
Ldaptor Documentation

Release 14.0

Tommi Virtanen and Bret Curtis

June 07, 2016

1	What is Ldaptor	1
2	Quick Usage Example	3
3	User's Guide	5
4	Meta	65
5	Indices and tables	69
	Python Module Index	71

What is Ldaptor

Ldaptor is a pure-Python Twisted library that implements:

- LDAP client logic
- separately-accessible LDAP and BER protocol message generation/parsing
- ASCII-format LDAP filter generation and parsing
- LDIF format data generation

Get it from [PyPI](#), find out what's new in the [Changelog](#)!

Quick Usage Example

```
from twisted.internet import reactor, defer
from ldaptor.protocols.ldap import ldapclient, ldapsyntax, ldapconnector

@defer.inlineCallbacks
def example():
    serverip = '192.168.128.21'
    basedn = 'dc=example,dc=com'
    binddn = 'bjensen@example.com'
    bindpw = 'secret'
    query = '(cn=Babs*)'
    c = ldapconnector.LDAPClientCreator(reactor, ldapclient.LDAPClient)
    overrides = {basedn: (serverip, 389)}
    client = yield c.connect(basedn, overrides=overrides)
    yield client.bind(binddn, bindpw)
    o = ldapsyntax.LDAPEntry(client, basedn)
    results = yield o.search(filterText=query)
    for entry in results:
        print entry

if __name__ == '__main__':
    df = example()
    df.addErrback(lambda err: err.printTraceback())
    df.addCallback(lambda _: reactor.stop())
    reactor.run()
```

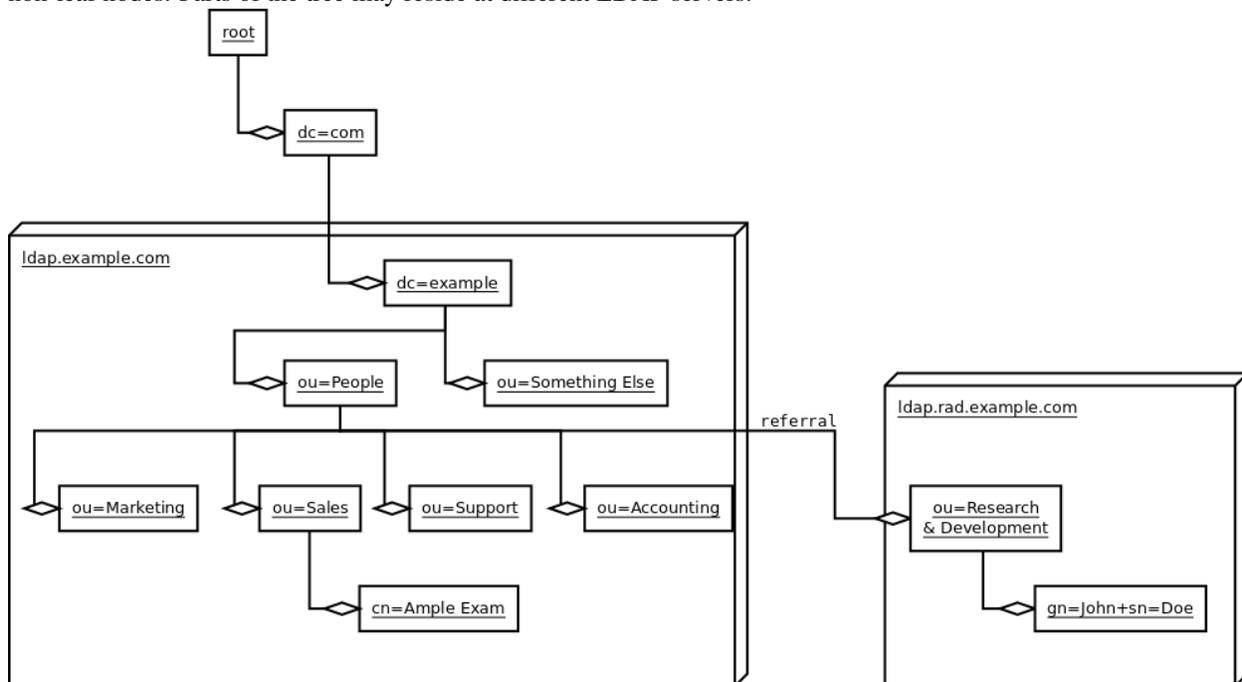

3.1 Introduction to LDAP

3.1.1 Foreword

This text is intended as a quick introduction to the interesting bits of the LDAP protocol, and should be useful whether you are managing an LDAP server, programming something using an LDAP library, or writing an LDAP library yourself. I welcome any feedback you might have.

3.1.2 LDAP Presents a Distributed Tree of Information

Probably the nicest way to get a mental model of LDAP information is to think of a tree with elements both in leaf and non-leaf nodes. Parts of the tree may reside at different LDAP servers.



An organization normally uses their DNS domain name as the root entry for their local LDAP tree. For example, `example.com` is free to use `dc=example`, `dc=com`. The `dc` stands for `domainComponent`. An alternative is to identify the organization via geographical location, as in `o=Example Inc.`, `c=US`, but this is cumbersome as

it requires registration to avoid name conflicts. The `o` stands for organization, `c` for country. You will also encounter `ou`, short for organizational unit.

Each node of the tree is called an “LDAP entry”, and can contain multiple attributes in the form of `attributeType=value` pairs, for example `surname=Wiesel`. One `attributeType` may appear multiple times, in effect having multiple values.

One or more of the attributes are chosen as a Relative Distinguished Name or RDN, and will be used to identify the node based on its parent. This means the RDN must be unique among the children of its parent. Listing all the RDNs, separated by commas, from the node to the root, gives us the Distinguished Name or DN of the entry.

- The RDN of the entry for Jack E. Wiesel is `cn=Jack E. Wiesel`.
- The DN is `cn=Jack E. Wiesel,ou=Sales,ou=People,dc=example,dc=com`.
- The `cn` is short for common name.

The RDN of the entry for John Doe consist of two attributes, `gn=John` and `sn=Doe`, joined with a plus sign to form `gn=John+sn=Doe`. `gn` is short for given name (first name), `sn` for surname (last name).

3.1.3 Objectclasses and Schemas

A special `attributeType` of `objectClass` lists all the objectclasses the LDAP entry manifests. An object class basically lists what attribute types an entry must have, and what optional attribute types it may have. For example, telephone directory entries must have a name and a telephone number, and may have a fax number and street address. `objectClass` can have multiple values, allowing the same entry to describe e.g. information about a person both for a telephone directory and for UNIX shell login.

An LDAP schema is a part of the configuration of the LDAP server, containing two things: definitions of attribute types and definitions of objectclasses. It is normally stored as ASCII text, but can e.g. be requested from the server over an LDAP connection.

An attribute type definition commonly contains a global identifier for the attribute type (a list of period-separated integers), a list of names for the attribute type, a free-form description and a reference to another attribute type this definition inherits from. It may also contain information about what sort of data the attribute values may contain, how to compare and sort them, how to find substrings in the value, whether the attribute type can have multiple values, etc.

An example `attributeType` definition:

```
attributetype ( 2.5.4.4 NAME ( 'sn' 'surname' )
DESC 'RFC2256: last (family) name(s) for which the entity is known by'
SUP name )
```

An object class definition also commonly contains a global identifier, name, description and inheritance information. It also commonly lists the attribute types entries having this object class must have, and additional attribute types they may have. An entry cannot have attribute types that are not listed as a `MUST` or `MAY` by one of the entries object classes or their parents.

An example `objectClass` definition:

```
objectclass ( 2.5.6.6 NAME 'person' DESC 'RFC2256: a person'
SUP top STRUCTURAL MUST ( sn $ cn ) MAY
( userPassword $ telephoneNumber $ seeAlso $ description ) )
```

There are a lot of pre-existing schemas, standardized in various RFCs. Also, anyone can create their own schemas. The only things you need are access to the LDAP server configuration, and a number reserved for you, which can be achieved by filling a web form.

3.1.4 Object-oriented look at LDAP entries

If you look at LDAP entries from the viewpoint of a programmer accustomed with object oriented programming, you will see a lot of similarities, but also some striking differences.

3.1.5 Writing Things Down: LDIF

There is a standardized way of writing down, in plain text, the contents of LDAP directories, individual entries and even add, delete and modify operations. This format is known as LDIF (LDAP Data Interchange Format) LDAP Data Interchange Format, and it is defined in RFC2849.

The rough format of LDIF is this: there is a paragraph per entry, where paragraphs are separated by blank lines. Each paragraph contains lines in the format keyword:value. Entries start by listing the keyword `dn`, and their DN, and then list all the attributes and values the entry has. Lines starting with space are appended to the previous line. The whole file starts with the keyword `version` and value `1`.

Note: The actual format is more complex, but this tutorial should allow you to read and write normal LDIF files fluently.

A simple LDAP file with two entries:

```
version: 1
dn: cn=Barbara Jensen, ou=Product Development, dc=airius, dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
cn: Barbara Jensen
cn: Barbara J Jensen
cn: Babs Jensen
sn: Jensen
uid: bjensen
telephonenumber: +1 408 555 1212
description: A big sailing fan.

dn: cn=Bjorn Jensen, ou=Accounting, dc=airius, dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
cn: Bjorn Jensen
sn: Jensen
telephonenumber: +1 408 555 1212
```

A file containing an entry with a folded attribute value, from [RFC 2849](#):

```
version: 1
dn:cn=Barbara Jensen, ou=Product Development, dc=airius, dc=com
objectclass:top
objectclass:person
objectclass:organizationalPerson
cn:Barbara Jensen
cn:Barbara J Jensen
cn:Babs Jensen
sn:Jensen
uid:bjensen
telephonenumber:+1 408 555 1212
```

```
description:Babs is a big sailing fan, and travels extensively in search of perfect sailing conditions
title:Product Manager, Rod and Reel Division
```

3.1.6 Searches and Search Filters

The most common LDAP operation is a search, and LDAP is purposefully designed for environments where searches are many times more common than modify operations. In general, LDAP servers index the entries and can effectively search for matches against a reasonably complex criteria among thousands of entries.

An LDAP search takes the following information as input:

Note: Once again, we are skipping some details for understandability.

Of these, the search filter is clearly the most interesting one. As with LDIF, search filters have a standardized plain text representation, even though they are not transmitted as plain text in the actual protocol.

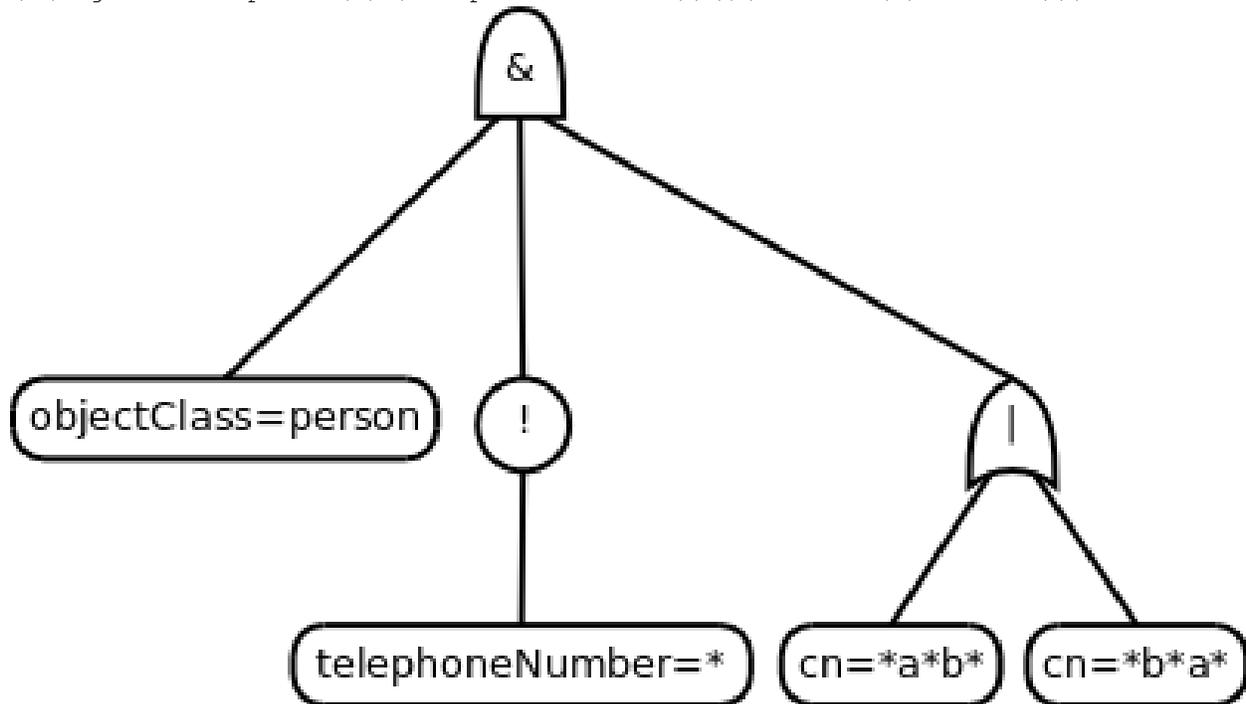
A search filter is basically a combination of tests an entry must fulfill in order to match the filter. They are always written inside parentheses. A simple example would be

```
(cn=John Smith)
```

but the filters can also match against presence, prefix, suffix, substring, rough equality, etc. Multiple matches can be combined freely with and, or and not operators, which are represented by `&`, `|` and `!`, respectively. For example, to match only objects that have objectClass `person`, where the full name contains the letters `a` and `b` in either order, and who don't have a telephone number listed, we could use the filter

Note: Yes, once again we are skipping details for understandability. See RFC2254 for more.

```
(&(objectClass=person)!(telephoneNumber=*)(|(cn=*a*b*)(cn=*b*a*)))
```



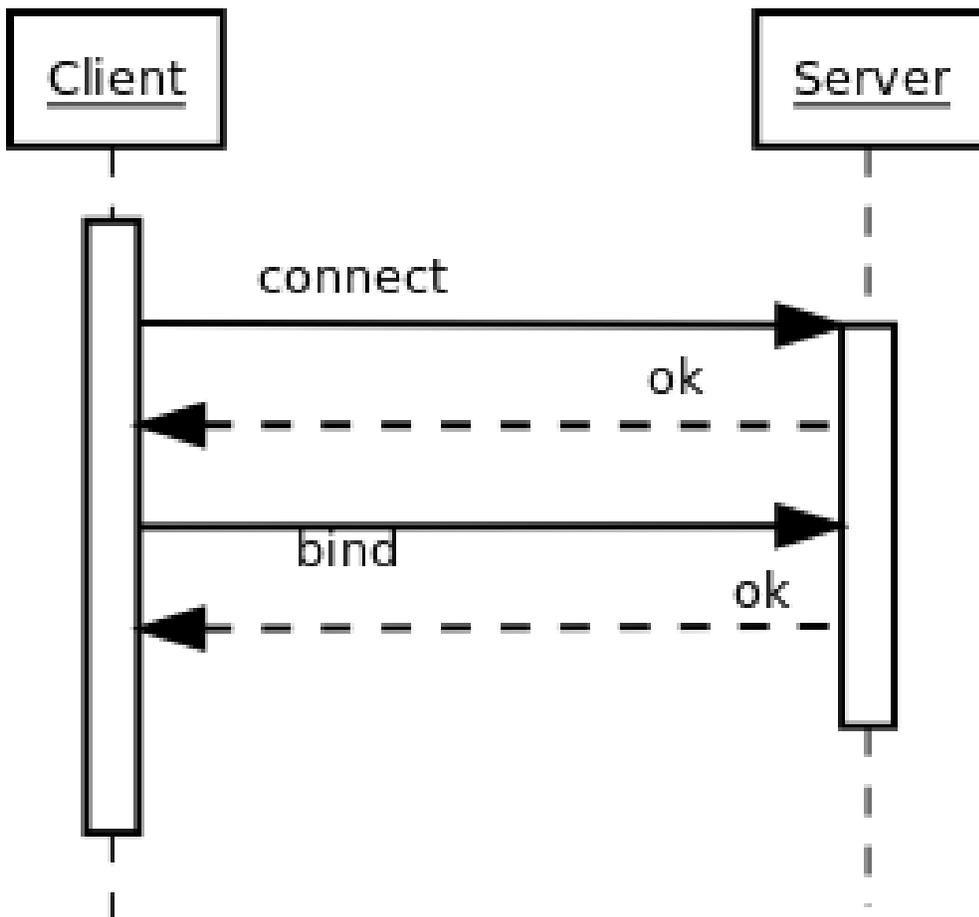
3.1.7 Phases of an LDAP Protocol Chat

An average LDAP protocol chat consists of three stages:

1. Opening the connection
2. Doing one or more searches
3. Closing the connection

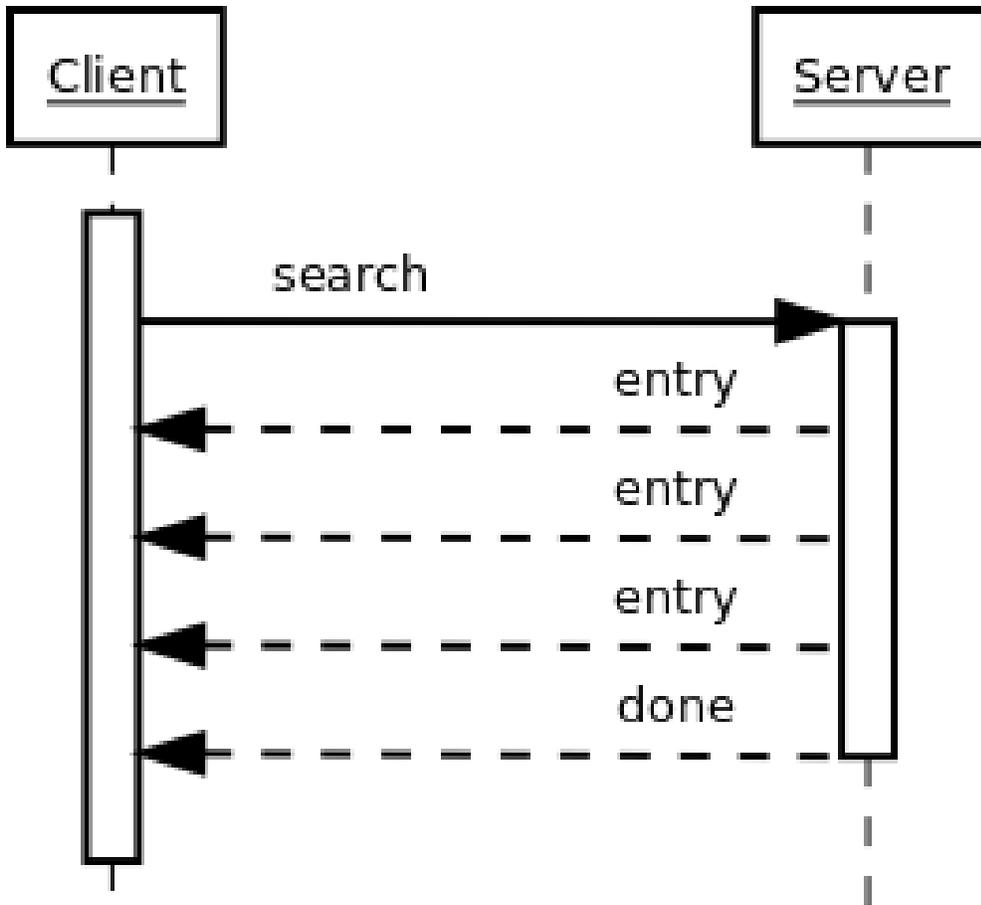
At the first stage, opening a connection, an LDAP client opens a TCP connection to the LDAP server, either as plain text, encrypted by TLS or starting with plaintext and switching to use TLS with STARTTLS.

The client authenticates itself and/or the user, providing any necessary authentication information. This is called binding. Normally, the connection is not really authenticated, but left as anonymous; the bind message is sent with no user or password information.

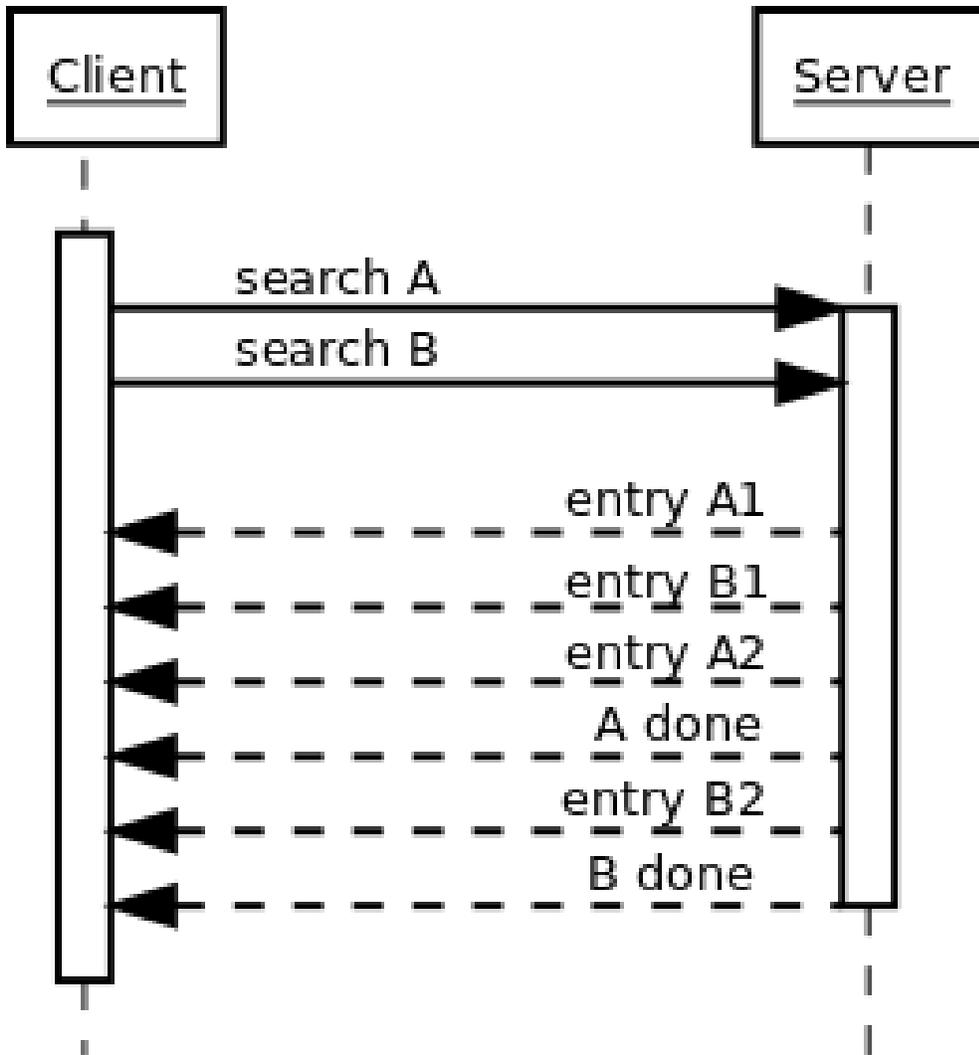


Next, the client sends a search request, containing the base DN for the search, the filter that entries must fulfill to match, and some extra settings discussed above.

The server replies by sending search result entries back, one message per matching entry. If no entry matched or there was an error before the search could even start, the server might not send any entries. Finally, the server sends a message indicating the search is done, and includes information on whether the search was completely successfully, or the error encountered.

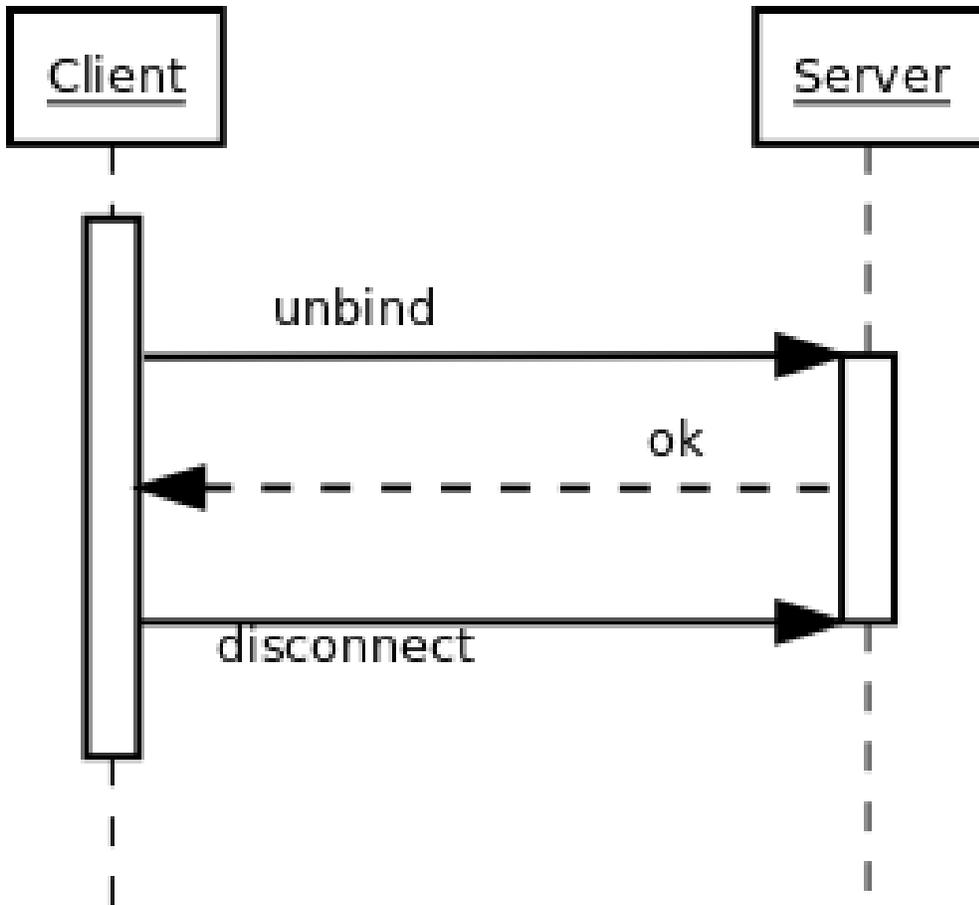


Note that the client could have sent another search request without waiting for the first search to complete. The order of results from the different search, or when they are completed, is in no way guaranteed.



One important detail we have skimmed over so far is how the LDAP client knows what message the server is replying to. Earlier we avoided this topic just by doing only one thing at a time, but now we have two searches getting their result entries interleaved. Clearly, there must be a mechanism to separate which entries belong to which search request. And exactly such a mechanism exists; each message sent by the client contains a number identifying the request, and the server replies by including the same number in the reply. Now, all the client needs to do is remember which numbers are still in use, and not reuse those. It can internally maintain search state based on these numbers, and process result entries based on them. The client can reuse a number when it is known that no more server replies will be sent using that number; for example, the search done message gives this guarantee.

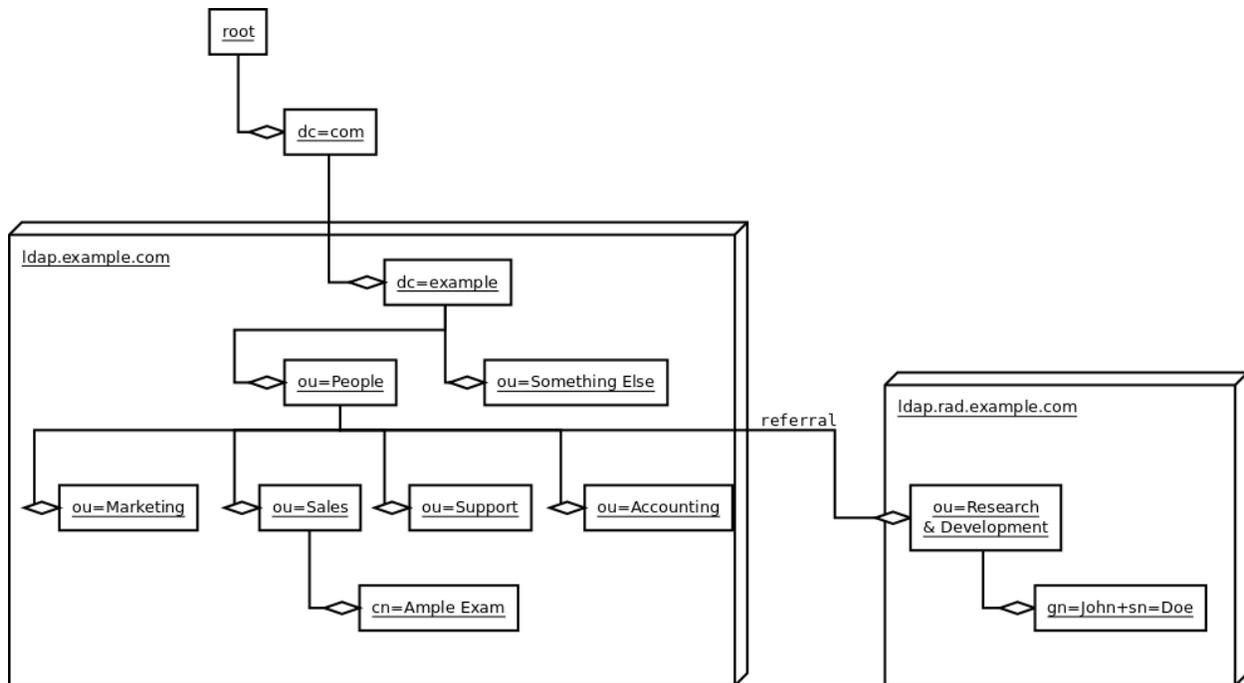
Finally, when the client no longer wants to talk to the server, it sends a message effectively saying “good bye”. This message is known as `unbind`. This only means that the state of connection is the same as when connected, before the first `bind`; that is, it un-authenticates the current user. If the client really wants to close the connection, it will then close the TCP socket.



Please understand that these were just examples, and in reality protocol chats are often more complicated. For example, one could connect some other protocol servers, say a web servers, authentication mechanism to actually act as an LDAP client, that tries to bind as the user authenticating himself to the web server, with the password given by the user. If this service had no other interest in the contents of LDAP, it would probably immediately after the bind close the connection. But opening and closing TCP connections repeatedly is slow; it is quite likely the authentication mechanism would be changed to keep a single TCP connection alive, and just do repeated binds over the same connection.

3.2 Creating a simple LDAP application

3.2.1 LDAP presents a distributed tree of information



Writing things down, John Doe LDIF:

```
dn: gn=John+sn=Doe,ou=Research & Development,ou=People,dc=example,dc=com
objectClass: addressbookPerson
gn: John
sn: Doe
street: Back alley
postOfficeBox: 123
postalCode: 54321
postalAddress: Backstreet
st: NY
l: New York City
c: US
```

Writing things down, John Smith LDIF:

```
dn: gn=John+sn=Smith,ou=Marketing,ou=People, dc=example,dc=com
objectClass: addressbookPerson
gn: John
sn: Smith
telephoneNumber: 555-1234
facsimileTelephoneNumber: 555-1235
description: This is a description that can span multi
  ple lines as long as the non-first lines are inden
  ted in the LDIF.
```

3.2.2 Setting up an LDAP server in 5 seconds

Python, an easy programming language

Batteries included!

Python combines remarkable power with very clear syntax.

Runs on many brands of UNIX, on Windows, OS/2, Mac, Amiga, and many other platforms.

3.2.3 The first step

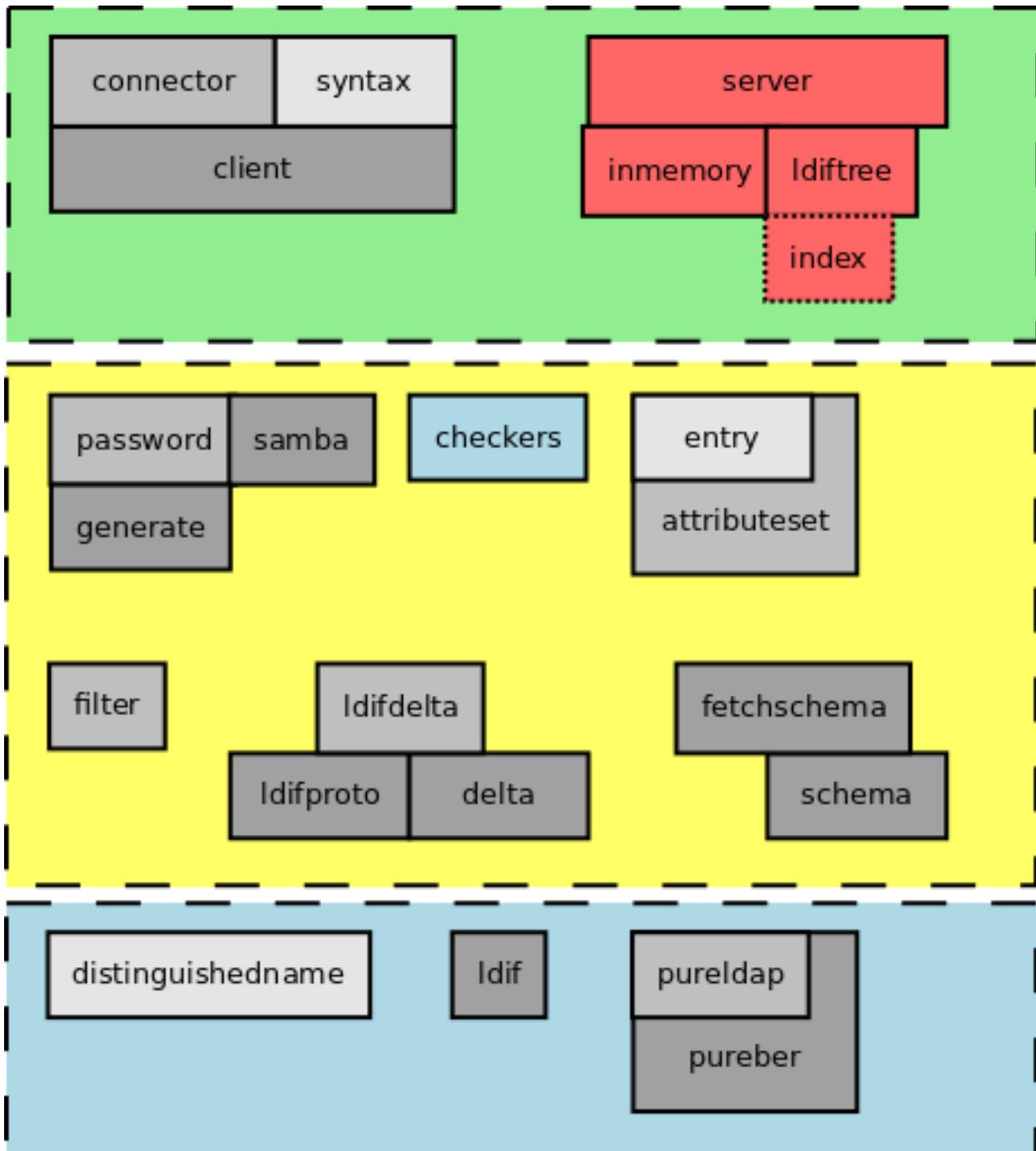
```
>>> from ldaptor.protocols.ldap import distinguishedname
>>> dn=distinguishedname.DistinguishedName (
...   'dc=example,dc=com')
>>> dn
DistinguishedName (listOfRDNs=(RelativeDistinguishedName (
attributeTypesAndValues=(LDAPAttributeTypeAndValue (
attributeType='dc', value='example'),)),
RelativeDistinguishedName (attributeTypesAndValues=(
LDAPAttributeTypeAndValue (attributeType='dc', value='com'),))))
>>> str (dn)
'dc=example,dc=com'
```

3.2.4 Ldaptor

Ldaptor is a set of pure-Python LDAP client programs, applications and a programming library.

It is licensed under the MIT (Expat) License.

3.2.5 Overview of Ldaptor



3.2.6 Preparing to connect

```
>>> from ldaptor.protocols.ldap import \
... ldapclient, ldapconnector
>>> from twisted.internet import reactor
>>> connector=ldapconnector.LDAPClientCreator(reactor,
... ldapclient.LDAPClient)
>>> connector
```

```
<ldaptor.protocols.ldap.ldapconnector.LDAPClientCreator
instance at 0x40619b6c>
```

3.2.7 Twisted

Twisted is an event-driven networking framework written in Python and licensed under the MIT (Expat) License.

Twisted supports TCP, UDP, SSL/TLS, multicast, Unix sockets, a large number of protocols (including HTTP, NNTP, SSH, IRC, FTP, and others), and much more.

Twisted includes many full-blown applications, such as web, SSH, FTP, DNS and news servers.

3.2.8 Connecting

```
>>> d=connector.connectAnonymously(dn,
... {dn: ('localhost', 10389)})
>>> d
<Deferred at 0x402d058c>
```

3.2.9 Deferreds

- A promise that a function will at some point have a result.
- You can attach callback functions to a Deferred.
- Once it gets a result these callbacks will be called.
- Also allows you to register a callback for an error, with the default behavior of logging the error.
- Standard way to handle all sorts of blocking or delayed operations.

3.2.10 Searching

```
>>> from twisted.trial.util import deferredResult
>>> proto=deferredResult(d)
>>> proto
<ldaptor.protocols.ldap.ldapclient.LDAPClient
instance at 0x40619dac>
>>> from ldaptor.protocols.ldap import ldapsyntax
>>> baseEntry=ldapsyntax.LDAPEntry(client=proto, dn=dn)
>>> d2=baseEntry.search(filterText='(gn=j*)')
>>> results=deferredResult(d2)
```

3.2.11 Results

```
>>> results
[LDAPEntry(dn='givenName=John+sn=Smith,ou=People,
dc=example,dc=com', attributes={'description': ['Some text.'],
'facsimileTelephoneNumber': ['555-1235'], 'givenName': ['John'],
'objectClass': ['addressbookPerson'], 'sn': ['Smith'],
'telephoneNumber': ['555-1234']}), LDAPEntry(dn=
'givenName=John+sn=Doe,ou=People,dc=example,dc=com',
attributes={'c': ['US'], 'givenName': ['John'], 'l': ['New York City'],
```

```
'objectClass': ['addressbookPerson'], 'postOfficeBox': ['123'],
'postalAddress': ['Backstreet'], 'postalCode': ['54321'],
'sn': ['Doe'], 'st': ['NY'], 'street': ['Back alley']}]])
```

3.2.12 Results one-by-one

```
>>> results[0]
LDAPEntry(dn=
'givenName=John+sn=Smith,ou=People,dc=example,dc=com',
attributes={'description': ['Some text.'],
'facsimileTelephoneNumber': ['555-1235'], 'givenName': ['John'],
'objectClass': ['addressbookPerson'], 'sn': ['Smith'],
'telephoneNumber': ['555-1234']})
>>> results[3]
Traceback (most recent call last):
  File "<stdin>", line 1, in ?
IndexError: list index out of range
```

3.2.13 LDIF output

```
>>> print results[0]
dn: givenName=John+sn=Smith,ou=People,dc=example,dc=com
objectClass: addressbookPerson
description: Some text.
facsimileTelephoneNumber: 555-1235
givenName: John
sn: Smith
telephoneNumber: 555-1234
```

3.2.14 Closing the connection

```
>>> proto.unbind()
```

3.2.15 Access to entry details

```
>>> smith=results[0]
>>> print smith.dn
givenName=John+sn=Smith,ou=People,dc=example,dc=com
>>> smith['givenName']
['John']
>>>
```

3.2.16 Object-oriented look at LDAP entries

A lot of similarities with OO programming languages, but some big differences, too.

An LDAP entry corresponds with an object.

Whereas object are usually instances of a single class, LDAP entries can “implement” multiple objectClasses.

All objectClasses can inherit zero, one or many objectClasses, just like programming classes.

All objectClasses have a root class, known as *top*; many object oriented programming languages have a root class, e.g. named *Object*.

All objectClasses are either *STRUCTURAL* or *AUXILIARY*; entries can only implement one *STRUCTURAL* objectClass.

Lastly, objectClasses of an entry can be changed at will; you only need to take care that the entry has all the *MUST* attribute types, and no attribute types outside of the ones that are *MUST* or *MAY*.

Note: Note that e.g. OpenLDAP doesn't implement this.

Attributes of an entry closely match attributes of objects in programming languages; however, LDAP attributes may have multiple values.

3.2.17 Search inputs

An example search filter: `(cn=John Smith)`

A search filter, specifying criteria an entry must fulfill to match.

Scope of the search, either look at the base DN only, only look one level below it, or look at the whole subtree rooted at the base DN.

Size limit of at most how many matching entries to return.

Attributes to return, or none for all attributes the matching entries happen to have.

3.2.18 Our first Python program

```
#!/usr/bin/python
from twisted.internet import reactor, defer

from ldaptor.protocols.ldap import ldapclient, ldapsyntax, ldapconnector, \
    distinguishedname
from ldaptor import ldapfilter

def search(config):
    c=ldapconnector.LDAPClientCreator(reactor, ldapclient.LDAPClient)
    d=c.connectAnonymously(config['base'],
                           config['serviceLocationOverrides'])

    def _doSearch(proto, config):
        searchFilter = ldapfilter.parseFilter('(gn=j*)')
        baseEntry = ldapsyntax.LDAPEntry(client=proto, dn=config['base'])
        d=baseEntry.search(filterObject=searchFilter)
        return d

    d.addCallback(_doSearch, config)
    return d

def main():
    import sys
    from twisted.python import log
    log.startLogging(sys.stderr, setStdout=0)

    config = {
```

```
'base':
    distinguishedname.DistinguishedName('ou=People,dc=example,dc=com'),
'serviceLocationOverrides': {
    distinguishedname.DistinguishedName('dc=example,dc=com'):
        ('localhost', 10389),
    }
}

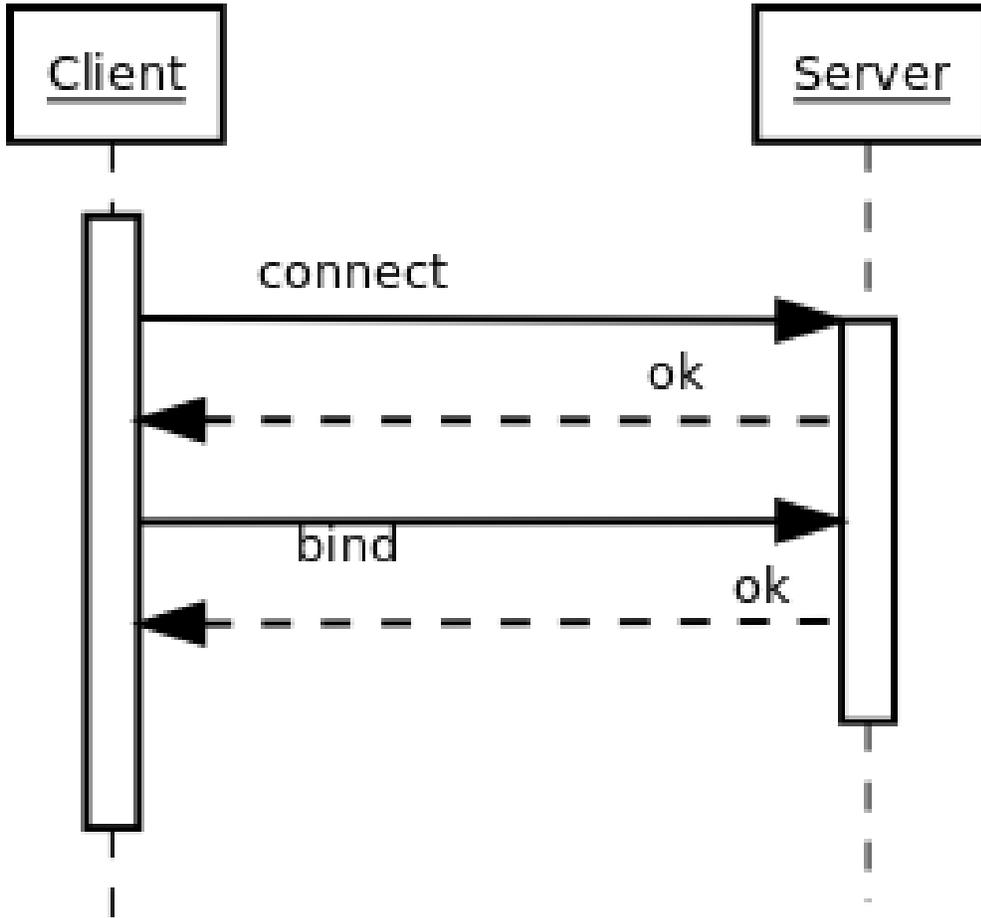
d = search(config)
def _show(results):
    for item in results:
        print item
d.addCallback(_show)
d.addErrback(defer.logError)
d.addBoth(lambda _: reactor.stop())
reactor.run()

if __name__ == '__main__':
    main()
```

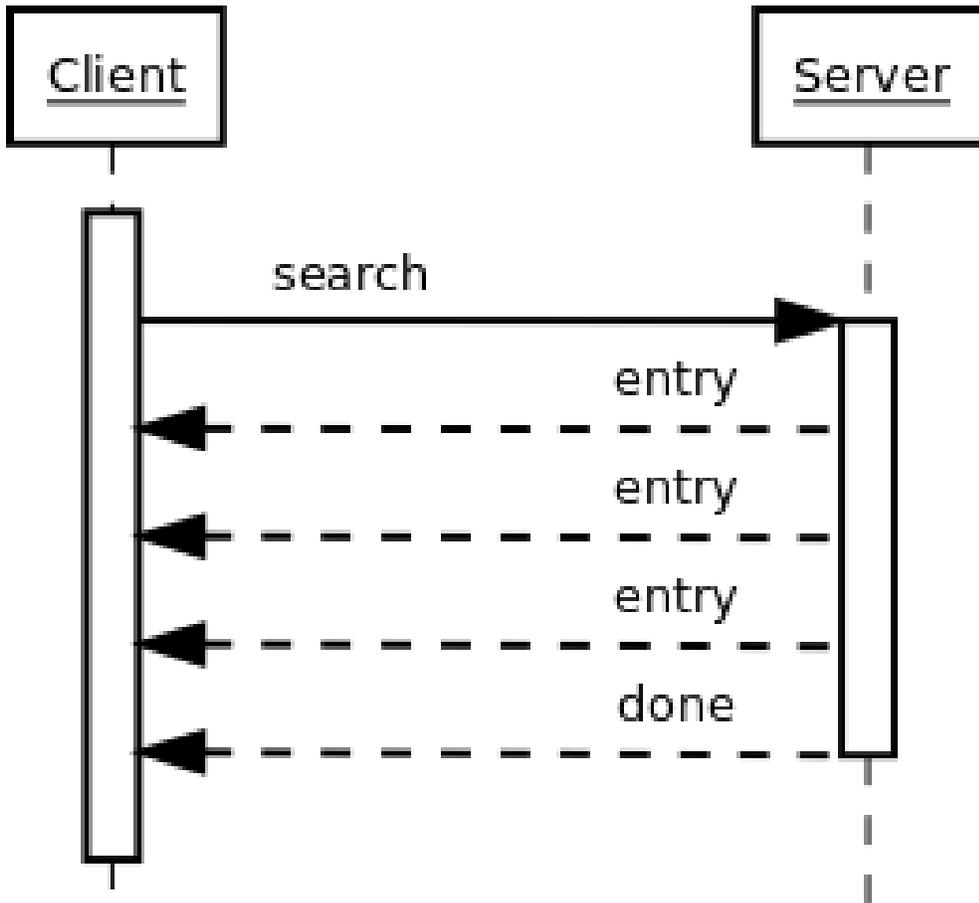
3.2.19 Phases of the protocol chat

- Open and bind
- Search (possibly many times)
- Unbind and close

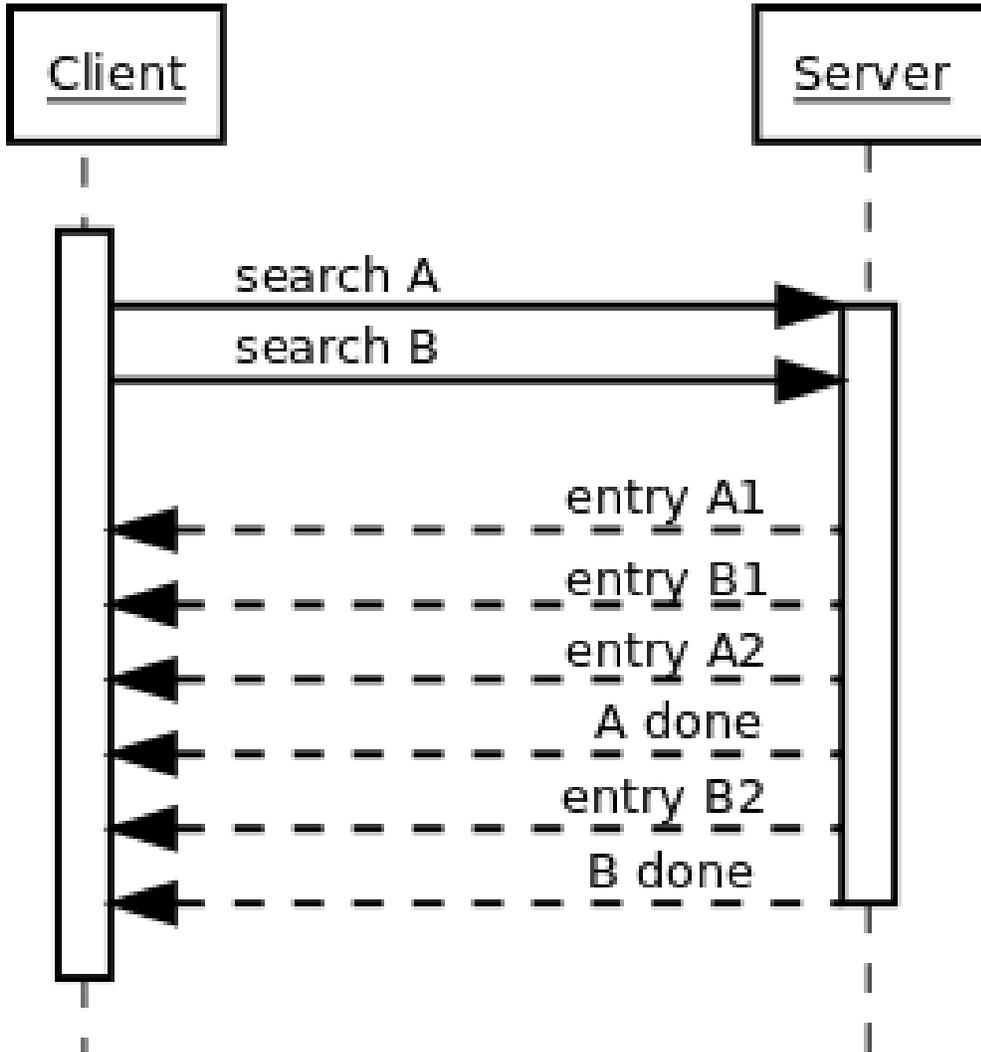
3.2.20 Opening and binding



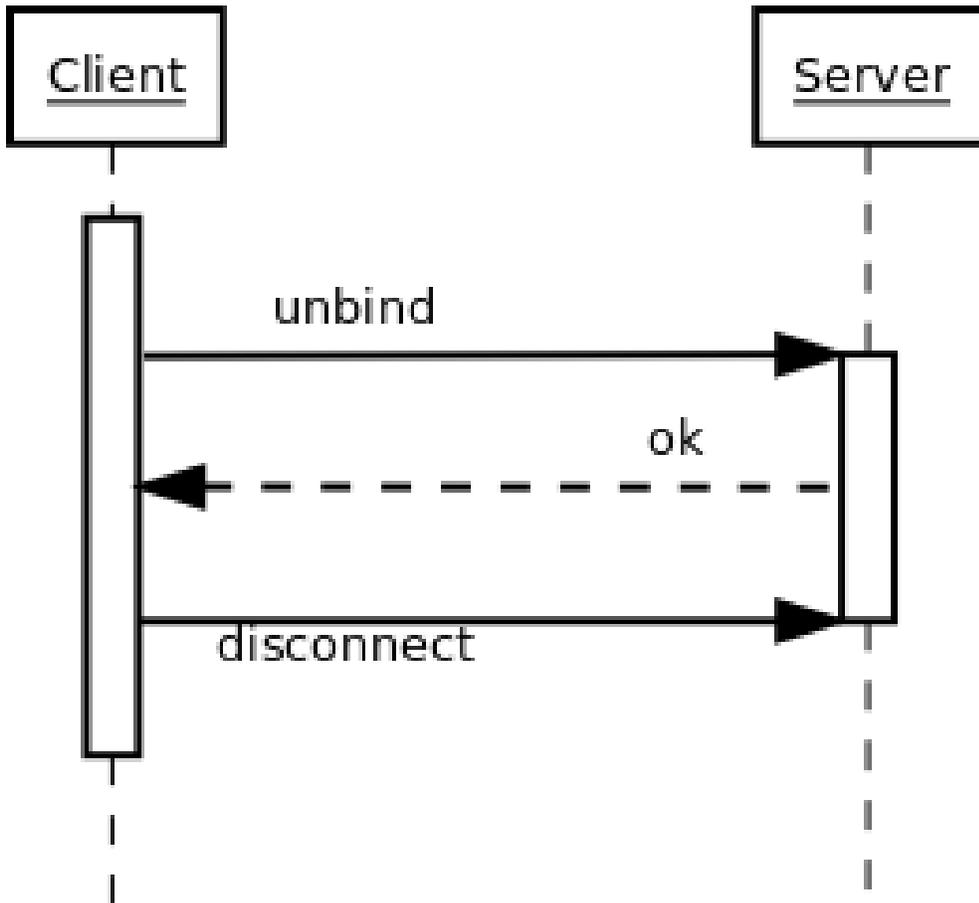
3.2.21 Doing a search



3.2.22 Doing multiple searches



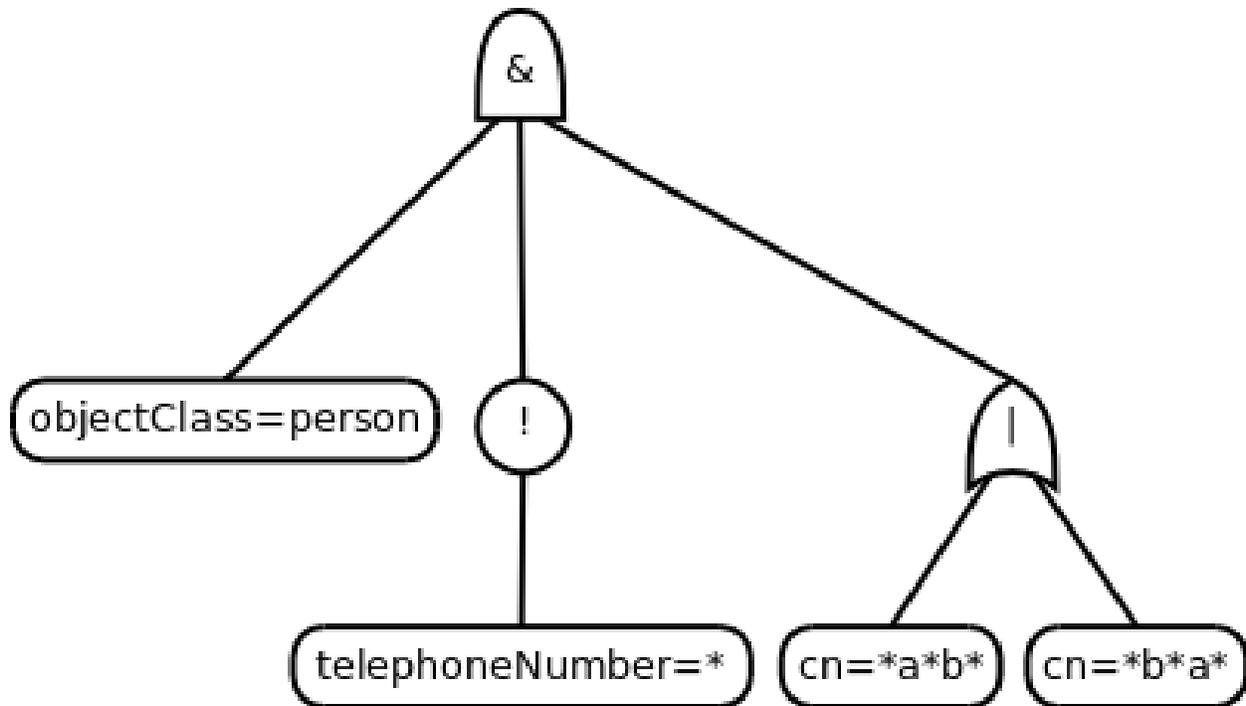
3.2.23 Unbinding and closing



3.2.24 A complex search filter

An example:

```
(&(objectClass=person)
  (!(telephoneNumber=*))
  (|(cn=*a*b*)(cn=*b*a*)))
```



3.2.25 Object classes

1. Special attribute `objectClass` lists all the objectclasses an LDAP entry manifests.
2. **Objectclass defines**
 - (a) What attributetypes an entry **MUST** have
 - (b) What attributetypes an entry **MAY** have
3. An entry in a phonebook must have a name and a telephone number, and may have a fax number and street address.

3.2.26 Schema

1. A configuration file included in the LDAP server configuration.
2. A combination of attribute type and object class definitions.
3. Stored as plain text
4. Can be requested over an LDAP connection

3.2.27 Attribute type

An example:

```

attributetype ( 2.5.4.4 NAME ( 'sn' 'surname' )
  DESC 'RFC2256: last (family) name(s) for which the entity is known by'
  SUP name )

```

Can also contain:

1. content data type
2. comparison and sort mechanism
3. substring search mechanism
4. whether multiple values are allowed

3.2.28 Object class

An example:

```
objectclass ( 2.5.6.6 NAME 'person'
  DESC 'RFC2256: a person'
  SUP top STRUCTURAL
  MUST ( sn $ cn )
  MAY ( userPassword $ telephoneNumber
    $ seeAlso $ description )
)
```

3.2.29 Creating schemas

1. Anyone can create their own schema
2. Need to be globally unique
3. But try to use already existing ones

3.2.30 Where to go from here?

Install OpenLDAP: <http://www.openldap.org/>

Install Ldaptor: <https://github.com/twisted/ldaptor>

Learn Python: <http://www.python.org/>

Learn Twisted. Write a client application for a simple protocol. Read the HOWTOs:
<http://twistedmatrix.com/documents/current/core/howto/clients.html>

3.3 Idaptor API Reference

3.3.1 Subpackages

ldaptor.protocols package

Subpackages

ldaptor.protocols.ldap package

Subpackages

ldaptor.protocols.ldap.autofill package

Submodules

ldaptor.protocols.ldap.autofill.posixAccount module

class `ldaptor.protocols.ldap.autofill.posixAccount.Autofill_posix` (*baseDN*,
freeNumberGetter=<function
getFreeNumber>)

notify (*ldapObject*, *attributeType*)

start (*ldapObject*)

ldaptor.protocols.ldap.autofill.sambaAccount module

class `ldaptor.protocols.ldap.autofill.sambaAccount.Autofill_samba`

notify (*ldapObject*, *attributeType*)

start (*ldapObject*)

ldaptor.protocols.ldap.autofill.sambaSamAccount module

class `ldaptor.protocols.ldap.autofill.sambaSamAccount.Autofill_samba` (*domainSID*,
fixedPriorityGroupSID=None)

notify (*ldapObject*, *attributeType*)

start (*ldapObject*)

Module contents LDAP object field value suggestion and autoupdate mechanism.

exception `ldaptor.protocols.ldap.autofill.AutofillException`

Bases: `exceptions.Exception`

exception `ldaptor.protocols.ldap.autofill.ObjectMissingObjectClassException`

Bases: `ldaptor.protocols.ldap.autofill.AutofillException`

The LDAPEntry is missing an objectClass this autofiller needs to operate.

Submodules

ldaptor.protocols.ldap.distinguishedname module

class `ldaptor.protocols.ldap.distinguishedname.DistinguishedName` (*magic*=None,
stringValue=None,
listOfRDNs=None)

LDAP Distinguished Name.

contains (*other*)

Does the tree rooted at DN contain or equal the other DN.

```

    getDomainName ()
    listOfRDNs = None
    split ()
    up ()
exception ldaptor.protocols.ldap.distinguishedname.InvalidRelativeDistinguishedName (rdn)
    Bases: exceptions.Exception
    Invalid relative distinguished name.

class ldaptor.protocols.ldap.distinguishedname.LDAPAttributeTypeAndValue (stringValue=None,
                                                                    at-
                                                                    tribute-
                                                                    Type=None,
                                                                    value=None)

    attributeType = None
    value = None

class ldaptor.protocols.ldap.distinguishedname.RelativeDistinguishedName (magic=None,
                                                                    string-
                                                                    Value=None,
                                                                    at-
                                                                    tribute-
                                                                    Type-
                                                                    sAnd-
                                                                    Val-
                                                                    ues=None)

    LDAP Relative Distinguished Name.
    attributeTypesAndValues = None
    count ()
    split ()

ldaptor.protocols.ldap.distinguishedname.escape (s)
ldaptor.protocols.ldap.distinguishedname.unescape (s)

```

ldaptor.protocols.ldap.fetchschema module

```
ldaptor.protocols.ldap.fetchschema.fetch (client, baseObject)
```

ldaptor.protocols.ldap.Ldapclient module LDAP protocol client

```
class ldaptor.protocols.ldap.ldapclient.LDAPClient
```

```
    Bases: twisted.internet.protocol.Protocol
```

```
    An LDAP client
```

```
    berdecoder = <LDAPBERDecoderContext_TopLevel identities={0x10: LDAPMessage} fallback=None inherit=<LDAP
```

```
    bind (dn='', auth='')
```

```
        @deprecated: Use e.bind(auth).
```

```
        @todo: Remove this method when there are no callers.
```

connectionLost (*reason*=<*twisted.python.failure.Failure* *twisted.internet.error.ConnectionDone*:
Connection was closed cleanly.>)
Called when TCP connection has been lost

connectionMade ()
TCP connection has opened

dataReceived (*recd*)

debug = False

handle (*msg*)

send (*op*)
Send an LDAP operation to the server.

@param op: the operation to send

@type op: LDAPProtocolRequest

@return: the response from server

@rtype: Deferred LDAPProtocolResponse

send_multiResponse (*op*, *handler*, **args*, ***kwargs*)
Send an LDAP operation to the server, expecting one or more responses.

@param op: the operation to send

@type op: LDAPProtocolRequest

@param handler: a callable that will be called for each response. It should return a boolean, whether this was the final response.

@param args: positional arguments to pass to handler

@param kwargs: keyword arguments to pass to handler

@return: the result from the last handler as a deferred that completes when the last response has been received

@rtype: Deferred LDAPProtocolResponse

send_noResponse (*op*)
Send an LDAP operation to the server, with no response expected.

@param op: the operation to send @type op: LDAPProtocolRequest

startTLS (*ctx*=None)

Start Transport Layer Security.

It is the callers responsibility to make sure other things are not happening at the same time.

@todo: server hostname check, see rfc2830 section 3.6.

unbind ()

unsolicitedNotification (*msg*)

exception `ldaptor.protocols.ldap.ldapclient.LDAPClientConnectionLostException` (*message*=None)
Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

exception `ldaptor.protocols.ldap.ldapclient.LDAPStartTLSBusyError` (*onwire*, *message*=None)
Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPOperationsError`

ldaptor.protocols.ldap.ldapconnector module

class `ldaptor.protocols.ldap.ldapconnector.LDAPClientCreator` (*reactor, protocolClass, *args, **kwargs*)

Bases: `twisted.internet.protocol.ClientCreator`

connect (*dn, overrides=None, bindAddress=None*)

Connect to remote host, return Deferred of resulting protocol instance.

connectAnonymously (*dn, overrides=None*)

Connect to remote host and bind anonymously, return Deferred of resulting protocol instance.

class `ldaptor.protocols.ldap.ldapconnector.LDAPConnector` (*reactor, dn, factory, overrides=None, bindAddress=None*)

Bases: `twisted.names.srvconnect.SRVConnector`

connect ()

pickServer ()

ldaptor.protocols.ldap.ldaperrors module

exception `ldaptor.protocols.ldap.ldaperrors.LDAPAdminLimitExceeded` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

name = 'adminLimitExceeded'

resultCode = 11

exception `ldaptor.protocols.ldap.ldaperrors.LDAPAffectsMultipleDSAs` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

name = 'affectsMultipleDSAs'

resultCode = 71

exception `ldaptor.protocols.ldap.ldaperrors.LDAPAliasDereferencingProblem` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

name = 'aliasDereferencingProblem'

resultCode = 36

exception `ldaptor.protocols.ldap.ldaperrors.LDAPAliasProblem` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

name = 'aliasProblem'

resultCode = 33

exception `ldaptor.protocols.ldap.ldaperrors.LDAPAttributeOrValueExists` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

name = 'attributeOrValueExists'

resultCode = 20

exception `ldaptor.protocols.ldap.ldaperrors.LDAPAuthMethodNotSupported` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

name = 'authMethodNotSupported'

resultCode = 7

exception `ldaptor.protocols.ldap.ldaperrors.LDAPBusy` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

name = 'busy'

resultCode = 51

exception `ldaptor.protocols.ldap.ldaperrors.LDAPCompareFalse` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

name = 'compareFalse'

resultCode = 5

exception `ldaptor.protocols.ldap.ldaperrors.LDAPCompareTrue` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

name = 'compareTrue'

resultCode = 6

exception `ldaptor.protocols.ldap.ldaperrors.LDAPConfidentialityRequired` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

name = 'confidentialityRequired'

resultCode = 13

exception `ldaptor.protocols.ldap.ldaperrors.LDAPConstraintViolation` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

name = 'constraintViolation'

resultCode = 19

exception `ldaptor.protocols.ldap.ldaperrors.LDAPEntryAlreadyExists` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

name = 'entryAlreadyExists'

resultCode = 68

exception `ldaptor.protocols.ldap.ldaperrors.LDAPException` (*message=None*)

Bases: `exceptions.Exception`, `ldaptor.protocols.ldap.ldaperrors.LDAPResult`

message

exception `ldaptor.protocols.ldap.ldaperrors.LDAPInappropriateAuthentication` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

name = 'inappropriateAuthentication'

resultCode = 48

exception `ldaptor.protocols.ldap.ldaperrors.LDAPInappropriateMatching` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

name = 'inappropriateMatching'

resultCode = 18

exception `ldaptor.protocols.ldap.ldaperrors.LDAPInsufficientAccessRights` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

name = 'insufficientAccessRights'

resultCode = 50

exception `ldaptor.protocols.ldap.ldaperrors.LDAPInvalidAttributeSyntax` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

name = 'invalidAttributeSyntax'

```
    resultCode = 21
exception ldaptor.protocols.ldap.ldaperrors.LDAPInvalidCredentials (message=None)
    Bases: ldaptor.protocols.ldap.ldaperrors.LDAPException
    name = 'invalidCredentials'
    resultCode = 49
exception ldaptor.protocols.ldap.ldaperrors.LDAPInvalidDNyntax (message=None)
    Bases: ldaptor.protocols.ldap.ldaperrors.LDAPException
    name = 'invalidDNyntax'
    resultCode = 34
exception ldaptor.protocols.ldap.ldaperrors.LDAPLoopDetect (message=None)
    Bases: ldaptor.protocols.ldap.ldaperrors.LDAPException
    name = 'loopDetect'
    resultCode = 54
exception ldaptor.protocols.ldap.ldaperrors.LDAPNamingViolation (message=None)
    Bases: ldaptor.protocols.ldap.ldaperrors.LDAPException
    name = 'namingViolation'
    resultCode = 64
exception ldaptor.protocols.ldap.ldaperrors.LDAPNoSuchAttribute (message=None)
    Bases: ldaptor.protocols.ldap.ldaperrors.LDAPException
    name = 'noSuchAttribute'
    resultCode = 16
exception ldaptor.protocols.ldap.ldaperrors.LDAPNoSuchObject (message=None)
    Bases: ldaptor.protocols.ldap.ldaperrors.LDAPException
    name = 'noSuchObject'
    resultCode = 32
exception ldaptor.protocols.ldap.ldaperrors.LDAPNotAllowedOnNonLeaf (message=None)
    Bases: ldaptor.protocols.ldap.ldaperrors.LDAPException
    name = 'notAllowedOnNonLeaf'
    resultCode = 66
exception ldaptor.protocols.ldap.ldaperrors.LDAPNotAllowedOnRDN (message=None)
    Bases: ldaptor.protocols.ldap.ldaperrors.LDAPException
    name = 'notAllowedOnRDN'
    resultCode = 67
exception ldaptor.protocols.ldap.ldaperrors.LDAPObjectClassModsProhibited (message=None)
    Bases: ldaptor.protocols.ldap.ldaperrors.LDAPException
    name = 'objectClassModsProhibited'
    resultCode = 69
exception ldaptor.protocols.ldap.ldaperrors.LDAPObjectClassViolation (message=None)
    Bases: ldaptor.protocols.ldap.ldaperrors.LDAPException
```

name = 'objectClassViolation'

resultCode = 65

exception ldaptor.protocols.ldap.ldaperrors.**LDAPOperationsError** (*message=None*)

Bases: *ldaptor.protocols.ldap.ldaperrors.LDAPException*

name = 'operationsError'

resultCode = 1

exception ldaptor.protocols.ldap.ldaperrors.**LDAPOther** (*message=None*)

Bases: *ldaptor.protocols.ldap.ldaperrors.LDAPException*

name = 'other'

resultCode = 80

exception ldaptor.protocols.ldap.ldaperrors.**LDAPProtocolError** (*message=None*)

Bases: *ldaptor.protocols.ldap.ldaperrors.LDAPException*

name = 'protocolError'

resultCode = 2

exception ldaptor.protocols.ldap.ldaperrors.**LDAPReferral** (*message=None*)

Bases: *ldaptor.protocols.ldap.ldaperrors.LDAPException*

name = 'referral'

resultCode = 10

class ldaptor.protocols.ldap.ldaperrors.**LDAPResult**

name = None

resultCode = None

exception ldaptor.protocols.ldap.ldaperrors.**LDAPSaslBindInProgress** (*message=None*)

Bases: *ldaptor.protocols.ldap.ldaperrors.LDAPException*

name = 'saslBindInProgress'

resultCode = 14

exception ldaptor.protocols.ldap.ldaperrors.**LDAPSizeLimitExceeded** (*message=None*)

Bases: *ldaptor.protocols.ldap.ldaperrors.LDAPException*

name = 'sizeLimitExceeded'

resultCode = 4

exception ldaptor.protocols.ldap.ldaperrors.**LDAPStrongAuthRequired** (*message=None*)

Bases: *ldaptor.protocols.ldap.ldaperrors.LDAPException*

name = 'strongAuthRequired'

resultCode = 8

exception ldaptor.protocols.ldap.ldaperrors.**LDAPTimeLimitExceeded** (*message=None*)

Bases: *ldaptor.protocols.ldap.ldaperrors.LDAPException*

name = 'timeLimitExceeded'

resultCode = 3

exception `ldaptor.protocols.ldap.ldaperrors.LDAPUnavailable` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

name = 'unavailable'

resultCode = 52

exception `ldaptor.protocols.ldap.ldaperrors.LDAPUnavailableCriticalExtension` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

name = 'unavailableCriticalExtension'

resultCode = 12

exception `ldaptor.protocols.ldap.ldaperrors.LDAPUndefinedAttributeType` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

name = 'undefinedAttributeType'

resultCode = 17

exception `ldaptor.protocols.ldap.ldaperrors.LDAPUnknownError` (*resultCode, message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

resultCode = None

exception `ldaptor.protocols.ldap.ldaperrors.LDAPUnwillingToPerform` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

name = 'unwillingToPerform'

resultCode = 53

class `ldaptor.protocols.ldap.ldaperrors.Success` (*msg*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPResult`

name = 'success'

resultCode = 0

`ldaptor.protocols.ldap.ldaperrors.get` (*resultCode, errorMessage*)

Get an instance of the correct exception for this resultCode.

`ldaptor.protocols.ldap.ldaperrors.init` (***errors*)

ldaptor.protocols.ldap.ldapserver module LDAP protocol server

class `ldaptor.protocols.ldap.ldapserver.BaseLDAPServer`

Bases: `twisted.internet.protocol.Protocol`

berdecoder = <LDAPBERDecoderContext_TopLevel identities={0x10: LDAPMessage} fallback=None inherit=<LDAP

checkControls (*controls*)

connectionLost (*reason=<twisted.python.failure.Failure twisted.internet.error.ConnectionDone: Connection was closed cleanly.>*)

Called when TCP connection has been lost

connectionMade ()

TCP connection has opened

dataReceived (*recd*)

debug = False

failDefault (*resultCode, errorMessage*)

handle (*msg*)

handleUnknown (*request, controls, callback*)

queue (*id, op*)

unsolicitedNotification (*msg*)

class `ldaptor.protocols.ldap.ldapserver.LDAPServer`
Bases: `ldaptor.protocols.ldap.ldapserver.BaseLDAPServer`

An LDAP server

boundUser = None

extendedRequest_LDAPPasswordModifyRequest (*data, reply*)

fail_LDAPAddRequest
alias of `LDAPAddResponse`

fail_LDAPBindRequest
alias of `LDAPBindResponse`

fail_LDAPDelRequest
alias of `LDAPDelResponse`

fail_LDAPExtendedRequest
alias of `LDAPExtendedResponse`

fail_LDAPModifyDNRequest
alias of `LDAPModifyDNResponse`

fail_LDAPModifyRequest
alias of `LDAPModifyResponse`

fail_LDAPSearchRequest
alias of `LDAPSearchResultDone`

getRootDSE (*request, reply*)

handle_LDAPAddRequest (*request, controls, reply*)

handle_LDAPBindRequest (*request, controls, reply*)

handle_LDAPDelRequest (*request, controls, reply*)

handle_LDAPExtendedRequest (*request, controls, reply*)

handle_LDAPModifyDNRequest (*request, controls, reply*)

handle_LDAPModifyRequest (*request, controls, reply*)

handle_LDAPSearchRequest (*request, controls, reply*)

handle_LDAPUnbindRequest (*request, controls, reply*)

exception `ldaptor.protocols.ldap.ldapserver.LDAPServerConnectionLostException` (*message=None*)
Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPException`

ldaptor.protocols.ldap.ldapsyntax module Pythonic API for LDAP operations.

exception `ldaptor.protocols.ldap.ldapsyntax.CannotRemoveRDNError` (*key, val=None*)
Bases: `exceptions.Exception`

The attribute to be removed is the RDN for the object and cannot be removed.

exception `ldaptor.protocols.ldap.ldapsyntax.DNNotPresentError`

Bases: `exceptions.Exception`

The requested DN cannot be found by the server.

class `ldaptor.protocols.ldap.ldapsyntax.JournaledLDAPAttributeSet` (*ldapObject*, **a*, ***kw*)

Bases: `ldaptor.attributeset.LDAPAttributeSet`

add (*value*)

clear ()

remove (*value*)

update (*sequence*)

`ldaptor.protocols.ldap.ldapsyntax.LDAPEntry`

alias of `LDAPEntryWithClient`

class `ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithAutoFill` (**args*, ***kwargs*)

Bases: `ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithClient`

addAutofiller (*autoFiller*)

journal (*journalOperation*)

class `ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithClient` (*client*, *dn*, *attributes*={}, *complete*=0)

Bases: `ldaptor.entry.EditableLDAPEntry`

addChild (*rdn*, *attributes*)

bind (*password*)

buildAttributeSet (*key*, *values*)

commit ()

delete ()

fetch (**attributes*)

get (**a*, ***kw*)

has_key (**a*, ***kw*)

items ()

journal (*journalOperation*)

Add a Modification into the list of modifications that need to be flushed to the LDAP server.

Normal callers should not use this, they should use the `o['foo']='bar', 'baz'` -style API that enforces schema, handles errors and updates the cached data.

keys ()

lookup (*dn*)

move (*newDN*)

namingContext ()

search (*filterText*=None, *filterObject*=None, *attributes*=(), *scope*=None, *derefAliases*=None, *sizeLimit*=0, *sizeLimitIsNonFatal*=False, *timeLimit*=0, *typesOnly*=0, *callback*=None)

setPassword (*newPasswd*)

setPasswordMaybe_ExtendedOperation (*newPasswd*)

Set the password on this object.

@param newPasswd: A string containing the new password.

@return: A Deferred that will complete when the operation is done.

setPasswordMaybe_Samba (*newPasswd*)

Set the Samba password on this object if it is a sambaSamAccount or sambaAccount.

@param newPasswd: A string containing the new password.

@return: A Deferred that will complete when the operation is done.

setPassword_ExtendedOperation (*newPasswd*)

Set the password on this object.

@param newPasswd: A string containing the new password.

@return: A Deferred that will complete when the operation is done.

setPassword_Samba (*newPasswd*, *style=None*)

Set the Samba password on this object.

@param newPasswd: A string containing the new password.

@param style: one of 'sambaSamAccount', 'sambaAccount' or None. Specifies the style of samba accounts used. None is default and is the same as 'sambaSamAccount'.

@return: A Deferred that will complete when the operation is done.

undo ()

exception `ldaptor.protocols.ldap.ldapsyntax.MatchNotImplemented` (*op*)

Bases: `exceptions.NotImplementedError`

Match type not implemented

exception `ldaptor.protocols.ldap.ldapsyntax.NoContainingNamingContext`

Bases: `exceptions.Exception`

The server contains to LDAP naming context that would contain this object.

exception `ldaptor.protocols.ldap.ldapsyntax.ObjectDeletedError`

Bases: `ldaptor.protocols.ldap.ldapsyntax.ObjectInBadStateError`

The LDAP object has already been removed, unable to perform operations on it.

exception `ldaptor.protocols.ldap.ldapsyntax.ObjectDirtyError`

Bases: `ldaptor.protocols.ldap.ldapsyntax.ObjectInBadStateError`

The LDAP object has a journal which needs to be committed or undone before this operation.

exception `ldaptor.protocols.ldap.ldapsyntax.ObjectInBadStateError`

Bases: `exceptions.Exception`

The LDAP object in in a bad state.

exception `ldaptor.protocols.ldap.ldapsyntax.PasswordSetAborted`

Bases: `exceptions.Exception`

Aborted

exception `ldaptor.protocols.ldap.ldapsyntax.PasswordSetAggregateError` (*errors*)

Bases: `exceptions.Exception`

Some of the password plugins failed

ldaptor.protocols.ldap.ldif module Support for writing a set of directory entries as LDIF. You probably want to use this only indirectly, as in `str(LDAPEntry(...))`.

TODO support writing modify operations TODO support reading modify operations

TODO implement rest of syntax from RFC2849

`ldaptor.protocols.ldap.ldif.asLDIF` (*dn, attributes*)

`ldaptor.protocols.ldap.ldif.attributeAsLDIF` (*attribute, value*)

`ldaptor.protocols.ldap.ldif.attributeAsLDIF_base64` (*attribute, value*)

`ldaptor.protocols.ldap.ldif.base64_encode` (*s*)

`ldaptor.protocols.ldap.ldif.containsNonprintable` (*s*)

`ldaptor.protocols.ldap.ldif.header` ()

`ldaptor.protocols.ldap.ldif.manyAsLDIF` (*objects*)

ldaptor.protocols.ldap.ldifdelta module

class `ldaptor.protocols.ldap.ldifdelta.LDIFDelta`

Bases: `ldaptor.protocols.ldap.ldifprotocol.LDIF`

MOD_SPEC_TO_DELTA = {'add': <class 'ldaptor.delta.Add'>, 'replace': <class 'ldaptor.delta.Replace'>, 'delete': <class 'ldaptor.delta.Delete'>}

state_IN_ADD_ENTRY (*line*)

state_IN_DELETE (*line*)

state_IN_MOD_SPEC (*line*)

state_WAIT_FOR_CHANGETYPE (*line*)

state_WAIT_FOR_DN (*line*)

state_WAIT_FOR_MOD_SPEC (*line*)

exception `ldaptor.protocols.ldap.ldifdelta.LDIFDeltaAddMissingAttributesError`

Bases: `ldaptor.protocols.ldap.ldifprotocol.LDIFParseError`

Add operation needs to have atleast one attribute type and value.

exception `ldaptor.protocols.ldap.ldifdelta.LDIFDeltaDeleteHasJunkAfterChangeTypeError`

Bases: `ldaptor.protocols.ldap.ldifprotocol.LDIFParseError`

Delete operation takes no attribute types or values.

exception `ldaptor.protocols.ldap.ldifdelta.LDIFDeltaMissingChangeTypeError`

Bases: `ldaptor.protocols.ldap.ldifprotocol.LDIFParseError`

LDIF delta entry has no changetype.

exception `ldaptor.protocols.ldap.ldifdelta.LDIFDeltaModificationDifferentAttributeTypeError`

Bases: `ldaptor.protocols.ldap.ldifprotocol.LDIFParseError`

The attribute type for the change is not the as in the mod-spec header line.

exception `ldaptor.protocols.ldap.ldifdelta.LDIFDeltaModificationMissingEndDashError`

Bases: `ldaptor.protocols.ldap.ldifprotocol.LDIFParseError`

LDIF delta modification has no ending dash.

exception `ldaptor.protocols.ldap.ldifdelta.LDIFDeltaUnknownModificationError`

Bases: `ldaptor.protocols.ldap.ldifprotocol.LDIFParseError`

LDIF delta modification has unknown mod-spec.

`ldaptor.protocols.ldap.ldifdelta.fromLDIFFile(f)`
Read LDIF data from a file.

ldaptor.protocols.ldap.ldifprotocol module

class `ldaptor.protocols.ldap.ldifprotocol.LDIF`
Bases: `object`, `twisted.protocols.basic.LineReceiver`

connectionLost (*reason*=`<twisted.python.failure.Failure twisted.internet.error.ConnectionDone: Connection was closed cleanly.>`)

data = `None`

delimiter = `'\n'`

dn = `None`

gotEntry (*obj*)

lastLine = `None`

lineReceived (*line*)

logicalLineReceived (*line*)

mode = `'HEADER'`

parseValue (*val*)

state_HEADER (*line*)

state_IN_ENTRY (*line*)

state_WAIT_FOR_DN (*line*)

version = `None`

exception `ldaptor.protocols.ldap.ldifprotocol.LDIFEntryStartsWithNonDNError`
Bases: `ldaptor.protocols.ldap.ldifprotocol.LDIFParseError`
LDIF entry starts with a non-DN line

exception `ldaptor.protocols.ldap.ldifprotocol.LDIFEntryStartsWithSpaceError`
Bases: `ldaptor.protocols.ldap.ldifprotocol.LDIFParseError`
Invalid LDIF value format

exception `ldaptor.protocols.ldap.ldifprotocol.LDIFLineWithoutSemicolonError`
Bases: `ldaptor.protocols.ldap.ldifprotocol.LDIFParseError`
LDIF line without semicolon seen

exception `ldaptor.protocols.ldap.ldifprotocol.LDIFParseError`
Bases: `exceptions.Exception`
Error parsing LDIF.

exception `ldaptor.protocols.ldap.ldifprotocol.LDIFTruncatedError`
Bases: `ldaptor.protocols.ldap.ldifprotocol.LDIFParseError`
LDIF appears to be truncated

exception `ldaptor.protocols.ldap.ldifprotocol.LDIFUnsupportedVersionError`
Bases: `ldaptor.protocols.ldap.ldifprotocol.LDIFParseError`
LDIF version not supported

exception `ldaptor.protocols.ldap.ldifprotocol.LDIFVersionNotANumberError`

Bases: `ldaptor.protocols.ldap.ldifprotocol.LDIFParseError`

Non-numeric LDIF version number

ldaptor.protocols.ldap.proxy module LDAP protocol proxy server

class `ldaptor.protocols.ldap.proxy.Proxy` (*config*)

Bases: `ldaptor.protocols.ldap.ldapserver.BaseLDAPServer`

client = None

connectionLost (*reason*)

connectionMade ()

handleUnknown (*request, controls, reply*)

handle_LDAPUnbindRequest (*request, controls, reply*)

protocol

alias of `LDAPClient`

unbound = False

waitingConnect = []

ldaptor.protocols.ldap.svcbindproxy module

class `ldaptor.protocols.ldap.svcbindproxy.ServiceBindingProxy` (*services=None, fallback=None, *a, **kw*)

Bases: `ldaptor.protocols.ldap.proxy.Proxy`

An LDAP proxy that handles non-anonymous bind requests specially.

BindRequests are intercepted and authentication is attempted against each configured service. This authentication is performed against a separate LDAP entry, found by searching for entries with

- objectClass: serviceSecurityObject
- owner: the DN of the original bind attempt
- cn: the service name.

starting at the identity-base as configured in the config file.

Finally, if the authentication does not succeed against any of the configured services, the proxy can fallback to passing the bind request to the real server.

fail_LDAPBindRequest

alias of `LDAPBindResponse`

fallback = False

handle_LDAPBindRequest (*request, controls, reply*)

services = []

timestamp ()

Module contents LDAP protocol logic

Submodules

ldaptor.protocols.pureber module

Pure, simple, BER encoding and decoding

class ldaptor.protocols.pureber.**BERBase** (*tag=None*)

identification ()

tag = None

class ldaptor.protocols.pureber.**BERBoolean** (*value=None, tag=None*)

Bases: *ldaptor.protocols.pureber.BERBase*

classmethod **fromBER** (*klass, tag, content, berdecoder=None*)

tag = 1

class ldaptor.protocols.pureber.**BERDecoderContext** (*fallback=None, inherit=None*)

Identities = {1: <class ldaptor.protocols.pureber.BERBoolean at 0x7f5a9c2461f0>, 2: <class ldaptor.protocols.pureber

inherit ()

lookup_id (*id*)

class ldaptor.protocols.pureber.**BEREnumerated** (*value=None, tag=None*)

Bases: *ldaptor.protocols.pureber.BERInteger*

tag = 10

exception ldaptor.protocols.pureber.**BERException**

Bases: *exceptions.Exception*

exception ldaptor.protocols.pureber.**BERExceptionInsufficientData**

Bases: *exceptions.Exception*

class ldaptor.protocols.pureber.**BERInteger** (*value=None, tag=None*)

Bases: *ldaptor.protocols.pureber.BERBase*

classmethod **fromBER** (*klass, tag, content, berdecoder=None*)

tag = 2

value = None

class ldaptor.protocols.pureber.**BERNull** (*tag=None*)

Bases: *ldaptor.protocols.pureber.BERBase*

classmethod **fromBER** (*klass, tag, content, berdecoder=None*)

tag = 5

class ldaptor.protocols.pureber.**BEROctetString** (*value=None, tag=None*)

Bases: *ldaptor.protocols.pureber.BERBase*

classmethod **fromBER** (*klass, tag, content, berdecoder=None*)

tag = 4

value = None

```

class ldaptor.protocols.pureber.BERSequence (value=None, tag=None)
    Bases: ldaptor.protocols.pureber.BERStructured, UserList.UserList

    classmethod fromBER (klass, tag, content, berdecoder=None)

    tag = 16

class ldaptor.protocols.pureber.BERSequenceOf (value=None, tag=None)
    Bases: ldaptor.protocols.pureber.BERSequence

class ldaptor.protocols.pureber.BERSet (value=None, tag=None)
    Bases: ldaptor.protocols.pureber.BERSequence

    tag = 17

class ldaptor.protocols.pureber.BERStructured (tag=None)
    Bases: ldaptor.protocols.pureber.BERBase

    identification ()

exception ldaptor.protocols.pureber.UnknownBERTag (tag, context)
    Bases: exceptions.Exception

ldaptor.protocols.pureber.ber2int (e, signed=True)

ldaptor.protocols.pureber.berDecodeLength (m, offset=0)
    Return a tuple of (length, lengthLength). m must be atleast one byte long.

ldaptor.protocols.pureber.berDecodeMultiple (content, berdecoder) → [objects]
    Decodes everything in content and returns a list of decoded objects.

    All of content will be decoded, and content must contain complete BER objects.

ldaptor.protocols.pureber.berDecodeObject (context, string) -> (berobject, bytesUsed)
    berobject may be None.

ldaptor.protocols.pureber.int2ber (i, signed=True)

ldaptor.protocols.pureber.int2berlen (i)

ldaptor.protocols.pureber.need (buf, n)

```

ldaptor.protocols.pureldap module

LDAP protocol message conversion; no application logic here.

```

class ldaptor.protocols.pureldap.LDAPAbandonRequest (value=None, id=None, tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPProtocolRequest,
           ldaptor.protocols.pureldap.LDAPInteger

    needs_answer = 0

    tag = 80

class ldaptor.protocols.pureldap.LDAPAddRequest (entry=None, attributes=None, tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPProtocolRequest,
           ldaptor.protocols.pureber.BERSequence

    classmethod fromBER (klass, tag, content, berdecoder=None)

    tag = 72

```

```
class ldaptor.protocols.pureldap.LDAPAddResponse (resultCode=None, matchedDN=None,
                                                errorMessage=None, referral=None,
                                                serverSaslCreds=None, tag=None)
```

```
    Bases: ldaptor.protocols.pureldap.LDAPResult
```

```
    tag = 73
```

```
class ldaptor.protocols.pureldap.LDAPAssertionValue (value=None, tag=None)
```

```
    Bases: ldaptor.protocols.pureber.BEROctetString
```

```
class ldaptor.protocols.pureldap.LDAPAttributeDescription (value=None, tag=None)
```

```
    Bases: ldaptor.protocols.pureber.BEROctetString
```

```
class ldaptor.protocols.pureldap.LDAPAttributeValue (value=None, tag=None)
```

```
    Bases: ldaptor.protocols.pureber.BEROctetString
```

```
class ldaptor.protocols.pureldap.LDAPAttributeValueAssertion (attributeDesc=None,
                                                             assertionValue=None,
                                                             tag=None)
```

```
    Bases: ldaptor.protocols.pureber.BERSequence
```

```
    classmethod fromBER (klass, tag, content, berdecoder=None)
```

```
class ldaptor.protocols.pureldap.LDAPBERDecoderContext (fallback=None, inherit=None)
```

```
    Bases: ldaptor.protocols.pureber.BERDecoderContext
```

```
    Identities = {64: <class 'ldaptor.protocols.pureldap.LDAPBindRequest'>, 65: <class 'ldaptor.protocols.pureldap.LDAPBindResponse'>}
```

```
class ldaptor.protocols.pureldap.LDAPBERDecoderContext_BindResponse (fallback=None,
                                                                      in-
                                                                      herit=None)
```

```
    Bases: ldaptor.protocols.pureber.BERDecoderContext
```

```
    Identities = {135: <class ldaptor.protocols.pureldap.LDAPBindResponse_serverSaslCreds at 0x7f5a9c246600>}
```

```
class ldaptor.protocols.pureldap.LDAPBERDecoderContext_Filter (fallback=None,
                                                                inherit=None)
```

```
    Bases: ldaptor.protocols.pureber.BERDecoderContext
```

```
    Identities = {128: <class 'ldaptor.protocols.pureldap.LDAPFilter_and'>, 129: <class 'ldaptor.protocols.pureldap.LDAPFilter_or'>}
```

```
class ldaptor.protocols.pureldap.LDAPBERDecoderContext_Filter_substrings (fallback=None,
                                                                            in-
                                                                            herit=None)
```

```
    Bases: ldaptor.protocols.pureber.BERDecoderContext
```

```
    Identities = {128: <class ldaptor.protocols.pureldap.LDAPFilter_substrings_initial at 0x7f5a9c246870>, 129: <class ldaptor.protocols.pureldap.LDAPFilter_substrings_substrings at 0x7f5a9c246870>}
```

```
class ldaptor.protocols.pureldap.LDAPBERDecoderContext_LDAPBindRequest (fallback=None,
                                                                           in-
                                                                           herit=None)
```

```
    Bases: ldaptor.protocols.pureber.BERDecoderContext
```

```
    Identities = {128: <class ldaptor.protocols.pureber.BEROctetString at 0x7f5a9c246120>, 131: <class 'ldaptor.protocols.pureldap.LDAPBindRequest_serverSaslCreds'>}
```

```
class ldaptor.protocols.pureldap.LDAPBERDecoderContext_LDAPControls (fallback=None,
                                                                       in-
                                                                       herit=None)
```

```
    Bases: ldaptor.protocols.pureber.BERDecoderContext
```

```
    Identities = {16: <class 'ldaptor.protocols.pureldap.LDAPControl'>}
```

```

class ldaptor.protocols.pureldap.LDAPBERDecoderContext_LDAPExtendedRequest (fallback=None,
                                                                           in-
                                                                           herit=None)
    Bases: ldaptor.protocols.pureber.BERDecoderContext
    Identities = {128: <class ldaptor.protocols.pureber.BEROctetString at 0x7f5a9c246120>, 129: <class ldaptor.protocols.pureber.BERInteger at 0x7f5a9c246120>}

class ldaptor.protocols.pureldap.LDAPBERDecoderContext_LDAPExtendedResponse (fallback=None,
                                                                              in-
                                                                              herit=None)
    Bases: ldaptor.protocols.pureber.BERDecoderContext
    Identities = {138: <class ldaptor.protocols.pureldap.LDAPResponseName at 0x7f5a9c1f3188>, 139: <class ldaptor.protocols.pureldap.LDAPResponseValue at 0x7f5a9c1f3188>}

class ldaptor.protocols.pureldap.LDAPBERDecoderContext_LDAPMessage (fallback=None,
                                                                      in-
                                                                      herit=None)
    Bases: ldaptor.protocols.pureber.BERDecoderContext
    Identities = {128: <class 'ldaptor.protocols.pureldap.LDAPControls'>, 83: <class ldaptor.protocols.pureldap.LDAPSearchFilter at 0x7f5a9c1f3188>}

class ldaptor.protocols.pureldap.LDAPBERDecoderContext_LDAPPasswordModifyRequest (fallback=None,
                                                                                    in-
                                                                                    herit=None)
    Bases: ldaptor.protocols.pureber.BERDecoderContext
    Identities = {128: <class ldaptor.protocols.pureldap.LDAPPasswordModifyRequest_userIdentity at 0x7f5a9c1f32c0>, 129: <class ldaptor.protocols.pureldap.LDAPPasswordModifyRequest_password at 0x7f5a9c1f32c0>}

class ldaptor.protocols.pureldap.LDAPBERDecoderContext_MatchingRuleAssertion (fallback=None,
                                                                                in-
                                                                                herit=None)
    Bases: ldaptor.protocols.pureber.BERDecoderContext
    Identities = {129: <class ldaptor.protocols.pureldap.LDAPMatchingRuleAssertion_matchingRule at 0x7f5a9c246b48>, 130: <class ldaptor.protocols.pureldap.LDAPMatchingRuleAssertion_matchingRuleValue at 0x7f5a9c246b48>}

class ldaptor.protocols.pureldap.LDAPBERDecoderContext_ModifyDNRequest (fallback=None,
                                                                           in-
                                                                           herit=None)
    Bases: ldaptor.protocols.pureber.BERDecoderContext
    Identities = {128: <class ldaptor.protocols.pureldap.LDAPModifyDNResponse_newSuperior at 0x7f5a9c246f58>, 129: <class ldaptor.protocols.pureldap.LDAPModifyDNRequest_modifyDN at 0x7f5a9c246f58>}

class ldaptor.protocols.pureldap.LDAPBERDecoderContext_TopLevel (fallback=None,
                                                                  inherit=None)
    Bases: ldaptor.protocols.pureber.BERDecoderContext
    Identities = {16: <class 'ldaptor.protocols.pureldap.LDAPMessage'>}

class ldaptor.protocols.pureldap.LDAPBindRequest (version=None, dn=None, auth=None,
                                                  tag=None, sasl=False)
    Bases:
        ldaptor.protocols.pureldap.LDAPProtocolRequest,
        ldaptor.protocols.pureber.BERSequence
    classmethod fromBER (klass, tag, content, berdecoder=None)
    tag = 64

class ldaptor.protocols.pureldap.LDAPBindResponse (resultCode=None, matchedDN=None,
                                                  errorMessage=None, referral=None,
                                                  serverSaslCreds=None, tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPResult
    errorMessage = None
    classmethod fromBER (klass, tag, content, berdecoder=None)

```

```
    matchedDN = None
    referral = None
    resultCode = None
    serverSaslCreds = None
    tag = 65
class ldaptor.protocols.pureldap.LDAPBindResponse_serverSaslCreds (value=None,
                                                                    tag=None)
    Bases: ldaptor.protocols.pureber.BEROctetString
    tag = 135
class ldaptor.protocols.pureldap.LDAPControl (controlType,          criticality=None,      con-
                                              trolValue=None, id=None, tag=None)
    Bases: ldaptor.protocols.pureber.BERSequence
    controlValue = None
    criticality = None
    classmethod fromBER (klass, tag, content, berdecoder=None)
class ldaptor.protocols.pureldap.LDAPControls (value=None, tag=None)
    Bases: ldaptor.protocols.pureber.BERSequence
    classmethod fromBER (klass, tag, content, berdecoder=None)
    tag = 128
class ldaptor.protocols.pureldap.LDAPDelRequest (value=None, entry=None, tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPProtocolRequest,
          ldaptor.protocols.pureldap.LDAPString
    tag = 74
class ldaptor.protocols.pureldap.LDAPDelResponse (resultCode=None,  matchedDN=None,
                                                  errorMessage=None,  referral=None,
                                                  serverSaslCreds=None, tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPResult
    tag = 75
class ldaptor.protocols.pureldap.LDAPExtendedRequest (requestName=None,  request-
                                                       Value=None, tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPProtocolRequest,
          ldaptor.protocols.pureber.BERSequence
    classmethod fromBER (klass, tag, content, berdecoder=None)
    requestName = None
    requestValue = None
    tag = 87
class ldaptor.protocols.pureldap.LDAPExtendedResponse (resultCode=None,
                                                       matchedDN=None,  errorMes-
                                                       sage=None,  referral=None,
                                                       serverSaslCreds=None,  respon-
                                                       seName=None,  response=None,
                                                       tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPResult
```

```

classmethod fromBER (klass, tag, content, berdecoder=None)

response = None

responseName = None

tag = 88

class ldaptor.protocols.pureldap.LDAPFilter (tag=None)
  Bases: ldaptor.protocols.pureber.BERStructured

class ldaptor.protocols.pureldap.LDAPFilterSet (value=None, tag=None)
  Bases: ldaptor.protocols.pureber.BERSet

  classmethod fromBER (klass, tag, content, berdecoder=None)

class ldaptor.protocols.pureldap.LDAPFilter_and (value=None, tag=None)
  Bases: ldaptor.protocols.pureldap.LDAPFilterSet

  asText ()

  tag = 128

class ldaptor.protocols.pureldap.LDAPFilter_approxMatch (attributeDesc=None, assertionValue=None, tag=None)
  Bases: ldaptor.protocols.pureldap.LDAPAttributeValueAssertion

  asText ()

  tag = 136

class ldaptor.protocols.pureldap.LDAPFilter_equalityMatch (attributeDesc=None, assertionValue=None, tag=None)
  Bases: ldaptor.protocols.pureldap.LDAPAttributeValueAssertion

  asText ()

  tag = 131

class ldaptor.protocols.pureldap.LDAPFilter_extensibleMatch (matchingRule=None, type=None, matchValue=None, dnAttributes=None, tag=None)
  Bases: ldaptor.protocols.pureldap.LDAPMatchingRuleAssertion

  tag = 137

class ldaptor.protocols.pureldap.LDAPFilter_greaterOrEqual (attributeDesc=None, assertionValue=None, tag=None)
  Bases: ldaptor.protocols.pureldap.LDAPAttributeValueAssertion

  asText ()

  tag = 133

class ldaptor.protocols.pureldap.LDAPFilter_lessOrEqual (attributeDesc=None, assertionValue=None, tag=None)
  Bases: ldaptor.protocols.pureldap.LDAPAttributeValueAssertion

  asText ()

  tag = 134

```

```
class ldaptor.protocols.pureldap.LDAPFilter_not (value, tag=130)
    Bases: ldaptor.protocols.pureldap.LDAPFilter

    asText ()

    classmethod fromBER (klass, tag, content, berdecoder=None)

    tag = 130

class ldaptor.protocols.pureldap.LDAPFilter_or (value=None, tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPFilterSet

    asText ()

    tag = 129

class ldaptor.protocols.pureldap.LDAPFilter_present (value=None, tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPAttributeDescription

    asText ()

    tag = 135

class ldaptor.protocols.pureldap.LDAPFilter_substrings (type=None, substrings=None,
                                                         tag=None)
    Bases: ldaptor.protocols.pureber.BERSequence

    asText ()

    classmethod fromBER (klass, tag, content, berdecoder=None)

    tag = 132

class ldaptor.protocols.pureldap.LDAPFilter_substrings_any (value=None, tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPString

    asText ()

    tag = 129

class ldaptor.protocols.pureldap.LDAPFilter_substrings_final (value=None,
                                                                tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPString

    asText ()

    tag = 130

class ldaptor.protocols.pureldap.LDAPFilter_substrings_initial (value=None,
                                                                tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPString

    asText ()

    tag = 128

class ldaptor.protocols.pureldap.LDAPInteger (value=None, tag=None)
    Bases: ldaptor.protocols.pureber.BERInteger

class ldaptor.protocols.pureldap.LDAPMatchingRuleAssertion (matchingRule=None,
                                                             type=None, match-
                                                             Value=None, dnAt-
                                                             tributes=None,
                                                             tag=None)

    Bases: ldaptor.protocols.pureber.BERSequence

    dnAttributes = None
```

```

classmethod fromBER (klass, tag, content, berdecoder=None)
matchValue = None
matchingRule = None
type = None
class ldaptor.protocols.pureldap.LDAPMatchingRuleAssertion_dnAttributes (value=None, tag=None)
    Bases: ldaptor.protocols.pureber.BERBoolean
tag = 132
class ldaptor.protocols.pureldap.LDAPMatchingRuleAssertion_matchValue (value=None, tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPAssertionValue
tag = 131
class ldaptor.protocols.pureldap.LDAPMatchingRuleAssertion_matchingRule (value=None, tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPMatchingRuleId
tag = 129
class ldaptor.protocols.pureldap.LDAPMatchingRuleAssertion_type (value=None, tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPAttributeDescription
tag = 130
class ldaptor.protocols.pureldap.LDAPMatchingRuleId (value=None, tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPString
class ldaptor.protocols.pureldap.LDAPMessage (value=None, controls=None, id=None, tag=None)
    Bases: ldaptor.protocols.pureber.BERSequence
classmethod fromBER (klass, tag, content, berdecoder=None)
id = None
value = None
class ldaptor.protocols.pureldap.LDAPModifyDNRequest (entry, newrdn, deleteoldrdn, newSuperior=None, tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPProtocolRequest, ldaptor.protocols.pureber.BERSequence
deleteoldrdn = None
entry = None
classmethod fromBER (klass, tag, content, berdecoder=None)
newSuperior = None
newrdn = None
tag = 76
class ldaptor.protocols.pureldap.LDAPModifyDNResponse (resultCode=None, matchedDN=None, errorMessage=None, referral=None, serverSaslCreds=None, tag=None)

```

Bases: *ldaptor.protocols.pureldap.LDAPResult*

tag = 77

class *ldaptor.protocols.pureldap.LDAPModifyDNResponse_newSuperior* (*value=None, tag=None*)

Bases: *ldaptor.protocols.pureldap.LDAPString*

tag = 128

class *ldaptor.protocols.pureldap.LDAPModifyRequest* (*object=None, modification=None, tag=None*)

Bases: *ldaptor.protocols.pureldap.LDAPProtocolRequest, ldaptor.protocols.pureber.BERSequence*

classmethod *fromBER* (*klass, tag, content, berdecoder=None*)

modification = None

object = None

tag = 70

class *ldaptor.protocols.pureldap.LDAPModifyResponse* (*resultCode=None, matchedDN=None, errorMessage=None, referral=None, serverSaslCreds=None, tag=None*)

Bases: *ldaptor.protocols.pureldap.LDAPResult*

tag = 71

class *ldaptor.protocols.pureldap.LDAPOID* (*value=None, tag=None*)

Bases: *ldaptor.protocols.pureber.BEROctetString*

class *ldaptor.protocols.pureldap.LDAPPasswordModifyRequest* (*requestName=None, userIdentity=None, oldPasswd=None, newPasswd=None, tag=None*)

Bases: *ldaptor.protocols.pureldap.LDAPExtendedRequest*

oid = '1.3.6.1.4.1.4203.1.11.1'

class *ldaptor.protocols.pureldap.LDAPPasswordModifyRequest_newPasswd* (*value=None, tag=None*)

Bases: *ldaptor.protocols.pureber.BEROctetString*

tag = 130

class *ldaptor.protocols.pureldap.LDAPPasswordModifyRequest_oldPasswd* (*value=None, tag=None*)

Bases: *ldaptor.protocols.pureber.BEROctetString*

tag = 129

class *ldaptor.protocols.pureldap.LDAPPasswordModifyRequest_userIdentity* (*value=None, tag=None*)

Bases: *ldaptor.protocols.pureber.BEROctetString*

tag = 128

class *ldaptor.protocols.pureldap.LDAPProtocolOp*

class *ldaptor.protocols.pureldap.LDAPProtocolRequest*
Bases: *ldaptor.protocols.pureldap.LDAPProtocolOp*

```

needs_answer = 1

class ldaptor.protocols.pureldap.LDAPProtocolResponse
    Bases: ldaptor.protocols.pureldap.LDAPProtocolOp

class ldaptor.protocols.pureldap.LDAPReferral (value=None, tag=None)
    Bases: ldaptor.protocols.pureldap.BERSequence

tag = 131

class ldaptor.protocols.pureldap.LDAPResponse (value=None, tag=None)
    Bases: ldaptor.protocols.pureldap.BEROctetString

tag = 139

class ldaptor.protocols.pureldap.LDAPResponseName (value=None, tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPOID

tag = 138

class ldaptor.protocols.pureldap.LDAPResult (resultCode=None, matchedDN=None, error
message=None, referral=None, serverSasl
Creds=None, tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPProtocolResponse,
ldaptor.protocols.pureldap.BERSequence
    classmethod fromBER (klass, tag, content, berdecoder=None)

class ldaptor.protocols.pureldap.LDAPSearchRequest (baseObject=None, scope=None,
derefAliases=None, sizeLimit=None,
timeLimit=None, typesOnly=None,
filter=None, attributes=None,
tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPProtocolRequest,
ldaptor.protocols.pureldap.BERSequence

attributes = []
baseObject = ''
derefAliases = 0
filter = LDAPFilter_present(value='objectClass')
    classmethod fromBER (klass, tag, content, berdecoder=None)
scope = 2
sizeLimit = 0
tag = 67
timeLimit = 0
typesOnly = 0

class ldaptor.protocols.pureldap.LDAPSearchResultDone (resultCode=None,
matchedDN=None, errorMes
sage=None, referral=None,
serverSaslCreds=None,
tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPResult
tag = 69

```

```
class ldaptor.protocols.pureldap.LDAPSearchResultEntry (objectName,          attributes,
                                                    tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPProtocolResponse,
           ldaptor.protocols.pureber.BERSequence
    classmethod fromBER (klass, tag, content, berdecoder=None)
    tag = 68

class ldaptor.protocols.pureldap.LDAPSearchResultReference
    Bases: ldaptor.protocols.pureldap.LDAPProtocolResponse
    classmethod fromBER (tag, content, berdecoder=None)
    tag = 83

class ldaptor.protocols.pureldap.LDAPStartTLSRequest (requestName=None, tag=None)
    Bases: ldaptor.protocols.pureldap.LDAPExtendedRequest
    Request to start Transport Layer Security.
    See RFC 2830 for details.
    oid = '1.3.6.1.4.1.1466.20037'

class ldaptor.protocols.pureldap.LDAPString (value=None, tag=None)
    Bases: ldaptor.protocols.pureber.BEROctetString

class ldaptor.protocols.pureldap.LDAPUnbindRequest (*args, **kwargs)
    Bases: ldaptor.protocols.pureldap.LDAPProtocolRequest,
           ldaptor.protocols.pureber.BERNull
    needs_answer = 0
    tag = 66

ldaptor.protocols.pureldap.alloc_ldap_message_id()
ldaptor.protocols.pureldap.escape (s)
```

Module contents

ldaptor.samba package

Submodules

ldaptor.samba.smbpassword module

ldaptor.samba.smbpassword.lmhash (password='')

Generates lanman password hash for a given password.

Note that the author thinks LanMan hashes should be banished from the face of the earth.

ldaptor.samba.smbpassword.lmhash_locked (password='')

Generates a lanman password hash that matches no password.

Note that the author thinks LanMan hashes should be banished from the face of the earth.

ldaptor.samba.smbpassword.nthash (password='')

Generates nt md4 password hash for a given password.

Module contents

3.3.2 Submodules

3.3.3 `ldaptor.attributeset` module

class `ldaptor.attributeset.LDAPAttributeSet` (*key, *a, **kw*)

Bases: `set`

copy ()

difference (*other*)

intersection (*other*)

symmetric_difference (*other*)

union (*other*)

3.3.4 `ldaptor.checkers` module

class `ldaptor.checkers.LDAPBindingChecker` (*cfg*)

The `avatarID` returned is an `LDAPEntry`.

credentialInterfaces = (`<InterfaceClass twisted.cred.credentials.IUsernamePassword>`),

requestAvatarId (*credentials*)

`ldaptor.checkers.makeFilter` (*name, template=None*)

3.3.5 `ldaptor.compat` module

`ldaptor.compat.print_` (**args, **kwds*)

The new-style print function.

`ldaptor.compat.callable` (*object*) → `bool`

Return whether the object is callable (i.e., some kind of function). Note that classes are callable, as are instances with a `__call__()` method.

`ldaptor.compat.base_string_types`

alias of `basestring`

`ldaptor.compat.u` (*s*)

`ldaptor.compat.b` (*s*)

class `ldaptor.compat.unicode` (*object=''*) → `unicode object`

Bases: `basestring`

`unicode(string[, encoding[, errors]])` → `unicode object`

Create a new Unicode object from the given encoded string. `encoding` defaults to the current default string encoding. `errors` can be 'strict', 'replace' or 'ignore' and defaults to 'strict'.

capitalize () → `unicode`

Return a capitalized version of `S`, i.e. make the first character have upper case and the rest lower case.

center (*width* [, *fillchar*]) → `unicode`

Return `S` centered in a Unicode string of length `width`. Padding is done using the specified fill character (default is a space)

- count** (*sub*[, *start*[, *end*]]) → int
Return the number of non-overlapping occurrences of substring *sub* in Unicode string *S*[*start*:*end*]. Optional arguments *start* and *end* are interpreted as in slice notation.
- decode** ([*encoding*[, *errors*]]) → string or unicode
Decodes *S* using the codec registered for encoding. *encoding* defaults to the default encoding. *errors* may be given to set a different error handling scheme. Default is 'strict' meaning that encoding errors raise a `UnicodeDecodeError`. Other possible values are 'ignore' and 'replace' as well as any other name registered with `codecs.register_error` that is able to handle `UnicodeDecodeErrors`.
- encode** ([*encoding*[, *errors*]]) → string or unicode
Encodes *S* using the codec registered for encoding. *encoding* defaults to the default encoding. *errors* may be given to set a different error handling scheme. Default is 'strict' meaning that encoding errors raise a `UnicodeEncodeError`. Other possible values are 'ignore', 'replace' and 'xmlcharrefreplace' as well as any other name registered with `codecs.register_error` that can handle `UnicodeEncodeErrors`.
- endswith** (*suffix*[, *start*[, *end*]]) → bool
Return True if *S* ends with the specified *suffix*, False otherwise. With optional *start*, test *S* beginning at that position. With optional *end*, stop comparing *S* at that position. *suffix* can also be a tuple of strings to try.
- expandtabs** ([*tabsize*]) → unicode
Return a copy of *S* where all tab characters are expanded using spaces. If *tabsize* is not given, a tab size of 8 characters is assumed.
- find** (*sub*[, *start*[, *end*]]) → int
Return the lowest index in *S* where substring *sub* is found, such that *sub* is contained within *S*[*start*:*end*]. Optional arguments *start* and *end* are interpreted as in slice notation.
Return -1 on failure.
- format** (**args*, ***kwargs*) → unicode
Return a formatted version of *S*, using substitutions from *args* and *kwargs*. The substitutions are identified by braces ('{' and '}').
- index** (*sub*[, *start*[, *end*]]) → int
Like *S*.find() but raise `ValueError` when the substring is not found.
- isalnum** () → bool
Return True if all characters in *S* are alphanumeric and there is at least one character in *S*, False otherwise.
- isalpha** () → bool
Return True if all characters in *S* are alphabetic and there is at least one character in *S*, False otherwise.
- isdecimal** () → bool
Return True if there are only decimal characters in *S*, False otherwise.
- isdigit** () → bool
Return True if all characters in *S* are digits and there is at least one character in *S*, False otherwise.
- islower** () → bool
Return True if all cased characters in *S* are lowercase and there is at least one cased character in *S*, False otherwise.
- isnumeric** () → bool
Return True if there are only numeric characters in *S*, False otherwise.
- isspace** () → bool
Return True if all characters in *S* are whitespace and there is at least one character in *S*, False otherwise.
- istitle** () → bool
Return True if *S* is a titlecased string and there is at least one character in *S*, i.e. upper- and titlecase

characters may only follow uncased characters and lowercase characters only cased ones. Return False otherwise.

isupper () → bool

Return True if all cased characters in S are uppercase and there is at least one cased character in S, False otherwise.

join (*iterable*) → unicode

Return a string which is the concatenation of the strings in the iterable. The separator between elements is S.

ljust (*width* [, *fillchar*]) → int

Return S left-justified in a Unicode string of length width. Padding is done using the specified fill character (default is a space).

lower () → unicode

Return a copy of the string S converted to lowercase.

lstrip ([*chars*]) → unicode

Return a copy of the string S with leading whitespace removed. If chars is given and not None, remove characters in chars instead. If chars is a str, it will be converted to unicode before stripping

partition (*sep*) -> (*head*, *sep*, *tail*)

Search for the separator sep in S, and return the part before it, the separator itself, and the part after it. If the separator is not found, return S and two empty strings.

replace (*old*, *new* [, *count*]) → unicode

Return a copy of S with all occurrences of substring old replaced by new. If the optional argument count is given, only the first count occurrences are replaced.

rfind (*sub* [, *start* [, *end*]]) → int

Return the highest index in S where substring sub is found, such that sub is contained within S[start:end]. Optional arguments start and end are interpreted as in slice notation.

Return -1 on failure.

rindex (*sub* [, *start* [, *end*]]) → int

Like S.rfind() but raise ValueError when the substring is not found.

rjust (*width* [, *fillchar*]) → unicode

Return S right-justified in a Unicode string of length width. Padding is done using the specified fill character (default is a space).

rpartition (*sep*) -> (*head*, *sep*, *tail*)

Search for the separator sep in S, starting at the end of S, and return the part before it, the separator itself, and the part after it. If the separator is not found, return two empty strings and S.

rsplit ([*sep* [, *maxsplit*]]) → list of strings

Return a list of the words in S, using sep as the delimiter string, starting at the end of the string and working to the front. If maxsplit is given, at most maxsplit splits are done. If sep is not specified, any whitespace string is a separator.

rstrip ([*chars*]) → unicode

Return a copy of the string S with trailing whitespace removed. If chars is given and not None, remove characters in chars instead. If chars is a str, it will be converted to unicode before stripping

split ([*sep* [, *maxsplit*]]) → list of strings

Return a list of the words in S, using sep as the delimiter string. If maxsplit is given, at most maxsplit splits are done. If sep is not specified or is None, any whitespace string is a separator and empty strings are removed from the result.

splitlines (*keepends=False*) → list of strings

Return a list of the lines in S, breaking at line boundaries. Line breaks are not included in the resulting list unless keepends is given and true.

startswith (*prefix* [, *start* [, *end*]]) → bool

Return True if S starts with the specified prefix, False otherwise. With optional start, test S beginning at that position. With optional end, stop comparing S at that position. prefix can also be a tuple of strings to try.

strip ([*chars*]) → unicode

Return a copy of the string S with leading and trailing whitespace removed. If chars is given and not None, remove characters in chars instead. If chars is a str, it will be converted to unicode before stripping

swapcase () → unicode

Return a copy of S with uppercase characters converted to lowercase and vice versa.

title () → unicode

Return a titlecased version of S, i.e. words start with title case characters, all remaining cased characters have lower case.

translate (*table*) → unicode

Return a copy of the string S, where all characters have been mapped through the given translation table, which must be a mapping of Unicode ordinals to Unicode ordinals, Unicode strings or None. Unmapped characters are left untouched. Characters mapped to None are deleted.

upper () → unicode

Return a copy of S converted to uppercase.

zfill (*width*) → unicode

Pad a numeric string S with zeros on the left, to fill a field of the specified width. The string S is never truncated.

`ldaptor.compat.bytes`

alias of `str`

`ldaptor.compat.uascii_to_str` (*s*)

helper to convert ascii unicode -> native str

`ldaptor.compat.bascii_to_str` (*s*)

helper to convert ascii bytes -> native str

`ldaptor.compat.str_to_uascii` (*s*)

helper to convert ascii native str -> unicode

`ldaptor.compat.str_to_bascii` (*s*)

helper to convert ascii native str -> bytes

`ldaptor.compat.join_unicode` ()

`S.join(iterable)` -> unicode

Return a string which is the concatenation of the strings in the iterable. The separator between elements is S.

`ldaptor.compat.join_bytes` ()

`S.join(iterable)` -> string

Return a string which is the concatenation of the strings in the iterable. The separator between elements is S.

`ldaptor.compat.join_byte_values` (*values*)

`ldaptor.compat.join_byte_elems` ()

`S.join(iterable)` -> string

Return a string which is the concatenation of the strings in the iterable. The separator between elements is S.

`ldaptor.compat.byte_elem_value()`
`ord(c) -> integer`

Return the integer ordinal of a one-character string.

`ldaptor.compat.iter_byte_values(s)`
`iterate over byte string as sequence of ints 0-255`

`ldaptor.compat.irange`
`alias of xrange`

class `ldaptor.compat.imap`
`Bases: object`

`imap(func, *iterables) -> imap object`

Make an iterator that computes the function using arguments from each of the iterables. Like `map()` except that it returns an iterator instead of a list and that it stops when the shortest iterable is exhausted instead of filling in `None` for shorter iterables.

next

`ldaptor.compat.lmap()`
`map(function, sequence[, sequence, ...]) -> list`

Return a list of the results of applying the function to the items of the argument sequence(s). If more than one sequence is given, the function is called with an argument list consisting of the corresponding item of each sequence, substituting `None` for missing values when not all sequences have the same length. If the function is `None`, return a list of the items of the sequence (or a list of tuples if more than one sequence).

`ldaptor.compat.iteritems(d)`

`ldaptor.compat.itervalues(d)`

`ldaptor.compat.next(iterator[, default])`

Return the next item from the iterator. If `default` is given and the iterator is exhausted, it is returned instead of raising `StopIteration`.

`ldaptor.compat.exc_err()`
`return current error object (to avoid try/except syntax change)`

`ldaptor.compat.get_method_function(func)`
`given (potential) method, return underlying function`

`ldaptor.compat.add_doc(obj, doc)`
`add docstring to an object`

3.3.6 Ldaptor.config module

class `ldaptor.config.LDAPConfig` (*baseDN=None, serviceLocationOverrides=None, identity-BaseDN=None, identitySearch=None*)

`Bases: object`

baseDN = None

copy (***kw*)

getBaseDN ()

getIdentityBaseDN ()

getIdentitySearch (*name*)

getServiceLocationOverrides ()

identityBaseDN = None

identitySearch = None

exception `ldaptor.config.MissingBaseDNError`

Bases: `exceptions.Exception`

Configuration must specify a base DN

`ldaptor.config.loadConfig` (*configFiles=None, reload=False*)

Load configuration file.

`ldaptor.config.useLMhash` ()

Read configuration file if necessary and return whether to use LanMan hashes or not.

3.3.7 `ldaptor.delta` module

Changes to the content of one single LDAP entry.

(This means these do not belong here: adding or deleting of entries, changing of location in tree)

class `ldaptor.delta.Add` (*key, *a, **kw*)

Bases: `ldaptor.delta.Modification`

asLDIF ()

patch (*entry*)

class `ldaptor.delta.AddOp` (*entry*)

Bases: `ldaptor.delta.Operation`

asLDIF ()

patch (*root*)

class `ldaptor.delta.Delete` (*key, *a, **kw*)

Bases: `ldaptor.delta.Modification`

asLDIF ()

patch (*entry*)

class `ldaptor.delta.DeleteOp` (*dn*)

Bases: `ldaptor.delta.Operation`

asLDIF ()

patch (*root*)

class `ldaptor.delta.Modification` (*key, *a, **kw*)

Bases: `ldaptor.attributeset.LDAPAttributeSet`

asLDAP ()

patch (*entry*)

class `ldaptor.delta.ModifyOp` (*dn, modifications=[]*)

Bases: `ldaptor.delta.Operation`

asLDAP ()

asLDIF ()

classmethod **fromLDAP** (*class_, request*)

patch (*root*)

```
class ldaptor.delta.Operation
```

```
Bases: object
```

```
patch (root)
```

```
Find the correct entry in IConnectedLDAPEntry and patch it.
```

```
@param root: IConnectedLDAPEntry that is at the root of the subtree the patch applies to.
```

```
@returns: Deferred with None or failure.
```

```
class ldaptor.delta.Replace (key, *a, **kw)
```

```
Bases: ldaptor.delta.Modification
```

```
asLDIF ()
```

```
patch (entry)
```

3.3.8 ldaptor.dns module

DNS-related utilities.

```
ldaptor.dns.atn (ip)
```

```
ldaptor.dns.atn_numbits (num)
```

```
ldaptor.dns.atn_octets (ip)
```

```
ldaptor.dns.netmaskToNumbits (netmask)
```

```
ldaptor.dns.ntoa (n)
```

```
ldaptor.dns.ptrSoaName (ip, netmask)
```

```
Convert an IP address and netmask to a CIDR delegation -style zone name.
```

3.3.9 ldaptor.entry module

```
class ldaptor.entry.BaseLDAPEntry (dn, attributes={})
```

```
Bases: object
```

```
bind (password)
```

```
buildAttributeSet (key, values)
```

```
diff (other)
```

```
Compute differences between this and another LDAP entry.
```

```
@param other: An LDAPEntry to compare to.
```

```
@return: None if equal, otherwise a ModifyOp that would make this entry look like other.
```

```
dn = None
```

```
get (key, default=None)
```

```
hasMember (dn)
```

```
has_key (key)
```

```
items ()
```

```
keys ()
```

```
class ldaptor.entry.EditableLDAPEntry (dn, attributes={})
```

```
Bases: ldaptor.entry.BaseLDAPEntry
```

```
commit ()  
delete ()  
move (newDN)  
setPassword (newPasswd, salt=None)  
undo ()
```

```
ldaptor.entry.sshaDigest (passphrase, salt=None)
```

3.3.10 ldaptor.entryhelpers module

```
class ldaptor.entryhelpers.DiffTreeMixin
```

```
Bases: object
```

```
diffTree (other, result=None)
```

```
class ldaptor.entryhelpers.MatchMixin
```

```
Bases: object
```

```
match (filter)
```

```
class ldaptor.entryhelpers.SearchByTreeWalkingMixin
```

```
Bases: object
```

```
search (filterText=None, filterObject=None, attributes=(), scope=None, derefAliases=None, sizeLimit=0, timeLimit=0, typesOnly=0, callback=None)
```

```
class ldaptor.entryhelpers.SubtreeFromChildrenMixin
```

```
Bases: object
```

```
subtree (callback=None)
```

3.3.11 ldaptor.generate_password module

```
exception ldaptor.generate_password.PwgenException
```

```
Bases: exceptions.Exception
```

```
class ldaptor.generate_password.ReadPassword (deferred, count=1)
```

```
Bases: twisted.internet.protocol.ProcessProtocol
```

```
errReceived (data)
```

```
outReceived (data)
```

```
processEnded (reason)
```

```
ldaptor.generate_password.generate (reactor, n=1)
```

3.3.12 ldaptor.inmemory module

```
class ldaptor.inmemory.InMemoryLDIFProtocol
```

```
Bases: ldaptor.protocols.ldap.ldifprotocol.LDIF
```

Receive LDIF data and gather results into an ReadOnlyInMemoryLDAPEntry.

You can override `lookupFailed` and `addFailed` to provide smarter error handling. They are called as Deferred `errbacks`; returning the reason causes error to pass onward and abort the whole operation. Returning `None` from `lookupFailed` skips that entry, but continues loading.

When the full LDIF data has been read, the completed Deferred will trigger.

addFailed (*reason*, *entry*)

connectionLost (*reason*)

gotEntry (*entry*)

lookupFailed (*reason*, *entry*)

exception `ldaptor.inmemory.LDAPCannotRemoveRootError` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPNamingViolation`

Cannot remove root of LDAP tree

class `ldaptor.inmemory.ReadOnlyInMemoryLDAPEntry` (**a*, ***kw*)

Bases: `ldaptor.entry.EditableLDAPEntry`, `ldaptor.entryhelpers.DiffTreeMixin`,
`ldaptor.entryhelpers.SubtreeFromChildrenMixin`, `ldaptor.entryhelpers.MatchMixin`,
`ldaptor.entryhelpers.SearchByTreeWalkingMixin`

addChild (*rdn*, *attributes*)

TODO ugly API. Returns the created entry.

children (*callback=None*)

commit ()

delete ()

deleteChild (*rdn*)

fetch (**attributes*)

lookup (*dn*)

move (*newDN*)

parent ()

`ldaptor.inmemory.fromLDIFFile` (*f*)

Read LDIF data from a file.

3.3.13 `ldaptor.insensitive` module

class `ldaptor.insensitive.InsensitiveString`

Bases: `str`

A `str` subclass that performs all matching without regard to case.

3.3.14 `ldaptor.interfaces` module

3.3.15 `ldaptor.ldapfilter` module

exception `ldaptor.ldapfilter.InvalidLDAPFilter` (*msg*, *loc*, *text*)

Bases: `exceptions.Exception`

`ldaptor.ldapfilter.parseExtensible` (*attr*, *s*)

`ldaptor.ldapfilter.parseFilter` (*s*)

`ldaptor.ldapfilter.parseMaybeSubstring` (*attrType*, *s*)

3.3.16 Ldaptor.Ldiftree module

Manage LDAP data as a tree of LDIF files.

exception `ldaptor.ldiftree.LDAPCannotRemoveRootError` (*message=None*)

Bases: `ldaptor.protocols.ldap.ldaperrors.LDAPNamingViolation`

Cannot remove root of LDAP tree

class `ldaptor.ldiftree.LDIFTreeEntry` (*path, dn=None, *a, **kw*)

Bases: `ldaptor.entry.EditableLDAPEntry`, `ldaptor.entryhelpers.DiffTreeMixin`,
`ldaptor.entryhelpers.SubtreeFromChildrenMixin`, `ldaptor.entryhelpers.MatchMixin`,
`ldaptor.entryhelpers.SearchByTreeWalkingMixin`

addChild (*rdn, attributes*)

children (*callback=None*)

commit ()

delete ()

deleteChild (*rdn*)

lookup (*dn*)

move (*newDN*)

parent ()

exception `ldaptor.ldiftree.LDIFTreeEntryContainsMultipleEntries`

Bases: `exceptions.Exception`

LDIFTree entry contains multiple LDIF entries.

exception `ldaptor.ldiftree.LDIFTreeEntryContainsNoEntries`

Bases: `exceptions.Exception`

LDIFTree entry does not contain a valid LDIF entry.

exception `ldaptor.ldiftree.LDIFTreeNoSuchObject`

Bases: `exceptions.Exception`

LDIFTree does not contain such entry.

class `ldaptor.ldiftree.StoreParsedLDIF`

Bases: `ldaptor.protocols.ldap.ldifprotocol.LDIF`

connectionLost (*reason*)

gotEntry (*obj*)

`ldaptor.ldiftree.get` (*path, dn*)

`ldaptor.ldiftree.put` (*path, entry*)

3.3.17 Ldaptor.md4 module

helper implementing insecure and obsolete md4 algorithm. used for NTHASH format, which is also insecure and broken, since it's just md4(password)

implemented based on rfc at <http://www.faqs.org/rfcs/rfc1320.html>

`ldaptor.md4.md4` (*content=None*)

wrapper for `hashlib.new('md4')`

3.3.18 Ldaptor.numberalloc module

Find an available uidNumber/gidNumber/other similar number.

class `Ldaptor.numberalloc.freeNumberGuesser` (*makeAGuess*, *min=None*, *max=None*)

startGuessing ()

`Ldaptor.numberalloc.getFreeNumber` (*ldapObject*, *numberType*, *min=None*, *max=None*)

class `Ldaptor.numberalloc.ldapGuesser` (*ldapObject*, *numberType*)

guess (*num*)

3.3.19 Ldaptor.schema module

class `Ldaptor.schema.ASN1ParserThingie`

class `Ldaptor.schema.AttributeTypeDescription` (*text*)

Bases: `Ldaptor.schema.ASN1ParserThingie`

ASN Syntax:

```
AttributeTypeDescription = "(" whsp
    numericoid whsp           ; AttributeType identifier
    [ "NAME" qdescrs ]       ; name used in AttributeType
    [ "DESC" qdstring ]      ; description
    [ "OBSOLETE" whsp ]
    [ "SUP" woid ]           ; derived from this other AttributeType
    [ "EQUALITY" woid ]      ; Matching Rule name
    [ "ORDERING" woid ]      ; Matching Rule name
    [ "SUBSTR" woid ]        ; Matching Rule name
    [ "SYNTAX" whsp noidlen whsp ] ; see section 4.3
    [ "SINGLE-VALUE" whsp ]   ; default multi-valued
    [ "COLLECTIVE" whsp ]    ; default not collective
    [ "NO-USER-MODIFICATION" whsp ]; default user modifiable
    [ "USAGE" whsp AttributeUsage ]; default userApplications
    whsp ")"

AttributeUsage =
    "userApplications" /
    "directoryOperation" /
    "distributedOperation" / ; DSA-shared
    "dSAOperation" ; DSA-specific, value depends on server

noidlen = numericoid [ "{" len "}" ]

len      = numericstring
```

class `Ldaptor.schema.MatchingRuleDescription` (*text*)

Bases: `Ldaptor.schema.ASN1ParserThingie`

ASN Syntax:

```
MatchingRuleDescription = "(" whsp
    numericoid whsp ; MatchingRule identifier
    [ "NAME" qdescrs ]
    [ "DESC" qdstring ]
```

```
[ "OBSOLETE" whsp ]
"SYNTAX" numericoid
whsp ")"
```

class `ldaptor.schema.ObjectClassDescription` (*text*)

Bases: `ldaptor.schema.ASN1ParserThingie`

ASN Syntax:

```
d                = "0" / "1" / "2" / "3" / "4" /
                  "5" / "6" / "7" / "8" / "9"

numericstring    = 1*d

numericoid       = numericstring *( "." numericstring )

space            = 1*" "

whsp             = [ space ]

descr            = keystring

qdescr           = whsp "'" descr "'" whsp

qdescrlist       = [ qdescr *( qdescr ) ]

; object descriptors used as schema element names
qdescrs          = qdescr / ( whsp "(" qdescrlist ")" whsp )

dstring          = 1*utf8

qdstring         = whsp "'" dstring "'" whsp

descr            = keystring

oid              = descr / numericoid

woid             = whsp oid whsp

; set of oids of either form
oids             = woid / ( "(" oidlist ")" )

ObjectClassDescription = "(" whsp
                        numericoid whsp      ; ObjectClass identifier
                        [ "NAME" qdescrs ]
                        [ "DESC" qdstring ]
                        [ "OBSOLETE" whsp ]
                        [ "SUP" oids ]        ; Superior ObjectClasses
                        [ ( "ABSTRACT" / "STRUCTURAL" / "AUXILIARY" ) whsp ]
                                                ; default structural
                        [ "MUST" oids ]       ; AttributeTypes
                        [ "MAY" oids ]       ; AttributeTypes
                        whsp ")"
```

class `ldaptor.schema.SyntaxDescription` (*text*)

Bases: `ldaptor.schema.ASN1ParserThingie`

ASN Syntax:

```
SyntaxDescription = "(" whsp
    numericoid whsp
    [ "DESC" qdstring ]
    whsp ")"
```

`ldaptor.schema.extractWord(text)`

`ldaptor.schema.peekWord(text)`

3.3.20 ldaptor.testutil module

Utilities for writing Twisted unit tests and debugging.

class `ldaptor.testutil.FakeTransport(proto)`

loseConnection()

class `ldaptor.testutil.LDAPClientTestDriver(*responses)`

A test driver that looks somewhat like a real LDAPClient.

Pass in a list of lists of LDAPProtocolResponses. For each sent LDAP message, the first item of said list is iterated through, and all the items are sent as responses to the callback. The sent LDAP messages are stored in `self.sent`, so you can assert that the sent messages are what they are supposed to be.

It is also possible to include a Failure instance instead of a list of LDAPProtocolResponses which will cause the `errback` to be called with the failure.

assertNothingSent()

assertSent(*shouldBeSent)

connectionLost(reason=None)

Called when TCP connection has been lost

connectionMade()

TCP connection has opened

send(op)

send_multiResponse(op, handler, *args, **kwargs)

send_noResponse(op)

unbind()

`ldaptor.testutil.calltrace()`

Print out all function calls. For debug use only.

`ldaptor.testutil.createServer(proto, *responses, **kw)`

`ldaptor.testutil.mustRaise(dummy)`

3.3.21 ldaptor.usage module

class `ldaptor.usage.Options`

Bases: `twisted.python.usage.Options`

optParameters = ()

postOptions()

```
class ldaptor.usage.Options_base
    Bases: ldaptor.usage.Options_base_optional

    postOptions_base ()

class ldaptor.usage.Options_base_optional

    optParameters = (('base', None, None, 'LDAP base dn'),)

class ldaptor.usage.Options_bind

    optParameters = (('binddn', None, None, 'use Distinguished Name to bind to the directory'), ('bind-auth-fd', None, None, 'bind authentication file descriptor'),)
    postOptions_bind_auth_fd_numeric ()

class ldaptor.usage.Options_bind_mandatory
    Bases: ldaptor.usage.Options_bind

    postOptions_bind_mandatory ()

class ldaptor.usage.Options_scope

    optParameters = (('scope', None, 'sub', 'LDAP search scope (one of base, one, sub)'),)
    postOptions_scope ()

class ldaptor.usage.Options_service_location

    opt_service_location (value)
        Service location, in the form BASEDN:HOST[:PORT]
    postOptions_service_location ()
```

3.3.22 Module contents

4.1 Changelog

4.1.1 Release 14.0 (2014-10-31)

Ldaptor has a new version schema. As a first-party library we now follow Twisted's example.

License

- Ldaptor's original author [Tommi Virtanen](#) changed the license to the MIT (Expat) license.
- `ldaptor.md4` has been replaced by a 3-clause BSD version.

API Changes

- Ldaptor client and server: None
- Everything having to do with `webui` and `Nevow` have been *removed*.

Features

- [Travis CI](#) is now used for continuous integration.
- Test coverage is now measured. We're currently at around 75%.
- `tox` is used now to test Ldaptor on all combinations of pypy, Python 2.6, Python 2.7 and Twisted versions from 10.0 until 14.0.
- A few ordering bugs that were exposed by that and are fixed now.
- `ldaptor.protocols.pureldap.LDAPExtendedRequest` now has additional tests.
- The new `ldaptor.protocols.pureldap.LDAPAbandonRequest` adds support for abandoning requests.
- `ldaptor.protocols.pureldap.LDAPBindRequest` has basic SASL support now. Higher-level APIs like `ldapclient` don't expose it yet though.

Bugfixes

- `ldaptor.protocols.ldap.Ldapclient`'s now uses `log.msg` for it's debug listing instead of the non-Twisted `log.debug`.
- String literal exceptions have been replaced by real Exceptions.
- “`bin/ldaptor-ldap2passwd -help`” now does not throws an exception anymore ([debian bug #526522](#)).
- `ldaptor.delta.Modification` and `ldaptor.protocols.ldap.Ldapsyntax.PasswordSetAggregateError` that are used for adding contacts now handle unicode arguments properly.
- `ldaptor.protocols.pureldap.LDAPExtendedRequest`'s constructor now handles STARTTLS in accordance to [RFC2251](#) so the constructor of `ldaptor.protocols.pureldap.LDAPStartTLSRequest` doesn't fail anymore.
- `ldaptor.protocols.ldap.Ldapserver.BaseLDAPServer` now uses the correct exception module in `dataReceived`.
- `ldaptor.protocols.ldap.Ldaperrors.LDAPException`: “Fix deprecated exception error”
- `bin/ldaptor-find-server` now imports `dns` from the correct twisted modules.
- `bin/ldaptor-find-server` now only prints SRV records.
- `ldaptor.protocols.ldap.Ldapsyntax.LDAPEntryWithClient` now correctly propagates errors on `search()`. The test suite has been adapted appropriately.
- `ldaptor.protocols.ldap.Ldapconnector.LDAPConnector` now supports specifying a local address when connecting to a server.
- The new `ldaptor.protocols.pureldap.LDAPSearchResultReference` now prevents ldaptor from choking on results containing `SearchResultReference` (usually from Active Directory servers). It is currently only a stub and silently ignored.
- `hashlib` and built-in `set()` are now used instead of deprecated modules.

Improved Documentation

- Added, updated and reworked documentation using Sphinx. `Dia` is required for converting diagrams to `svg/png`, this might change in the future.
- `Dia` is now invoked correctly for diagram generation in a headless environment.
- The documentation is now hosted on <https://ldaptor.readthedocs.org/>.

4.1.2 Prehistory

All versions up to and including 0.0.43 didn't have a changelog.

4.2 Status and History

Ldaptor was created by [Tommi Virtanen](#) who developed it during the years 2001-2008. From 2007 and onwards mainly bug fixes were added, many contributed by Debian maintainers. Development picked back up in 2014 by [Bret Curtis](#) with Tommi's consent and was migrated to Twisted where it is a first-party Twisted library. Ldaptor can be found here:

<https://github.com/twisted/ldaptor>

The LDAP client library functionality is in active use. It is stable and works very well.

4.3 Contributions

4.3.1 How to Contribute

Head over to: <https://github.com/twisted/ldaptor> and submit your bugs or feature requests. If you wish to contribute code, just fork it, make a branch and send us a pull request. We'll review it, and push back if necessary.

Ldaptor generally follows the coding and documentation standards of the Twisted project.

4.3.2 Contributors

- Anton Gyllenberg
- Bret Curtis
- David Strauss
- HawkOwl
- Hynek Schlawack
- Kenny MacDermid
- Stefan Andersson
- Tommi Virtanen

Indices and tables

- `genindex`
- `modindex`
- `search`

|

- ldaptor, 64
- ldaptor.attributeset, 51
- ldaptor.checkers, 51
- ldaptor.compat, 51
- ldaptor.config, 55
- ldaptor.delta, 56
- ldaptor.dns, 57
- ldaptor.entry, 57
- ldaptor.entryhelpers, 58
- ldaptor.generate_password, 58
- ldaptor.inmemory, 58
- ldaptor.insensitive, 59
- ldaptor.interfaces, 59
- ldaptor.ldapfilter, 59
- ldaptor.ldiftree, 60
- ldaptor.md4, 60
- ldaptor.numberalloc, 61
- ldaptor.protocols, 50
- ldaptor.protocols.ldap, 39
- ldaptor.protocols.ldap.autofill, 26
- ldaptor.protocols.ldap.autofill.posixAccount, 26
- ldaptor.protocols.ldap.autofill.sambaAccount, 26
- ldaptor.protocols.ldap.autofill.sambaSamAccount, 26
- ldaptor.protocols.ldap.distinguishedname, 26
- ldaptor.protocols.ldap.fetchschema, 27
- ldaptor.protocols.ldap.ldapclient, 27
- ldaptor.protocols.ldap.ldapconnector, 29
- ldaptor.protocols.ldap.ldaperrors, 29
- ldaptor.protocols.ldap.ldapserver, 33
- ldaptor.protocols.ldap.ldapsyntax, 34
- ldaptor.protocols.ldap.ldif, 37
- ldaptor.protocols.ldap.ldifdelta, 37
- ldaptor.protocols.ldap.ldifprotocol, 38
- ldaptor.protocols.ldap.proxy, 39
- ldaptor.protocols.ldap.svcbindproxy, 39
- ldaptor.protocols.pureber, 40
- ldaptor.protocols.pureldap, 41
- ldaptor.samba, 51
- ldaptor.samba.smbpassword, 50
- ldaptor.schema, 61
- ldaptor.testutil, 63
- ldaptor.usage, 63

A

- Add (class in `ldaptor.delta`), 56
- `add()` (`ldaptor.protocols.ldap.ldapsyntax.JournaledLDAPAttributeSet` method), 35
- `add_doc()` (in module `ldaptor.compat`), 55
- `addAutofiller()` (`ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithAutofill` method), 35
- `addChild()` (`ldaptor.inmemory.ReadOnlyInMemoryLDAPEntry` method), 59
- `addChild()` (`ldaptor.ldifree.LDIFTreeEntry` method), 60
- `addChild()` (`ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithClient` method), 35
- `addFailed()` (`ldaptor.inmemory.InMemoryLDIFProtocol` method), 59
- AddOp (class in `ldaptor.delta`), 56
- `alloc_ldap_message_id()` (in module `ldaptor.protocols.pureldap`), 50
- `asLDAP()` (`ldaptor.delta.Modification` method), 56
- `asLDAP()` (`ldaptor.delta.ModifyOp` method), 56
- `asLDIF()` (in module `ldaptor.protocols.ldap.ldif`), 37
- `asLDIF()` (`ldaptor.delta.Add` method), 56
- `asLDIF()` (`ldaptor.delta.AddOp` method), 56
- `asLDIF()` (`ldaptor.delta.Delete` method), 56
- `asLDIF()` (`ldaptor.delta.DeleteOp` method), 56
- `asLDIF()` (`ldaptor.delta.ModifyOp` method), 56
- `asLDIF()` (`ldaptor.delta.Replace` method), 57
- ASN1ParserThingie (class in `ldaptor.schema`), 61
- `assertNothingSent()` (`ldaptor.testutil.LDAPClientTestDriver` method), 63
- `assertSent()` (`ldaptor.testutil.LDAPClientTestDriver` method), 63
- `asText()` (`ldaptor.protocols.pureldap.LDAPFilter_and` method), 45
- `asText()` (`ldaptor.protocols.pureldap.LDAPFilter_approxMatch` method), 45
- `asText()` (`ldaptor.protocols.pureldap.LDAPFilter_equalityMatch` method), 45
- `asText()` (`ldaptor.protocols.pureldap.LDAPFilter_greaterOrEqual` method), 45
- `asText()` (`ldaptor.protocols.pureldap.LDAPFilter_lessOrEqual` method), 45
- `asText()` (`ldaptor.protocols.pureldap.LDAPFilter_not` method), 46
- `asText()` (`ldaptor.protocols.pureldap.LDAPFilter_or` method), 46
- `asText()` (`ldaptor.protocols.pureldap.LDAPFilter_present` method), 46
- `asText()` (`ldaptor.protocols.pureldap.LDAPFilter_substrings` method), 46
- `asText()` (`ldaptor.protocols.pureldap.LDAPFilter_substrings_any` method), 46
- `asText()` (`ldaptor.protocols.pureldap.LDAPFilter_substrings_final` method), 46
- `asText()` (`ldaptor.protocols.pureldap.LDAPFilter_substrings_initial` method), 46
- `aton()` (in module `ldaptor.dns`), 57
- `aton_numbits()` (in module `ldaptor.dns`), 57
- `aton_octets()` (in module `ldaptor.dns`), 57
- `attributeAsLDIF()` (in module `ldaptor.protocols.ldap.ldif`), 37
- `attributeAsLDIF_base64()` (in module `ldaptor.protocols.ldap.ldif`), 37
- `attributes` (`ldaptor.protocols.pureldap.LDAPSearchRequest` attribute), 49
- `attributeType` (`ldaptor.protocols.ldap.distinguishedname.LDAPAttributeType` attribute), 27
- AttributeTypeDescription (class in `ldaptor.schema`), 61
- `attributeTypesAndValues` (`ldaptor.protocols.ldap.distinguishedname.RelativeDistinguishedName` attribute), 27
- Autofill_posix (class in `ldaptor.protocols.ldap.autofill.posixAccount`), 26
- Autofill_samba (class in `ldaptor.protocols.ldap.autofill.sambaAccount`), 26
- Autofill_samba (class in `ldaptor.protocols.ldap.autofill.sambaSamAccount`), 26
- AutofillException, 26

B

- b() (in module `ldaptor.compat`), 51
- `ascii_to_str()` (in module `ldaptor.compat`), 54
- `base64_encode()` (in module `ldaptor.protocols.ldap.ldif`), 37
- `base_string_types` (in module `ldaptor.compat`), 51
- `baseDN` (`ldaptor.config.LDAPConfig` attribute), 55
- `BaseLDAPEntry` (class in `ldaptor.entry`), 57
- `BaseLDAPServer` (class in `ldaptor.protocols.ldap.ldapserver`), 33
- `baseObject` (`ldaptor.protocols.pureldap.LDAPSearchRequest` attribute), 49
- `ber2int()` (in module `ldaptor.protocols.pureber`), 41
- `BERBase` (class in `ldaptor.protocols.pureber`), 40
- `BERBoolean` (class in `ldaptor.protocols.pureber`), 40
- `berDecodeLength()` (in module `ldaptor.protocols.pureber`), 41
- `berDecodeMultiple()` (in module `ldaptor.protocols.pureber`), 41
- `berDecodeObject()` (in module `ldaptor.protocols.pureber`), 41
- `berdecoder` (`ldaptor.protocols.ldap.ldapclient.LDAPClient` attribute), 27
- `berdecoder` (`ldaptor.protocols.ldap.ldapserver.BaseLDAPServer` attribute), 33
- `BERDecoderContext` (class in `ldaptor.protocols.pureber`), 40
- `BEREnumerated` (class in `ldaptor.protocols.pureber`), 40
- `BERException`, 40
- `BERExceptionInsufficientData`, 40
- `BERInteger` (class in `ldaptor.protocols.pureber`), 40
- `BERNull` (class in `ldaptor.protocols.pureber`), 40
- `BEROctetString` (class in `ldaptor.protocols.pureber`), 40
- `BERSequence` (class in `ldaptor.protocols.pureber`), 40
- `BERSequenceOf` (class in `ldaptor.protocols.pureber`), 41
- `BERSet` (class in `ldaptor.protocols.pureber`), 41
- `BERStructured` (class in `ldaptor.protocols.pureber`), 41
- `bind()` (`ldaptor.entry.BaseLDAPEntry` method), 57
- `bind()` (`ldaptor.protocols.ldap.ldapclient.LDAPClient` method), 27
- `bind()` (`ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithClient` method), 35
- `boundUser` (`ldaptor.protocols.ldap.ldapserver.LDAPServer` attribute), 34
- `buildAttributeSet()` (`ldaptor.entry.BaseLDAPEntry` method), 57
- `buildAttributeSet()` (`ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithClient` method), 35
- `byte_elem_value()` (in module `ldaptor.compat`), 54
- `bytes` (in module `ldaptor.compat`), 54
- `calltrace()` (in module `ldaptor.testutil`), 63
- `CannotRemoveRDNErrors`, 34
- `capitalize()` (`ldaptor.compat.unicode` method), 51
- `center()` (`ldaptor.compat.unicode` method), 51
- `checkControls()` (`ldaptor.protocols.ldap.ldapserver.BaseLDAPServer` method), 33
- `children()` (`ldaptor.inmemory.ReadOnlyInMemoryLDAPEntry` method), 59
- `children()` (`ldaptor.ldifree.LDIFTreeEntry` method), 60
- `clear()` (`ldaptor.protocols.ldap.ldapsyntax.JournaleLDAPAttributeSet` method), 35
- `client` (`ldaptor.protocols.ldap.proxy.Proxy` attribute), 39
- `commit()` (`ldaptor.entry.EditableLDAPEntry` method), 57
- `commit()` (`ldaptor.inmemory.ReadOnlyInMemoryLDAPEntry` method), 59
- `commit()` (`ldaptor.ldifree.LDIFTreeEntry` method), 60
- `commit()` (`ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithClient` method), 35
- `connect()` (`ldaptor.protocols.ldap.ldapconnector.LDAPClientCreator` method), 29
- `connect()` (`ldaptor.protocols.ldap.ldapconnector.LDAPConnector` method), 29
- `connectAnonymously()` (`ldaptor.protocols.ldap.ldapconnector.LDAPClientCreator` method), 29
- `connectionLost()` (`ldaptor.inmemory.InMemoryLDIFProtocol` method), 59
- `connectionLost()` (`ldaptor.ldifree.StoreParsedLDIF` method), 60
- `connectionLost()` (`ldaptor.protocols.ldap.ldapclient.LDAPClient` method), 27
- `connectionLost()` (`ldaptor.protocols.ldap.ldapserver.BaseLDAPServer` method), 33
- `connectionLost()` (`ldaptor.protocols.ldap.ldifprotocol.LDIF` method), 38
- `connectionLost()` (`ldaptor.protocols.ldap.proxy.Proxy` method), 39
- `connectionLost()` (`ldaptor.testutil.LDAPClientTestDriver` method), 63
- `connectionMade()` (`ldaptor.protocols.ldap.ldapclient.LDAPClient` method), 28
- `connectionMade()` (`ldaptor.protocols.ldap.ldapserver.BaseLDAPServer` method), 33
- `connectionMade()` (`ldaptor.protocols.ldap.proxy.Proxy` method), 39
- `connectionMade()` (`ldaptor.testutil.LDAPClientTestDriver` method), 63

C

`callable()` (in module `ldaptor.compat`), 51

- contains() (Ldaptor.protocols.ldap.distinguishedname.DistinguishedName method), 26
- containsNonprintable() (in module Ldaptor.protocols.ldap.ldif), 37
- controlValue (Ldaptor.protocols.pureldap.LDAPControl attribute), 44
- copy() (Ldaptor.attributeset.LDAPAttributeSet method), 51
- copy() (Ldaptor.config.LDAPConfig method), 55
- count() (Ldaptor.compat.unicode method), 51
- count() (Ldaptor.protocols.ldap.distinguishedname.RelativeDistinguishedName method), 27
- createServer() (in module Ldaptor.testutil), 63
- credentialInterfaces (Ldaptor.checkers.LDAPBindingChecker attribute), 51
- criticality (Ldaptor.protocols.pureldap.LDAPControl attribute), 44
- ## D
- data (Ldaptor.protocols.ldap.ldifprotocol.LDIF attribute), 38
- dataReceived() (Ldaptor.protocols.ldap.Ldapclient.LDAPClient method), 28
- dataReceived() (Ldaptor.protocols.ldap.Ldapserver.BaseLDAPServer method), 33
- debug (Ldaptor.protocols.ldap.Ldapclient.LDAPClient attribute), 28
- debug (Ldaptor.protocols.ldap.Ldapserver.BaseLDAPServer attribute), 33
- decode() (Ldaptor.compat.unicode method), 52
- Delete (class in Ldaptor.delta), 56
- delete() (Ldaptor.entry.EditableLDAPEntry method), 58
- delete() (Ldaptor.inmemory.ReadOnlyInMemoryLDAPEntry method), 59
- delete() (Ldaptor.ldiftree.LDIFTreeEntry method), 60
- delete() (Ldaptor.protocols.ldap.Ldapsyntax.LDAPEntryWithClient method), 35
- deleteChild() (Ldaptor.inmemory.ReadOnlyInMemoryLDAPEntry method), 59
- deleteChild() (Ldaptor.ldiftree.LDIFTreeEntry method), 60
- deleteOldrdn (Ldaptor.protocols.pureldap.LDAPModifyDNRequest attribute), 47
- DeleteOp (class in Ldaptor.delta), 56
- delimiter (Ldaptor.protocols.ldap.ldifprotocol.LDIF attribute), 38
- derefAliases (Ldaptor.protocols.pureldap.LDAPSearchRequest attribute), 49
- diff() (Ldaptor.entry.BaseLDAPEntry method), 57
- difference() (Ldaptor.attributeset.LDAPAttributeSet method), 51
- diffTree() (Ldaptor.entryhelpers.DiffTreeMixin method), 58
- DiffTreeMixin (class in Ldaptor.entryhelpers), 58
- DistinguishedName (class in Ldaptor.protocols.ldap.distinguishedname), 26
- dn (Ldaptor.entry.BaseLDAPEntry attribute), 57
- dn (Ldaptor.protocols.ldap.ldifprotocol.LDIF attribute), 38
- dnAttributes (Ldaptor.protocols.pureldap.LDAPMatchingRuleAssertion attribute), 46
- DNNotPresentError, 34
- ## E
- EditableLDAPEntry (class in Ldaptor.entry), 57
- encode() (Ldaptor.compat.unicode method), 52
- endswith() (Ldaptor.compat.unicode method), 52
- entry (Ldaptor.protocols.pureldap.LDAPModifyDNRequest attribute), 47
- errorMessage (Ldaptor.protocols.pureldap.LDAPBindResponse attribute), 43
- errReceived() (Ldaptor.generate_password.ReadPassword method), 58
- escape() (in module Ldaptor.protocols.ldap.distinguishedname), 27
- escape() (in module Ldaptor.protocols.pureldap), 50
- exit_err() (in module Ldaptor.compat), 55
- expandtabs() (Ldaptor.compat.unicode method), 52
- ExtendedRequest_LDAPPasswordModifyRequest() (Ldaptor.protocols.ldap.Ldapserver.LDAPServer method), 34
- extractWord() (in module Ldaptor.schema), 63
- ## F
- fail_LDAPAddRequest (Ldaptor.protocols.ldap.Ldapserver.LDAPServer attribute), 34
- fail_LDAPBindRequest (Ldaptor.protocols.ldap.Ldapserver.LDAPServer attribute), 34
- fail_LDAPBindRequest (Ldaptor.protocols.ldap.svcbindproxy.ServiceBindingProxy attribute), 39
- fail_LDAPDelRequest (Ldaptor.protocols.ldap.Ldapserver.LDAPServer attribute), 34
- fail_LDAPExtendedRequest (Ldaptor.protocols.ldap.Ldapserver.LDAPServer attribute), 34
- fail_LDAPModifyDNRequest (Ldaptor.protocols.ldap.Ldapserver.LDAPServer attribute), 34
- fail_LDAPModifyRequest (Ldaptor.protocols.ldap.Ldapserver.LDAPServer attribute), 34
- fail_LDAPSearchRequest (Ldaptor.protocols.ldap.Ldapserver.LDAPServer attribute), 34

- failDefault() (ldaptor.protocols.ldap.ldapsrv.BaseLDAPServer class method), 33
 - FakeTransport (class in ldaptor.testutil), 63
 - fallback (ldaptor.protocols.ldap.svcbindproxy.ServiceBindingProxy attribute), 39
 - fetch() (in module ldaptor.protocols.ldap.fetchschema), 27
 - fetch() (ldaptor.inmemory.ReadOnlyInMemoryLDAPEntry method), 59
 - fetch() (ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithClient method), 35
 - filter (ldaptor.protocols.pureldap.LDAPSearchRequest attribute), 49
 - find() (ldaptor.compat.unicode method), 52
 - format() (ldaptor.compat.unicode method), 52
 - freeNumberGuesser (class in ldaptor.numberalloc), 61
 - fromBER() (ldaptor.protocols.pureber.BERBoolean class method), 40
 - fromBER() (ldaptor.protocols.pureber.BERInteger class method), 40
 - fromBER() (ldaptor.protocols.pureber.BERNull class method), 40
 - fromBER() (ldaptor.protocols.pureber.BEROctetString class method), 40
 - fromBER() (ldaptor.protocols.pureber.BERSequence class method), 41
 - fromBER() (ldaptor.protocols.pureldap.LDAPAddRequest class method), 41
 - fromBER() (ldaptor.protocols.pureldap.LDAPAttributeValueAssertion class method), 42
 - fromBER() (ldaptor.protocols.pureldap.LDAPBindRequest class method), 43
 - fromBER() (ldaptor.protocols.pureldap.LDAPBindResponse class method), 43
 - fromBER() (ldaptor.protocols.pureldap.LDAPControl class method), 44
 - fromBER() (ldaptor.protocols.pureldap.LDAPControls class method), 44
 - fromBER() (ldaptor.protocols.pureldap.LDAPExtendedRequest class method), 44
 - fromBER() (ldaptor.protocols.pureldap.LDAPExtendedResponse class method), 44
 - fromBER() (ldaptor.protocols.pureldap.LDAPFilter_not class method), 46
 - fromBER() (ldaptor.protocols.pureldap.LDAPFilter_substrings class method), 46
 - fromBER() (ldaptor.protocols.pureldap.LDAPFilterSet class method), 45
 - fromBER() (ldaptor.protocols.pureldap.LDAPMatchingRuleAssertion class method), 46
 - fromBER() (ldaptor.protocols.pureldap.LDAPMessage class method), 47
 - fromBER() (ldaptor.protocols.pureldap.LDAPModifyDNRequest class method), 47
 - fromBER() (ldaptor.protocols.pureldap.LDAPModifyRequest class method), 48
 - fromBER() (ldaptor.protocols.pureldap.LDAPResult class method), 49
 - fromBER() (ldaptor.protocols.pureldap.LDAPSearchRequest class method), 49
 - fromBER() (ldaptor.protocols.pureldap.LDAPSearchResultEntry class method), 50
 - fromBER() (ldaptor.protocols.pureldap.LDAPSearchResultReference class method), 50
 - fromLDAP() (ldaptor.delta.ModifyOp class method), 56
 - fromLDIFFile() (in module ldaptor.inmemory), 59
 - fromLDIFFile() (in module ldaptor.protocols.ldap.ldifdelta), 37
- ## G
- generate() (in module ldaptor.generate_password), 58
 - get() (in module ldaptor.ldiftree), 60
 - get() (in module ldaptor.protocols.ldap.ldaperrors), 33
 - get() (ldaptor.entry.BaseLDAPEntry method), 57
 - get() (ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithClient method), 35
 - get_method_function() (in module ldaptor.compat), 55
 - getBaseDN() (ldaptor.config.LDAPConfig method), 55
 - getDomainName() (ldaptor.protocols.ldap.distinguishedname.DistinguishedName method), 26
 - getFreeNumber() (in module ldaptor.numberalloc), 61
 - getIdentityBaseDN() (ldaptor.config.LDAPConfig method), 55
 - getIdentitySearch() (ldaptor.config.LDAPConfig method), 55
 - getRootDSE() (ldaptor.protocols.ldap.ldapsrv.LDAPServer method), 34
 - getServiceLocationOverrides() (ldaptor.config.LDAPConfig method), 55
 - gotEntry() (ldaptor.inmemory.InMemoryLDIFProtocol method), 59
 - gotEntry() (ldaptor.ldiftree.StoreParsedLDIF method), 60
 - gotEntry() (ldaptor.protocols.ldap.ldifprotocol.LDIF method), 38
 - guess() (ldaptor.numberalloc.ldapGuesser method), 61
- ## H
- handle() (ldaptor.protocols.ldap.ldapclient.LDAPClient method), 28
 - handle() (ldaptor.protocols.ldap.ldapsrv.BaseLDAPServer method), 34
 - handle_LDAPAddRequest() (ldaptor.protocols.ldap.ldapsrv.LDAPServer method), 34
 - handle_LDAPBindRequest() (ldaptor.protocols.ldap.ldapsrv.LDAPServer method), 34

handle_LDAPBindRequest() (ldap- Identities (ldapTOR.protocols.pureldap.LDAPBERDecoderContext_LDAPC
 tor.protocols.ldap.svcbindproxy.ServiceBindingProxy attribute), 42
 method), 39 Identities (ldapTOR.protocols.pureldap.LDAPBERDecoderContext_LDAPEx
 handle_LDAPDelRequest() (ldap- attribute), 43
 tor.protocols.ldap.ldapsrvr.LDAPServer Identities (ldapTOR.protocols.pureldap.LDAPBERDecoderContext_LDAPEx
 method), 34 attribute), 43
 handle_LDAPExtendedRequest() (ldap- Identities (ldapTOR.protocols.pureldap.LDAPBERDecoderContext_LDAPMe
 tor.protocols.ldap.ldapsrvr.LDAPServer attribute), 43
 method), 34 Identities (ldapTOR.protocols.pureldap.LDAPBERDecoderContext_LDAPPa
 handle_LDAPModifyDNRequest() (ldap- attribute), 43
 tor.protocols.ldap.ldapsrvr.LDAPServer Identities (ldapTOR.protocols.pureldap.LDAPBERDecoderContext_Matching
 method), 34 attribute), 43
 handle_LDAPModifyRequest() (ldap- Identities (ldapTOR.protocols.pureldap.LDAPBERDecoderContext_ModifyD
 tor.protocols.ldap.ldapsrvr.LDAPServer attribute), 43
 method), 34 Identities (ldapTOR.protocols.pureldap.LDAPBERDecoderContext_TopLevel
 handle_LDAPSearchRequest() (ldap- attribute), 43
 tor.protocols.ldap.ldapsrvr.LDAPServer identityBaseDN (ldapTOR.config.LDAPConfig attribute),
 method), 34 55
 handle_LDAPUnbindRequest() (ldap- identitySearch (ldapTOR.config.LDAPConfig attribute), 56
 tor.protocols.ldap.ldapsrvr.LDAPServer imap (class in ldapTOR.compat), 55
 method), 34 index() (ldapTOR.compat.unicode method), 52
 handle_LDAPUnbindRequest() (ldap- inherit() (ldapTOR.protocols.pureber.BERDecoderContext
 tor.protocols.ldap.proxy.Proxy method),
 39 method), 40
 handleUnknown() (ldap- init() (in module ldapTOR.protocols.ldap.ldaperrors), 33
 tor.protocols.ldap.ldapsrvr.BaseLDAPServer InMemoryLDIFProtocol (class in ldapTOR.inmemory), 58
 method), 34 InsensitiveString (class in ldapTOR.insensitive), 59
 handleUnknown() (ldapTOR.protocols.ldap.proxy.Proxy int2ber() (in module ldapTOR.protocols.pureber), 41
 method), 39 int2berlen() (in module ldapTOR.protocols.pureber), 41
 has_key() (ldapTOR.entry.BaseLDAPEntry method), 57 intersection() (ldapTOR.attributeset.LDAPAttributeSet
 has_key() (ldapTOR.protocols.ldap.ldapsyntax.LDAPEntryWithClient method), 51
 method), 35 InvalidRelativeDistinguishedName, 27
 hasMember() (ldapTOR.entry.BaseLDAPEntry method), 57 irange (in module ldapTOR.compat), 55
 header() (in module ldapTOR.protocols.ldap.ldif), 37 isalnum() (ldapTOR.compat.unicode method), 52
 isalpha() (ldapTOR.compat.unicode method), 52
 isdecimal() (ldapTOR.compat.unicode method), 52
 isdigit() (ldapTOR.compat.unicode method), 52
 islower() (ldapTOR.compat.unicode method), 52
 isnumeric() (ldapTOR.compat.unicode method), 52
 isspace() (ldapTOR.compat.unicode method), 52
 istitle() (ldapTOR.compat.unicode method), 52
 isupper() (ldapTOR.compat.unicode method), 53
 items() (ldapTOR.entry.BaseLDAPEntry method), 57
 items() (ldapTOR.protocols.ldap.ldapsyntax.LDAPEntryWithClient
 method), 35
 iter_byte_values() (in module ldapTOR.compat), 55
 iteritems() (in module ldapTOR.compat), 55
 itervalues() (in module ldapTOR.compat), 55
 join() (ldapTOR.compat.unicode method), 53
 join_byte_elems() (in module ldapTOR.compat), 54
 join_bytes() (in module ldapTOR.compat), 54

- join_unicode() (in module ldaptor.compat), 54
- journal() (ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithAutoFill (class in ldaptor.protocols.pureldap), 43 method), 35
- journal() (ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithClient (class in ldaptor.protocols.pureldap), 43 method), 35
- JournaledLDAPAttributeSet (class in ldaptor.protocols.ldap.ldapsyntax), 35
- ## K
- keys() (ldaptor.entry.BaseLDAPEntry method), 57
- keys() (ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithClient (class in ldaptor.protocols.pureldap), 43 method), 35
- ## L
- lastLine (ldaptor.protocols.ldap.ldifprotocol.LDIF attribute), 38
- LDAPAbandonRequest (class in ldaptor.protocols.pureldap), 41
- LDAPAddRequest (class in ldaptor.protocols.pureldap), 41
- LDAPAddResponse (class in ldaptor.protocols.pureldap), 41
- LDAPAdminLimitExceeded, 29
- LDAPAffectsMultipleDSAs, 29
- LDAPAliasDereferencingProblem, 29
- LDAPAliasProblem, 29
- LDAPAssertionValue (class in ldaptor.protocols.pureldap), 42
- LDAPAttributeDescription (class in ldaptor.protocols.pureldap), 42
- LDAPAttributeOrValueExists, 29
- LDAPAttributeSet (class in ldaptor.attributeset), 51
- LDAPAttributeTypeAndValue (class in ldaptor.protocols.ldap.distinguishedname), 27
- LDAPAttributeValue (class in ldaptor.protocols.pureldap), 42
- LDAPAttributeValueAssertion (class in ldaptor.protocols.pureldap), 42
- LDAPAuthMethodNotSupported, 29
- LDAPBERDecoderContext (class in ldaptor.protocols.pureldap), 42
- LDAPBERDecoderContext_BindResponse (class in ldaptor.protocols.pureldap), 42
- LDAPBERDecoderContext_Filter (class in ldaptor.protocols.pureldap), 42
- LDAPBERDecoderContext_Filter_substrings (class in ldaptor.protocols.pureldap), 42
- LDAPBERDecoderContext_LDAPBindRequest (class in ldaptor.protocols.pureldap), 42
- LDAPBERDecoderContext_LDAPControls (class in ldaptor.protocols.pureldap), 42
- LDAPBERDecoderContext_LDAPExtendedRequest (class in ldaptor.protocols.pureldap), 42
- LDAPBERDecoderContext_LDAPExtendedResponse (class in ldaptor.protocols.pureldap), 43
- LDAPBERDecoderContext_LDAPMessage (class in ldaptor.protocols.pureldap), 43
- LDAPBERDecoderContext_LDAPPasswordModifyRequest (class in ldaptor.protocols.pureldap), 43
- LDAPBERDecoderContext_MatchingRuleAssertion (class in ldaptor.protocols.pureldap), 43
- LDAPBERDecoderContext_ModifyDNRequest (class in ldaptor.protocols.pureldap), 43
- LDAPBERDecoderContext_TopLevel (class in ldaptor.protocols.pureldap), 43
- LDAPBindingChecker (class in ldaptor.checkers), 51
- LDAPBindRequest (class in ldaptor.protocols.pureldap), 43
- LDAPBindResponse (class in ldaptor.protocols.pureldap), 43
- LDAPBindResponse_serverSaslCreds (class in ldaptor.protocols.pureldap), 44
- LDAPBusy, 29
- LDAPCannotRemoveRootError, 59, 60
- LDAPClient (class in ldaptor.protocols.ldap.ldapclient), 27
- LDAPClientConnectionLostException, 28
- LDAPClientCreator (class in ldaptor.protocols.ldap.ldapconnector), 29
- LDAPClientTestDriver (class in ldaptor.testutil), 63
- LDAPCompareFalse, 30
- LDAPCompareTrue, 30
- LDAPConfidentialityRequired, 30
- LDAPConfig (class in ldaptor.config), 55
- LDAPConnector (class in ldaptor.protocols.ldap.ldapconnector), 29
- LDAPConstraintViolation, 30
- LDAPControl (class in ldaptor.protocols.pureldap), 44
- LDAPControls (class in ldaptor.protocols.pureldap), 44
- LDAPDelRequest (class in ldaptor.protocols.pureldap), 44
- LDAPDelResponse (class in ldaptor.protocols.pureldap), 44
- LDAPEntry (in module ldaptor.protocols.ldap.ldapsyntax), 35
- LDAPEntryAlreadyExists, 30
- LDAPEntryWithAutoFill (class in ldaptor.protocols.ldap.ldapsyntax), 35
- LDAPEntryWithClient (class in ldaptor.protocols.ldap.ldapsyntax), 35
- LDAPException, 30
- LDAPExtendedRequest (class in ldaptor.protocols.pureldap), 44
- LDAPExtendedResponse (class in ldaptor.protocols.pureldap), 44
- LDAPFilter (class in ldaptor.protocols.pureldap), 45
- LDAPFilter_and (class in ldaptor.protocols.pureldap), 45

LDAPFilter_approxMatch (class in ldap-tor.protocols.pureldap), 45

LDAPFilter_equalityMatch (class in ldap-tor.protocols.pureldap), 45

LDAPFilter_extensibleMatch (class in ldap-tor.protocols.pureldap), 45

LDAPFilter_greaterOrEqual (class in ldap-tor.protocols.pureldap), 45

LDAPFilter_lessOrEqual (class in ldap-tor.protocols.pureldap), 45

LDAPFilter_not (class in ldap-tor.protocols.pureldap), 45

LDAPFilter_or (class in ldap-tor.protocols.pureldap), 46

LDAPFilter_present (class in ldap-tor.protocols.pureldap), 46

LDAPFilter_substrings (class in ldap-tor.protocols.pureldap), 46

LDAPFilter_substrings_any (class in ldap-tor.protocols.pureldap), 46

LDAPFilter_substrings_final (class in ldap-tor.protocols.pureldap), 46

LDAPFilter_substrings_initial (class in ldap-tor.protocols.pureldap), 46

LDAPFilterSet (class in ldap-tor.protocols.pureldap), 45

ldapGuesser (class in ldap-tor.numberalloc), 61

LDAPInappropriateAuthentication, 30

LDAPInappropriateMatching, 30

LDAPInsufficientAccessRights, 30

LDAPInteger (class in ldap-tor.protocols.pureldap), 46

LDAPInvalidAttributeSyntax, 30

LDAPInvalidCredentials, 31

LDAPInvalidDNyntax, 31

LDAPLoopDetect, 31

LDAPMatchingRuleAssertion (class in ldap-tor.protocols.pureldap), 46

LDAPMatchingRuleAssertion_dnAttributes (class in ldap-tor.protocols.pureldap), 47

LDAPMatchingRuleAssertion_matchingRule (class in ldap-tor.protocols.pureldap), 47

LDAPMatchingRuleAssertion_matchValue (class in ldap-tor.protocols.pureldap), 47

LDAPMatchingRuleAssertion_type (class in ldap-tor.protocols.pureldap), 47

LDAPMatchingRuleId (class in ldap-tor.protocols.pureldap), 47

LDAPMessage (class in ldap-tor.protocols.pureldap), 47

LDAPModifyDNRequest (class in ldap-tor.protocols.pureldap), 47

LDAPModifyDNResponse (class in ldap-tor.protocols.pureldap), 47

LDAPModifyDNResponse_newSuperior (class in ldap-tor.protocols.pureldap), 48

LDAPModifyRequest (class in ldap-tor.protocols.pureldap), 48

LDAPModifyResponse (class in ldap-tor.protocols.pureldap), 48

LDAPNamingViolation, 31

LDAPNoSuchAttribute, 31

LDAPNoSuchObject, 31

LDAPNotAllowedOnNonLeaf, 31

LDAPNotAllowedOnRDN, 31

LDAPObjectClassModsProhibited, 31

LDAPObjectClassViolation, 31

LDAPOID (class in ldap-tor.protocols.pureldap), 48

LDAPOperationsError, 32

LDAPOther, 32

LDAPPasswordModifyRequest (class in ldap-tor.protocols.pureldap), 48

LDAPPasswordModifyRequest_newPasswd (class in ldap-tor.protocols.pureldap), 48

LDAPPasswordModifyRequest_oldPasswd (class in ldap-tor.protocols.pureldap), 48

LDAPPasswordModifyRequest_userIdentity (class in ldap-tor.protocols.pureldap), 48

LDAPProtocolError, 32

LDAPProtocolOp (class in ldap-tor.protocols.pureldap), 48

LDAPProtocolRequest (class in ldap-tor.protocols.pureldap), 48

LDAPProtocolResponse (class in ldap-tor.protocols.pureldap), 49

LDAPReferral, 32

LDAPReferral (class in ldap-tor.protocols.pureldap), 49

LDAPResponse (class in ldap-tor.protocols.pureldap), 49

LDAPResponseName (class in ldap-tor.protocols.pureldap), 49

LDAPResult (class in ldap-tor.protocols.ldap.ldaperrors), 32

LDAPResult (class in ldap-tor.protocols.pureldap), 49

LDAPSaslBindInProgress, 32

LDAPSearchRequest (class in ldap-tor.protocols.pureldap), 49

LDAPSearchResultDone (class in ldap-tor.protocols.pureldap), 49

LDAPSearchResultEntry (class in ldap-tor.protocols.pureldap), 49

LDAPSearchResultReference (class in ldap-tor.protocols.pureldap), 50

LDAPServer (class in ldap-tor.protocols.ldap.ldapserver), 34

LDAPServerConnectionLostException, 34

LDAPSizeLimitExceeded, 32

LDAPStartTLSBusyError, 28

LDAPStartTLSRequest (class in ldap-tor.protocols.pureldap), 50

LDAPString (class in ldap-tor.protocols.pureldap), 50

LDAPStrongAuthRequired, 32

LDAPTimeLimitExceeded, 32

ldaptor (module), 64

- ldaptor.attributeset (module), 51
 - ldaptor.checkers (module), 51
 - ldaptor.compat (module), 51
 - ldaptor.config (module), 55
 - ldaptor.delta (module), 56
 - ldaptor.dns (module), 57
 - ldaptor.entry (module), 57
 - ldaptor.entryhelpers (module), 58
 - ldaptor.generate_password (module), 58
 - ldaptor.inmemory (module), 58
 - ldaptor.insensitive (module), 59
 - ldaptor.interfaces (module), 59
 - ldaptor.ldapfilter (module), 59
 - ldaptor.ldiftree (module), 60
 - ldaptor.md4 (module), 60
 - ldaptor.numberalloc (module), 61
 - ldaptor.protocols (module), 50
 - ldaptor.protocols.ldap (module), 39
 - ldaptor.protocols.ldap.autofill (module), 26
 - ldaptor.protocols.ldap.autofill.posixAccount (module), 26
 - ldaptor.protocols.ldap.autofill.sambaAccount (module), 26
 - ldaptor.protocols.ldap.autofill.sambaSamAccount (module), 26
 - ldaptor.protocols.ldap.distinguishedname (module), 26
 - ldaptor.protocols.ldap.fetchschema (module), 27
 - ldaptor.protocols.ldap.ldapclient (module), 27
 - ldaptor.protocols.ldap.ldapconnector (module), 29
 - ldaptor.protocols.ldap.ldaperrors (module), 29
 - ldaptor.protocols.ldap.ldapservers (module), 33
 - ldaptor.protocols.ldap.ldapsyntax (module), 34
 - ldaptor.protocols.ldap.ldif (module), 37
 - ldaptor.protocols.ldap.ldifdelta (module), 37
 - ldaptor.protocols.ldap.ldifprotocol (module), 38
 - ldaptor.protocols.ldap.proxy (module), 39
 - ldaptor.protocols.ldap.svcbindproxy (module), 39
 - ldaptor.protocols.pureber (module), 40
 - ldaptor.protocols.pureldap (module), 41
 - ldaptor.samba (module), 51
 - ldaptor.samba.smbpassword (module), 50
 - ldaptor.schema (module), 61
 - ldaptor.testutil (module), 63
 - ldaptor.usage (module), 63
 - LDAPUnavailable, 32
 - LDAPUnavailableCriticalExtension, 33
 - LDAPUnbindRequest (class in ldaptor.protocols.pureldap), 50
 - LDAPUndefinedAttributeType, 33
 - LDAPUnknownError, 33
 - LDAPUnwillingToPerform, 33
 - LDIF (class in ldaptor.protocols.ldap.ldifprotocol), 38
 - LDIFDelta (class in ldaptor.protocols.ldap.ldifdelta), 37
 - LDIFDeltaAddMissingAttributesError, 37
 - LDIFDeltaDeleteHasJunkAfterChangeTypeError, 37
 - LDIFDeltaMissingChangeTypeError, 37
 - LDIFDeltaModificationDifferentAttributeTypeError, 37
 - LDIFDeltaModificationMissingEndDashError, 37
 - LDIFDeltaUnknownModificationError, 37
 - LDIFEntryStartsWithNonDNError, 38
 - LDIFEntryStartsWithSpaceError, 38
 - LDIFLineWithoutSemicolonError, 38
 - LDIFParseError, 38
 - LDIFTreeEntry (class in ldaptor.ldiftree), 60
 - LDIFTreeEntryContainsMultipleEntries, 60
 - LDIFTreeEntryContainsNoEntries, 60
 - LDIFTreeNoSuchObject, 60
 - LDIFTruncatedError, 38
 - LDIFUnsupportedVersionError, 38
 - LDIFVersionNotANumberError, 38
 - lineReceived() (ldaptor.protocols.ldap.ldifprotocol.LDIF method), 38
 - listOfRDNs (ldaptor.protocols.ldap.distinguishedname.DistinguishedName attribute), 27
 - ljust() (ldaptor.compat.unicode method), 53
 - lmap() (in module ldaptor.compat), 55
 - lmd5() (in module ldaptor.samba.smbpassword), 50
 - lmd5_locked() (in module ldaptor.samba.smbpassword), 50
 - loadConfig() (in module ldaptor.config), 56
 - logicalLineReceived() (ldaptor.protocols.ldap.ldifprotocol.LDIF method), 38
 - lookup() (ldaptor.inmemory.ReadOnlyInMemoryLDAPEntry method), 59
 - lookup() (ldaptor.ldiftree.LDIFTreeEntry method), 60
 - lookup() (ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithClient method), 35
 - lookup_id() (ldaptor.protocols.pureber.BERDecoderContext method), 40
 - lookupFailed() (ldaptor.inmemory.InMemoryLDIFProtocol method), 59
 - loseConnection() (ldaptor.testutil.FakeTransport method), 63
 - lower() (ldaptor.compat.unicode method), 53
 - lstrip() (ldaptor.compat.unicode method), 53
- ## M
- makeFilter() (in module ldaptor.checkers), 51
 - manyAsLDIF() (in module ldaptor.protocols.ldap.ldif), 37
 - match() (ldaptor.entryhelpers.MatchMixin method), 58
 - matchedDN (ldaptor.protocols.pureldap.LDAPBindResponse attribute), 43
 - matchingRule (ldaptor.protocols.pureldap.LDAPMatchingRuleAssertion attribute), 47
 - MatchingRuleDescription (class in ldaptor.schema), 61
 - MatchMixin (class in ldaptor.entryhelpers), 58
 - MatchNotImplemented, 36

matchValue (ldaptor.protocols.pureldap.LDAPMatchingRuleAssertion (attribute), 47)

md4() (in module ldaptor.md4), 60

message (ldaptor.protocols.ldap.ldaperrors.LDAPException attribute), 30

MissingBaseDNError, 56

MOD_SPEC_TO_DELTA (ldaptor.protocols.ldap.ldifdelta.LDIFDelta attribute), 37

mode (ldaptor.protocols.ldap.ldifprotocol.LDIF attribute), 38

Modification (class in ldaptor.delta), 56

modification (ldaptor.protocols.pureldap.LDAPModifyRequest attribute), 48

ModifyOp (class in ldaptor.delta), 56

move() (ldaptor.entry.EditableLDAPEntry method), 58

move() (ldaptor.inmemory.ReadOnlyInMemoryLDAPEntry method), 59

move() (ldaptor.ldifree.LDIFTreeEntry method), 60

move() (ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithClient method), 35

mustRaise() (in module ldaptor.testutil), 63

N

name (ldaptor.protocols.ldap.ldaperrors.LDAPAdminLimitExceeded attribute), 29

name (ldaptor.protocols.ldap.ldaperrors.LDAPAffectsMultipleDNs attribute), 29

name (ldaptor.protocols.ldap.ldaperrors.LDAPAliasDereferencingProblem attribute), 29

name (ldaptor.protocols.ldap.ldaperrors.LDAPAliasProblem attribute), 29

name (ldaptor.protocols.ldap.ldaperrors.LDAPAttributeOrValueExists attribute), 29

name (ldaptor.protocols.ldap.ldaperrors.LDAPAuthMethodNotSupported attribute), 29

name (ldaptor.protocols.ldap.ldaperrors.LDAPBusy attribute), 29

name (ldaptor.protocols.ldap.ldaperrors.LDAPCompareFalse attribute), 30

name (ldaptor.protocols.ldap.ldaperrors.LDAPCompareTrue attribute), 30

name (ldaptor.protocols.ldap.ldaperrors.LDAPConfidentialityRequired attribute), 30

name (ldaptor.protocols.ldap.ldaperrors.LDAPConstraintViolation attribute), 30

name (ldaptor.protocols.ldap.ldaperrors.LDAPEntryAlreadyExists attribute), 30

name (ldaptor.protocols.ldap.ldaperrors.LDAPInappropriateAuthentication attribute), 30

name (ldaptor.protocols.ldap.ldaperrors.LDAPInappropriateMatching attribute), 30

name (ldaptor.protocols.ldap.ldaperrors.LDAPInsufficientRights attribute), 30

name (ldaptor.protocols.ldap.ldaperrors.LDAPInvalidAttributeSyntax attribute), 30

name (ldaptor.protocols.ldap.ldaperrors.LDAPInvalidCredentials attribute), 31

name (ldaptor.protocols.ldap.ldaperrors.LDAPInvalidDNSyntax attribute), 31

name (ldaptor.protocols.ldap.ldaperrors.LDAPLoopDetect attribute), 31

name (ldaptor.protocols.ldap.ldaperrors.LDAPNamingViolation attribute), 31

name (ldaptor.protocols.ldap.ldaperrors.LDAPNoSuchAttribute attribute), 31

name (ldaptor.protocols.ldap.ldaperrors.LDAPNoSuchObject attribute), 31

name (ldaptor.protocols.ldap.ldaperrors.LDAPNotAllowedOnNonLeaf attribute), 31

name (ldaptor.protocols.ldap.ldaperrors.LDAPNotAllowedOnRDN attribute), 31

name (ldaptor.protocols.ldap.ldaperrors.LDAPObjectClassModsProhibited attribute), 31

name (ldaptor.protocols.ldap.ldaperrors.LDAPObjectClassViolation attribute), 31

name (ldaptor.protocols.ldap.ldaperrors.LDAPOperationsError attribute), 32

name (ldaptor.protocols.ldap.ldaperrors.LDAPOther attribute), 32

name (ldaptor.protocols.ldap.ldaperrors.LDAPProtocolError attribute), 32

name (ldaptor.protocols.ldap.ldaperrors.LDAPReferral attribute), 32

name (ldaptor.protocols.ldap.ldaperrors.LDAPResult attribute), 32

name (ldaptor.protocols.ldap.ldaperrors.LDAPSSaslBindInProgress attribute), 32

name (ldaptor.protocols.ldap.ldaperrors.LDAPSizeLimitExceeded attribute), 32

name (ldaptor.protocols.ldap.ldaperrors.LDAPStrongAuthRequired attribute), 32

name (ldaptor.protocols.ldap.ldaperrors.LDAPTimeLimitExceeded attribute), 32

name (ldaptor.protocols.ldap.ldaperrors.LDAPUnavailable attribute), 33

name (ldaptor.protocols.ldap.ldaperrors.LDAPUnavailableCriticalExtension attribute), 33

name (ldaptor.protocols.ldap.ldaperrors.LDAPUndefinedAttributeType attribute), 33

name (ldaptor.protocols.ldap.ldaperrors.LDAPUnwillingToPerform attribute), 33

name (ldaptor.protocols.ldap.ldaperrors.Success attribute), 33

name (ldaptor.protocols.ldap.ldaperrors.WarningContext() (ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithClient method), 35

need() (in module ldaptor.protocols.pureber), 41

- needs_answer (Ldaptor.protocols.pureldap.LDAPAbandonRequest attribute), 41
 - needs_answer (Ldaptor.protocols.pureldap.LDAPProtocolRequest attribute), 48
 - needs_answer (Ldaptor.protocols.pureldap.LDAPUnbindRequest attribute), 50
 - netmaskToNumbits() (in module Ldaptor.dns), 57
 - newrdn (Ldaptor.protocols.pureldap.LDAPModifyDNRequest attribute), 47
 - newSuperior (Ldaptor.protocols.pureldap.LDAPModifyDNRequest attribute), 47
 - next (Ldaptor.compat.imap attribute), 55
 - next() (in module Ldaptor.compat), 55
 - NoContainingNamingContext, 36
 - notify() (Ldaptor.protocols.ldap.autofill.posixAccount.Autofill_posix method), 26
 - notify() (Ldaptor.protocols.ldap.autofill.sambaAccount.Autofill_samba method), 26
 - notify() (Ldaptor.protocols.ldap.autofill.sambaSamAccount.Autofill_samba method), 26
 - nthash() (in module Ldaptor.samba.smbpassword), 50
 - ntoa() (in module Ldaptor.dns), 57
- O**
- object (Ldaptor.protocols.pureldap.LDAPModifyRequest attribute), 48
 - ObjectClassDescription (class in Ldaptor.schema), 62
 - ObjectDeletedError, 36
 - ObjectDirtyError, 36
 - ObjectInBadStateError, 36
 - ObjectMissingObjectClassException, 26
 - oid (Ldaptor.protocols.pureldap.LDAPPasswordModifyRequest attribute), 48
 - oid (Ldaptor.protocols.pureldap.LDAPStartTLSRequest attribute), 50
 - Operation (class in Ldaptor.delta), 56
 - opt_service_location() (Ldaptor.usage.Options_service_location method), 64
 - Options (class in Ldaptor.usage), 63
 - Options_base (class in Ldaptor.usage), 63
 - Options_base_optional (class in Ldaptor.usage), 64
 - Options_bind (class in Ldaptor.usage), 64
 - Options_bind_mandatory (class in Ldaptor.usage), 64
 - Options_scope (class in Ldaptor.usage), 64
 - Options_service_location (class in Ldaptor.usage), 64
 - optParameters (Ldaptor.usage.Options attribute), 63
 - optParameters (Ldaptor.usage.Options_base_optional attribute), 64
 - optParameters (Ldaptor.usage.Options_bind attribute), 64
 - optParameters (Ldaptor.usage.Options_scope attribute), 64
 - outReceived() (Ldaptor.generate_password.ReadPassword method), 58
- P**
- parent() (Ldaptor.inmemory.ReadOnlyInMemoryLDAPEntry method), 59
 - parent() (Ldaptor.ldifree.LDIFreeEntry method), 60
 - parseExtensible() (in module Ldaptor.ldapfilter), 59
 - parseFilter() (in module Ldaptor.ldapfilter), 59
 - parseMaybeSubstring() (in module Ldaptor.ldapfilter), 59
 - parseValue() (Ldaptor.protocols.ldap.ldifprotocol.LDIF method), 38
 - partition() (Ldaptor.compat.unicode method), 53
 - PasswordSetAborted, 36
 - PasswordSetAggregateError, 36
 - patch() (Ldaptor.delta.Add method), 56
 - patch() (Ldaptor.delta.AddOp method), 56
 - patch() (Ldaptor.delta.Delete method), 56
 - patch() (Ldaptor.delta.DeleteOp method), 56
 - patch() (Ldaptor.delta.Modification method), 56
 - patch() (Ldaptor.delta.ModifyOp method), 56
 - patch() (Ldaptor.delta.Operation method), 57
 - patch() (Ldaptor.delta.Replace method), 57
 - peekWord() (in module Ldaptor.schema), 63
 - pickServer() (Ldaptor.protocols.ldap.ldapconnector.LDAPConnector method), 29
 - postOptions() (Ldaptor.usage.Options method), 63
 - postOptions_base() (Ldaptor.usage.Options_base method), 64
 - postOptions_bind_auth_fd_numeric() (Ldaptor.usage.Options_bind method), 64
 - postOptions_bind_mandatory() (Ldaptor.usage.Options_bind_mandatory method), 64
 - postOptions_scope() (Ldaptor.usage.Options_scope method), 64
 - postOptions_service_location() (Ldaptor.usage.Options_service_location method), 64
 - print_() (in module Ldaptor.compat), 51
 - processEnded() (Ldaptor.generate_password.ReadPassword method), 58
 - protocol (Ldaptor.protocols.ldap.proxy.Proxy attribute), 39
 - Proxy (class in Ldaptor.protocols.ldap.proxy), 39
 - ptrSoaName() (in module Ldaptor.dns), 57
 - put() (in module Ldaptor.ldifree), 60
 - PwgenException, 58
- Q**
- queue() (Ldaptor.protocols.ldap.ldapserver.BaseLDAPServer method), 34
- R**
- ReadOnlyInMemoryLDAPEntry (class in Ldaptor.inmemory), 59

ReadPassword (class in Ldaptor.generate_password), 58
 referral (Ldaptor.protocols.pureldap.LDAPBindResponse attribute), 44
 RelativeDistinguishedName (class in Ldaptor.protocols.ldap.distinguishedname), 27
 remove() (Ldaptor.protocols.ldap.Ldapsyntax.JournaledLDAPAttributeSet attribute), 31
 remove() (Ldaptor.protocols.ldap.Ldapsyntax.JournaledLDAPAttributeSet method), 35
 Replace (class in Ldaptor.delta), 57
 replace() (Ldaptor.compat.unicode method), 53
 requestAvatarId() (Ldaptor.checkers.LDAPBindingChecker method), 51
 requestName (Ldaptor.protocols.pureldap.LDAPExtendedRequest attribute), 44
 requestValue (Ldaptor.protocols.pureldap.LDAPExtendedRequest attribute), 44
 response (Ldaptor.protocols.pureldap.LDAPExtendedResponse attribute), 45
 responseName (Ldaptor.protocols.pureldap.LDAPExtendedResponse attribute), 45
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPAdminLimitExceeded attribute), 29
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPEffectsMultipleDSAs attribute), 29
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPAliasDerivationProblem attribute), 29
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPAliasProblem attribute), 29
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPAttributeOrValueExists attribute), 29
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPAuthMethodNotSupported attribute), 29
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPBusy attribute), 29
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPCompareFailed attribute), 30
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPCompareResult attribute), 30
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPConfidentialityRequired attribute), 30
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPConstraintViolation attribute), 30
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPEntryAlreadyExists attribute), 30
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPInappropriateAttribute attribute), 30
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPInappropriateAttributeForModification attribute), 30
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPInsufficientRights attribute), 30
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPInvalidArgument attribute), 30
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPInvalidChange operation attribute), 31
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPInvalidDNSyntax attribute), 31
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPLoopDetect attribute), 31
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPNamingViolation attribute), 31
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPNoSuchAttribute attribute), 31
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPNoSuchObject attribute), 31
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPNotAllowedOnNonLeaf attribute), 31
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPNotAllowedOnRDN attribute), 31
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPObjectClassModsProhibited attribute), 31
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPObjectClassViolation attribute), 32
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPOperationsError attribute), 32
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPOther attribute), 32
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPProtocolError attribute), 32
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPReferral attribute), 32
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPResult attribute), 32
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPSaslBindInProgress attribute), 32
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPSizeLimitExceeded attribute), 32
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPStrongAuthRequired attribute), 32
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPTimeLimitExceeded attribute), 32
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPUnavailable attribute), 33
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPUnavailableCriticalExtension attribute), 33
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPUndefinedAttributeType attribute), 33
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPUnknownError attribute), 33
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.LDAPUnwillingToPerform attribute), 33
 resultCode (Ldaptor.protocols.ldap.Ldaperrors.Success attribute), 33
 resultCode (Ldaptor.protocols.pureldap.LDAPBindResponse attribute), 44
 rindex() (Ldaptor.compat.unicode method), 53
 reset() (Ldaptor.compat.unicode method), 53
 rpartition() (Ldaptor.compat.unicode method), 53

rsplit() (Ldaptor.compat.unicode method), 53
 rstrip() (Ldaptor.compat.unicode method), 53

S

scope (Ldaptor.protocols.pureldap.LDAPSearchRequest attribute), 49
 search() (Ldaptor.entryhelpers.SearchByTreeWalkingMixin method), 58
 search() (Ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithClient method), 35
 SearchByTreeWalkingMixin (class in Ldaptor.entryhelpers), 58
 send() (Ldaptor.protocols.ldap.ldapclient.LDAPClient method), 28
 send() (Ldaptor.testutil.LDAPClientTestDriver method), 63
 send_multiResponse() (Ldaptor.protocols.ldap.ldapclient.LDAPClient method), 28
 send_multiResponse() (Ldaptor.testutil.LDAPClientTestDriver method), 63
 send_noResponse() (Ldaptor.protocols.ldap.ldapclient.LDAPClient method), 28
 send_noResponse() (Ldaptor.testutil.LDAPClientTestDriver method), 63
 serverSaslCreds (Ldaptor.protocols.pureldap.LDAPBindResponse attribute), 44
 ServiceBindingProxy (class in Ldaptor.protocols.ldap.svcbindproxy), 39
 services (Ldaptor.protocols.ldap.svcbindproxy.ServiceBindingProxy attribute), 39
 setPassword() (Ldaptor.entry.EditableLDAPEntry method), 58
 setPassword() (Ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithClient method), 35
 setPassword_ExtendedOperation() (Ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithClient method), 36
 setPassword_Samba() (Ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithClient method), 36
 setPasswordMaybe_ExtendedOperation() (Ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithClient method), 35
 setPasswordMaybe_Samba() (Ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithClient method), 36
 sizeLimit (Ldaptor.protocols.pureldap.LDAPSearchRequest attribute), 49
 split() (Ldaptor.compat.unicode method), 53
 split() (Ldaptor.protocols.ldap.distinguishedname.DistinguishedName method), 27
 split() (Ldaptor.protocols.ldap.distinguishedname.RelativeDistinguishedName method), 27
 splitlines() (Ldaptor.compat.unicode method), 53
 sshadigest() (in module Ldaptor.entry), 58
 start() (Ldaptor.protocols.ldap.autofill.posixAccount.Autofill_posix method), 26
 start() (Ldaptor.protocols.ldap.autofill.sambaAccount.Autofill_samba method), 26
 start() (Ldaptor.protocols.ldap.autofill.sambaSamAccount.Autofill_samba method), 26
 startGuessing() (Ldaptor.numberalloc.freeNumberGuesser method), 61
 startswith() (Ldaptor.compat.unicode method), 54
 startTLS() (Ldaptor.protocols.ldap.ldapclient.LDAPClient method), 28
 state_HEADER() (Ldaptor.protocols.ldap.ldifprotocol.LDIF method), 38
 state_IN_ADD_ENTRY() (Ldaptor.protocols.ldap.ldifdelta.LDIFDelta method), 37
 state_IN_DELETE() (Ldaptor.protocols.ldap.ldifdelta.LDIFDelta method), 37
 state_IN_ENTRY() (Ldaptor.protocols.ldap.ldifprotocol.LDIF method), 38
 state_IN_MOD_SPEC() (Ldaptor.protocols.ldap.ldifdelta.LDIFDelta method), 37
 state_WAIT_FOR_CHANGETYPE() (Ldaptor.protocols.ldap.ldifdelta.LDIFDelta method), 37
 state_WAIT_FOR_DN() (Ldaptor.protocols.ldap.ldifdelta.LDIFDelta method), 37
 state_WAIT_FOR_DN() (Ldaptor.protocols.ldap.ldifprotocol.LDIF method), 38
 state_WAIT_FOR_MOD_SPEC() (Ldaptor.protocols.ldap.ldifdelta.LDIFDelta method), 37
 StoreParsedLDIF (class in Ldaptor.ldiftree), 60
 str_to_basctii() (in module Ldaptor.compat), 54
 str_to_uasctii() (in module Ldaptor.compat), 54
 strip() (Ldaptor.compat.unicode method), 54
 subtree() (Ldaptor.entryhelpers.SubtreeFromChildrenMixin method), 58
 SubtreeFromChildrenMixin (class in Ldaptor.entryhelpers), 58
 Success (class in Ldaptor.protocols.ldap.ldaperrors), 33
 swapcase() (Ldaptor.compat.unicode method), 54

`symmetric_difference()` (Ldaptor.attributeset.LDAPAttributeSet method), 51

`SyntaxDescription` (class in Ldaptor.schema), 62

T

tag (Ldaptor.protocols.pureber.BERBase attribute), 40

tag (Ldaptor.protocols.pureber.BERBoolean attribute), 40

tag (Ldaptor.protocols.pureber.BEREnumerated attribute), 40

tag (Ldaptor.protocols.pureber.BERInteger attribute), 40

tag (Ldaptor.protocols.pureber.BERNull attribute), 40

tag (Ldaptor.protocols.pureber.BEROctetString attribute), 40

tag (Ldaptor.protocols.pureber.BERSequence attribute), 41

tag (Ldaptor.protocols.pureber.BERSet attribute), 41

tag (Ldaptor.protocols.pureldap.LDAPAbandonRequest attribute), 41

tag (Ldaptor.protocols.pureldap.LDAPAddRequest attribute), 41

tag (Ldaptor.protocols.pureldap.LDAPAddResponse attribute), 42

tag (Ldaptor.protocols.pureldap.LDAPBindRequest attribute), 43

tag (Ldaptor.protocols.pureldap.LDAPBindResponse attribute), 44

tag (Ldaptor.protocols.pureldap.LDAPBindResponse_serverSaslCreds attribute), 44

tag (Ldaptor.protocols.pureldap.LDAPControls attribute), 44

tag (Ldaptor.protocols.pureldap.LDAPDelRequest attribute), 44

tag (Ldaptor.protocols.pureldap.LDAPDelResponse attribute), 44

tag (Ldaptor.protocols.pureldap.LDAPExtendedRequest attribute), 44

tag (Ldaptor.protocols.pureldap.LDAPExtendedResponse attribute), 45

tag (Ldaptor.protocols.pureldap.LDAPFilter_and attribute), 45

tag (Ldaptor.protocols.pureldap.LDAPFilter_approxMatch attribute), 45

tag (Ldaptor.protocols.pureldap.LDAPFilter_equalityMatch attribute), 45

tag (Ldaptor.protocols.pureldap.LDAPFilter_extensibleMatch attribute), 45

tag (Ldaptor.protocols.pureldap.LDAPFilter_greaterOrEqual attribute), 45

tag (Ldaptor.protocols.pureldap.LDAPFilter_lessOrEqual attribute), 45

tag (Ldaptor.protocols.pureldap.LDAPFilter_not attribute), 46

tag (Ldaptor.protocols.pureldap.LDAPFilter_or attribute), 46

tag (Ldaptor.protocols.pureldap.LDAPFilter_present attribute), 46

tag (Ldaptor.protocols.pureldap.LDAPFilter_substrings attribute), 46

tag (Ldaptor.protocols.pureldap.LDAPFilter_substrings_any attribute), 46

tag (Ldaptor.protocols.pureldap.LDAPFilter_substrings_final attribute), 46

tag (Ldaptor.protocols.pureldap.LDAPFilter_substrings_initial attribute), 46

tag (Ldaptor.protocols.pureldap.LDAPMatchingRuleAssertion_dnAttributes attribute), 47

tag (Ldaptor.protocols.pureldap.LDAPMatchingRuleAssertion_matchingRule attribute), 47

tag (Ldaptor.protocols.pureldap.LDAPMatchingRuleAssertion_matchValue attribute), 47

tag (Ldaptor.protocols.pureldap.LDAPMatchingRuleAssertion_type attribute), 47

tag (Ldaptor.protocols.pureldap.LDAPModifyDNRequest attribute), 47

tag (Ldaptor.protocols.pureldap.LDAPModifyDNResponse attribute), 48

tag (Ldaptor.protocols.pureldap.LDAPModifyDNResponse_newSuperior attribute), 48

tag (Ldaptor.protocols.pureldap.LDAPModifyRequest attribute), 48

tag (Ldaptor.protocols.pureldap.LDAPModifyResponse attribute), 48

tag (Ldaptor.protocols.pureldap.LDAPPasswordModifyRequest_newPassword attribute), 48

tag (Ldaptor.protocols.pureldap.LDAPPasswordModifyRequest_oldPassword attribute), 48

tag (Ldaptor.protocols.pureldap.LDAPPasswordModifyRequest_userIdentity attribute), 48

tag (Ldaptor.protocols.pureldap.LDAPReferral attribute), 49

tag (Ldaptor.protocols.pureldap.LDAPResponse attribute), 49

tag (Ldaptor.protocols.pureldap.LDAPResponseName attribute), 49

tag (Ldaptor.protocols.pureldap.LDAPSearchRequest attribute), 49

tag (Ldaptor.protocols.pureldap.LDAPSearchResultDone attribute), 49

tag (Ldaptor.protocols.pureldap.LDAPSearchResultEntry attribute), 50

tag (Ldaptor.protocols.pureldap.LDAPSearchResultReference attribute), 50

tag (Ldaptor.protocols.pureldap.LDAPUnbindRequest attribute), 50

timeLimit (Ldaptor.protocols.pureldap.LDAPSearchRequest attribute), 49

timestamp() (ldaptor.protocols.ldap.svcbindproxy.ServiceBindingProxy method), 39

title() (ldaptor.compat.unicode method), 54

translate() (ldaptor.compat.unicode method), 54

type (ldaptor.protocols.pureldap.LDAPMatchingRuleAssertion attribute), 47

typesOnly (ldaptor.protocols.pureldap.LDAPSearchRequest attribute), 49

U

u() (in module ldaptor.compat), 51

uascii_to_str() (in module ldaptor.compat), 54

unbind() (ldaptor.protocols.ldap.ldapclient.LDAPClient method), 28

unbind() (ldaptor.testutil.LDAPClientTestDriver method), 63

unbound (ldaptor.protocols.ldap.proxy.Proxy attribute), 39

undo() (ldaptor.entry.EditableLDAPEntry method), 58

undo() (ldaptor.protocols.ldap.ldapsyntax.LDAPEntryWithClient method), 36

unescape() (in module ldaptor.protocols.ldap.distinguishedname), 27

unicode (class in ldaptor.compat), 51

union() (ldaptor.attributeset.LDAPAttributeSet method), 51

UnknownBERTag, 41

unsolicitedNotification() (ldaptor.protocols.ldap.ldapclient.LDAPClient method), 28

unsolicitedNotification() (ldaptor.protocols.ldap.ldapservice.BaseLDAPServer method), 34

up() (ldaptor.protocols.ldap.distinguishedname.DistinguishedName method), 27

update() (ldaptor.protocols.ldap.ldapsyntax.JournaledLDAPAttributeSet method), 35

upper() (ldaptor.compat.unicode method), 54

useLMhash() (in module ldaptor.config), 56

V

value (ldaptor.protocols.ldap.distinguishedname.LDAPAttributeTypeAndValue attribute), 27

value (ldaptor.protocols.pureber.BERInteger attribute), 40

value (ldaptor.protocols.pureber.BEROctetString attribute), 40

value (ldaptor.protocols.pureldap.LDAPMessage attribute), 47

version (ldaptor.protocols.ldap.ldifprotocol.LDIF attribute), 38

W

waitingConnect (ldaptor.protocols.ldap.proxy.Proxy attribute), 39