**NAME**

ldap_is_ldap_url, ldap_url_parse, ldap_free_urldesc - LDAP Uniform Resource Locator routines

**LIBRARY**

OpenLDAP LDAP (libldap, -lldap)

**SYNOPSIS**

**#include <ldap.h>**

**int ldap_is_ldap_url( const char *url )**

**int ldap_url_parse( const char *url, LDAPURLDesc **ludpp )**

```
typedef struct ldap_url_desc {
    char *    lud_scheme;   /* URI scheme */
    char *    lud_host;       /* LDAP host to contact */
    int       lud_port;      /* port on host */
    char *    lud_dn;         /* base for search */
    char **   lud_attrs;     /* list of attributes */
    int       lud_scope;     /* a LDAP_SCOPE_... value */
    char *    lud_filter;    /* LDAP search filter */
    char **   lud_exts;       /* LDAP extensions */
    int       lud_crit_exts; /* true if any extension is critical */
    /* may contain additional fields for internal use */
} LDAPURLDesc;
```

**void ldap_free_urldesc( LDAPURLDesc *ludp );**

**DESCRIPTION**

These routines support the use of LDAP URLs (Uniform Resource Locators) as detailed in RFC 4516. LDAP URLs look like this:

**ldap://***hostport*/*dn*[**?***attrs*[**?***scope*[**?***filter*[**?***exts*]]]]

where:
  *hostport* is a host name with an optional ":portnumber"
  *dn* is the search base
  *attrs* is a comma separated list of attributes to request
  *scope* is one of these three strings:
    base one sub (default=base)

*filter* is filter

*exts* are recognized set of LDAP and/or API extensions.

Example:
  ldap://ldap.example.net/dc=example,dc=net?cn,sn?sub?(cn=*)


URLs that are wrapped in angle-brackets and/or preceded by "URL:" are also tolerated.  Alternative LDAP schemes such as ldaps:// and ldapi:// may be parsed using the below routines as well.

**ldap_is_ldap_url()** returns a non-zero value if *url* looks like an LDAP URL (as opposed to some other kind of URL).  It can be used as a quick check for an LDAP URL; the **ldap_url_parse()** routine should be used if a more thorough check is needed.

**ldap_url_parse()** breaks down an LDAP URL passed in *url* into its component pieces.  If successful, zero is returned, an LDAP URL description is allocated, filled in, and *ludpp* is set to point to it.  If an error occurs, a non-zero URL error code is returned.

**ldap_free_urldesc()** should be called to free an LDAP URL description that was obtained from a call to **ldap_url_parse().**

## SEE ALSO
**ldap**(3)
**RFC 4516** <http://www.rfc-editor.org/rfc/rfc4516.txt>

## 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.