NAME

ldap_get_dn, ldap_explode_dn, ldap_explode_rdn, ldap_dn2ufn - LDAP DN handling routines

LIBRARY

OpenLDAP LDAP (libldap, -lldap)

SYNOPSIS

#include <ldap.h>

char *ldap_get_dn(LDAP *ld, LDAPMessage *entry)

int ldap_str2dn(const char *str, LDAPDN *dn, unsigned flags)

void ldap_dnfree(LDAPDN dn)

int ldap_dn2str(LDAPDN dn, char **str, unsigned flags)

char **ldap_explode_dn(const char *dn, int notypes)

char **ldap_explode_rdn(const char *rdn, int notypes)

char *ldap_dn2ufn(const char * dn)

char *ldap_dn2dcedn(const char * dn)

char *ldap_dcedn2dn(const char * dn)

char *ldap_dn2ad_canonical(const char * dn)

DESCRIPTION

These routines allow LDAP entry names (Distinguished Names, or DNs) to be obtained, parsed, converted to a user-friendly form, and tested. A DN has the form described in RFC 4414 "Lightweight Directory Access Protocol (LDAP): String Representation of Distinguished Names".

The **ldap_get_dn()** routine takes an *entry* as returned by **ldap_first_entry**(3) or **ldap_next_entry**(3) and returns a copy of the entry’s DN. Space for the DN will be obtained dynamically and should be freed by the caller using **ldap_memfree**(3).

ldap_str2dn() parses a string representation of a distinguished name contained in **str** into its components, which are stored in **dn** as **ldap_ava** structures, arranged in **LDAPAVA, LDAPRDN,** and

LDAPDN terms. Space for **dn** will be obtained dynamically and should be freed by the caller using **ldap_dnfree**(3). The **LDAPDN** is defined as:

```
typedef struct ldap_ava {
   struct berval la_attr;
   struct berval la_value;
   unsigned la_flags;
} LDAPAVA;

typedef LDAPAVA** LDAPRDN;
typedef LDAPRDN* LDAPDN;
```

The attribute types and the attribute values are not normalized. The **la_flags** can be either **LDAP_AVA_STRING** or **LDAP_AVA_BINARY,** the latter meaning that the value is BER/DER encoded and thus must be represented as, quoting from RFC 4514, " ... an octothorpe character (’#’ ASCII 35) followed by the hexadecimal representation of each of the bytes of the BER encoding of the X.500 AttributeValue." The **flags** parameter to **ldap_str2dn()** can be

LDAP_DN_FORMAT_LDAPV3
LDAP_DN_FORMAT_LDAPV2
LDAP_DN_FORMAT_DCE

which defines what DN syntax is expected (according to RFC 4514, RFC 1779 and DCE, respectively). The format can be *OR*ed to the flags

LDAP_DN_P_NO_SPACES
LDAP_DN_P_NO_SPACE_AFTER_RDN
...
LDAP_DN_PEDANTIC

The latter is a shortcut for all the previous limitations.

LDAP_DN_P_NO_SPACES does not allow extra spaces in the dn; the default is to silently eliminate spaces around AVA separators (’=’), RDN component separators (’+’ for LDAPv3/LDAPv2 or ’,’ for DCE) and RDN separators (’,’ LDAPv3/LDAPv2 or ’/’ for DCE).

LDAP_DN_P_NO_SPACE_AFTER_RDN does not allow a single space after RDN separators.

ldap_dn2str() performs the inverse operation, yielding in **str** a string representation of **dn.** It allows the same values for **flags** as **ldap_str2dn(),** plus

LDAP_DN_FORMAT_UFN
LDAP_DN_FORMAT_AD_CANONICAL

for user-friendly naming (RFC 1781) and AD canonical.

The following routines are viewed as deprecated in favor of **ldap_str2dn()** and **ldap_dn2str().** They are provided to support legacy applications.

The **ldap_explode_dn()** routine takes a DN as returned by **ldap_get_dn()** and breaks it up into its component parts. Each part is known as a Relative Distinguished Name, or RDN. **ldap_explode_dn()** returns a NULL-terminated array, each component of which contains an RDN from the DN. The *notypes* parameter is used to request that only the RDN values be returned, not their types. For example, the DN "cn=Bob, c=US" would return as either { "cn=Bob", "c=US", NULL } or { "Bob", "US", NULL }, depending on whether notypes was 0 or 1, respectively. Assertion values in RDN strings may included escaped characters. The result can be freed by calling **ldap_value_free**(3).

Similarly, the **ldap_explode_rdn()** routine takes an RDN as returned by **ldap_explode_dn(dn,0)** and breaks it up into its "type=value" component parts (or just "value", if the *notypes* parameter is set). Note the value is not unescaped. The result can be freed by calling **ldap_value_free**(3).

ldap_dn2ufn() is used to turn a DN as returned by **ldap_get_dn**(3) into a more user-friendly form, stripping off all type names. See "Using the Directory to Achieve User Friendly Naming" (RFC 1781) for more details on the UFN format. Due to the ambiguous nature of the format, it is generally only used for display purposes. The space for the UFN returned is obtained dynamically and the user is responsible for freeing it via a call to **ldap_memfree**(3).

ldap_dn2dcedn() is used to turn a DN as returned by **ldap_get_dn**(3) into a DCE-style DN, e.g. a string with most-significant to least significant rdns separated by slashes (’/’); rdn components are separated by commas (’,’). Only printable chars (e.g. LDAPv2 printable string) are allowed, at least in this implementation. **ldap_dcedn2dn()** performs the opposite operation. **ldap_dn2ad_canonical()** turns a DN into a AD canonical name, which is basically a DCE dn with attribute types omitted. The trailing domain, if present, is turned in a DNS-like domain. The space for the returned value is obtained dynamically and the user is responsible for freeing it via a call to **ldap_memfree**(3).

ERRORS

If an error occurs in **ldap_get_dn()**, NULL is returned and the **ld_errno** field in the *ld* parameter is set to indicate the error. See **ldap_error**(3) for a description of possible error codes. **ldap_explode_dn()**, **ldap_explode_rdn()**, **ldap_dn2ufn(), ldap_dn2dcedn(), ldap_dcedn2dn(),** and **ldap_dn2ad_canonical()** will return NULL with **errno**(3) set appropriately in case of trouble.

NOTES

These routines dynamically allocate memory that the caller must free.

SEE ALSO

ldap(3), **ldap_error**(3), **ldap_first_entry**(3), **ldap_memfree**(3), **ldap_value_free**(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.