

### The `ldapurl` command

The `ldapurl` command allows one to either compose or decompose LDAP URIs.

When invoked with the `-H` option, `ldapurl` extracts the components of the `ldapuri` option argument, un-escaping hex-escaped chars as required. It basically acts as a frontend to the `ldap_url_parse(3)` call. Otherwise, it builds an LDAP URI based on the components passed with the appropriate options, performing the inverse operation. Option `-H` is incompatible with options `-a`, `-b`, `-E`, `-f`, `-H`, `-h`, `-p`, `-S`, and `-s`.

Options:

- `-a attrs`  
Set a comma-separated list of attribute selectors.
- `-b searchbase`  
Set the searchbase.
- `-e [!]ext[=extparam]`  
Specify general extensions with `-e '!'` indicates criticality.  
General extensions:
  - `[!]assert=<filter>` (an RFC 4515 Filter)
  - `!authzid=<authzid>` ("dn:<dn>" or "u:<user>")
  - `[!]bauthzid` (RFC 3829 authzid control)
  - `[!]chaining[=<resolve>[/<cont>]]`
  - `[!]manageDSAit`
  - `[!]noop`
  - `ppolicy`
  - `[!]postread[=<attrs>]` (a comma-separated attribute list)
  - `[!]preread[=<attrs>]` (a comma-separated attribute list)
  - `[!]relax`
  - `sessiontracking`
  - `abandon, cancel, ignore` (SIGINT sends abandon/cancel, or ignores response; if critical, doesn't wait for SIGINT not really controls)
- `-E [!]ext[=extparam]`  
Set URL extensions; incompatible with `-H`.
- `-f filter`  
Set the URL filter. No particular check on conformity with RFC 4515 LDAP filters is performed, but the value is hex-escaped as required.
- `-H ldapuri`  
Specify URI to be exploded.
- `-h ldaphost`  
Set the host.
- `-p ldapport`  
Set the TCP port. If not specified, port 389 is default
- `-S scheme`  
Set the URL scheme. Defaults for other fields, like `ldapport`, may depend on the value of scheme.
- `-s {base|one|sub|children}`  
Specify the scope of the search to be one of base, one, sub, or children to specify a base object, one-level, subtree, or children search. The default is sub. Note: children scope requires LDAPv3 subordinate feature extension.

Output Format:

If the `-H` option is used, the `ldapuri` supplied is exploded in its components, which are printed to standard output in an LDIF-like form.

Otherwise, the URI built using the values passed with the other options is printed to standard output.

Examples:

Single Attribute for Single User:



# Symas OpenLDAP

## Handbook

```
ldapurl -h localhost -p 389 -b dc=example,dc=com -f "(uid=Hung_Nehring)" -a description
```

Returns:

```
ldap://localhost:389/dc=example,dc=com?description??(uid=Hung_Nehring)
```

Single Attribute for Single User with Sub Scope:

```
ldapurl -h localhost -b dc=example,dc=com -s sub -f "(cn=Some One)"
```

Returns:

```
ldap://localhost:389/dc=example,dc=com??sub?(cn=Some%20One)
```

Multiple Attributes for Single User:

```
ldapurl -h localhost -p 389 -b dc=example,dc=com -f "(uid=Hung_Nehring)" -a description,title,ou
```

Returns:

```
ldap://localhost:389/dc=example,dc=com?description,title,ou??(uid=Hung_Nehring)
```

Multiple Attributes for Multiple Users:

```
ldapurl -h localhost -p 389 -b dc=example,dc=com -f "(objectClass=person)" -a description,title,ou
```

Returns:

```
ldap://localhost:389/dc=example,dc=com?description,title,ou??(objectClass=person)
```

Attribute(s) for Filter with Space

```
ldapurl -h localhost -p 389 -b dc=example,dc=com -f "(ou=Product Development)" -a description
```

Returns:

```
ldap://localhost:389/dc=example,dc=com?description??(ou=Product%20Development)
```

Parsing an LDAP URI:

```
ldapurl -H ldap://localhost:389/dc=example,dc=com?description,title,ou??  
(objectClass=person)
```

Returns:

```
scheme: ldap  
host: localhost  
port: 389  
dn: dc=example,dc=com  
selector: description  
selector: title  
selector: ou  
scope: base  
filter: (objectClass=person)
```

Parsing an LDAP URI with a Scope:

```
ldapurl -H ldap://localhost:389/dc=example,dc=com??sub?(cn=Some%20One)
```

Returns:

```
scheme: ldap  
host: localhost  
port: 389  
dn: dc=example,dc=com  
scope: sub  
filter: (cn=Some One)
```