routines. Return values are contained in <ldap.h>.

Note that *res* parameter of **ldap_search_ext_s()** and **ldap_search_s()** should be freed with **ldap_msgfree()** regardless of return value of these functions.

DEPRECATED INTERFACES

The **ldap_search()** routine is deprecated in favor of the **ldap_search_ext()** routine. The **ldap_search_s()** and **ldap_search_st()** routines are deprecated in favor of the **ldap_search_ext_s()** routine.

Deprecated interfaces generally remain in the library. The macro LDAP_DEPRECATED can be defined to a non-zero value (e.g., -DLDAP_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.

SEE ALSO

ldap(3), **ldap_result**(3), **ldap_error**(3)

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.