Open Source Open Training

LDAP Basics

Andrew Findlay
Skills 1st Ltd

November 2015

Dr Andrew Findlay
Skills 1st Ltd
2 Cedar Chase
Taplow
Maidenhead
SL6 0EU
01628 782565
andrew.findlay@skills-1st.co.uk

Open Source Open Training

Course Overview

- Directory service basics
- LDAP data model
- LDAP service model
- Authentication with LDAP

This version of the course matches exercises version 0.2.

Open Source Open Training

1: Directories and LDAP

Open Source Open Training

Directories everywhere

- You look things up in them...
- The Phone Book (White Pages)
- Yellow Pages
- Business Directories
- DNS
- Google? No: unstructured data

Open Source Open Training

Directory services need

- Standard network protocol
- Highly structured data
- Very fast search and read
- Ability to distribute data
- Ability to cache data
- Ability to replicate data

Having a standard *protocol* rather than a standard *API* or *ABI* means that clients and servers can come from different suppliers and still work together.

Structured data is necessary to support non-human data users.

Some applications generate a lot of directory operations per second, so we need to replicate and cache for performance.

Distributing data allows easier delegation of data management, and thus better data quality.

Open Source Open Training

Directory service standards

- ECMA TR32 (1985)
- X.500: ISO/CCITT standard 1988/1993...
  - Data model
  - Directory Access Protocol
  - Directory System Protocol
  - X.509 Certificates for strong authentication
- LDAP: Internet RFCs 1993 onwards
  - RFC4510 (2006) is current master doc
  - 60+ relevant RFCs

The X.500/ISO9594 standards were jointly developed by ISO and the CCITT (now known as ITU-T). LDAP standards reference the 1993 edition of the X.500 documents, so you should have copies of those for reference. As the 1993 version has now been superseded, it is available without charge from ITU:
http://www.itu.int/itu-t/recommendations/index.aspx?ser=X

A handy collection of LDAP-related RFCs is distributed with the OpenLDAP source code, in the doc/rfc directory.

Open Source Open Training

LDAP Overview

- Lightweight Directory Access Protocol
- Based on X.500 / ISO9594
- Read-mostly datastore
- Replication, distributed data
- Standard *protocol* rather than *API*
- Tree of data - the *DIT*
- Attribute-value in nodes

LDAP was originally designed purely as an access protocol for X.500-based directory systems. It is now mainly used with LDAP-only servers.

Standards define the protocol. There are several APIs (Application Programming Interfaces) defined by software developers. Some of these – particularly the ones that try to use SQL – are truly dreadful, but they can all make basic queries against any LDAP server.

Open Source Open Training

Data model 1: entries

- An entry represents a person, organisation, room, printer...
- Attribute-value data:

  commonName: Dr A J Findlay

  commonName: Andrew Findlay

  surname: Findlay

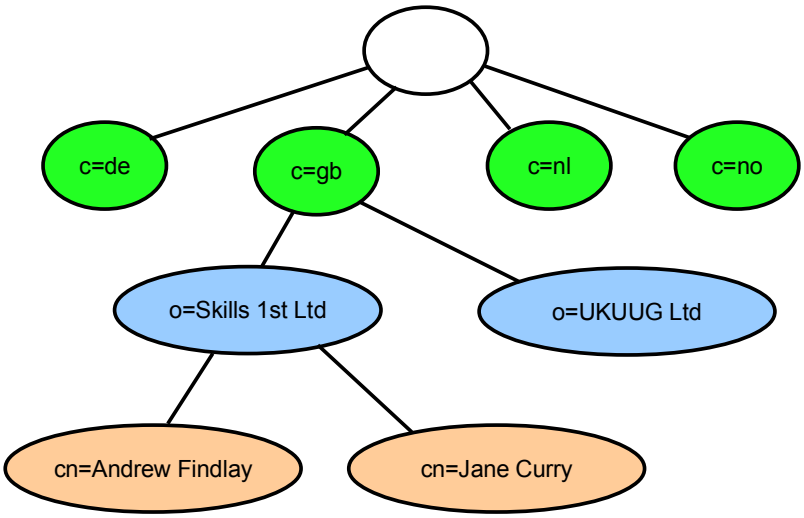
  mail: andrew.findlay@skills-1st.co.uk

  telephoneNumber: +44 1628 782565

Many attributes are allowed to have multiple values. These are **sets**, not **sequences** so do not depend on the order of the values being preserved.

Open Source Open Training

Data model 2: the DIT



The root node does not have a name, and in the CCITT version of the standard it is a purely virtual entity

The naming scheme shown here is the one used in the standards and in most books on directory services, but is not recommended for real-world systems.

Open Source Open Training

Entry names

- Select one attribute-value pair
  - This becomes the RDN
- The full DN is the catenation of all entry names on the path up to the root
  - cn=J Smith,o=Big PLC,c=GB
- Potential for clashes
- Multi-valued RDNs are permitted
  - cn=J Smith+uid=js763,o=Big PLC,c=GB

RDN is *Relative Distinguished Name*
DN is *Distinguished Name*
Very like relative and absolute filenames in Unix, but the root is at the right-hand-end and is not marked explicitly. In practice you can assume that any DN with two or more components is absolute (there is no concept of a 'current position' in LDAP). The protocol almost always uses the absolute DN form.
Although multi-valued RDNs *can* be used to resolve name clashes, they still do not solve the problem of people who change their name. It is better to an opaque ID as the naming attribute.

Open Source Open Training

Simple Search

- Specify:
  - A subtree to be searched
    `o=Skills 1st,c=gb`
  - A filter to match entries of interest
    `sn=Findlay`
    `cn=Andrew*`
- Get back:
  - Zero or more entries
  - Status

There are lots more options to the search operation, which we will cover later. This is enough to do something useful.

Open Source Open Training

Exercise 1

- Login and explore
- Create LDAP Server
- Simple searches

Open Source Open Training

2: LDAP Data Definitions

Open Source Open Training

Acronyms

- DSA – Directory System Agent
  - LDAP Server
- DUA – Directory User Agent
  - LDAP client library
- DIT – Directory Information Tree
- DN – Distinguished Name

Open Source Open Training

Schema and other difficult words

- Attribute Type
- Syntax
- Matching Rule
- Object Class
- Inheritance
- OID

Open Source Open Training

Inheritance

- X.500 and LDAP are object-oriented
- Things defined as 'like this, but with these extras'
- Inheritance indicated in schema by 'SUP' (superior)

Inheritance applies to attribute types and object classes. There is a hierarchy of each.

Open Source Open Training

Attribute types

- Names used to describe a type of data
  - cn, sn, mail, telephoneNumber ...
- Attribute definition includes:
  - name
  - OID
  - syntax
  - permitted matching rules
  - single-value flag

Open Source Open Training

Attribute definition

- Varies from one server to another

```
attributetype ( 0.9.2342.19200300.100.1.5
  NAME ( 'drink' 'favouriteDrink' )
  DESC 'RFC1274: favorite drink'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256}
)
attributetype ( 2.5.4.3
  NAME ('cn' 'commonName')
  DESC 'RFC2256: name of the entity'
  SUP name
)
```

The syntax shown here matches that used in the RFCs. Some LDAP servers require schema to be specified differently.

LDAP schemas often give multiple synonyms for attributes. This is a hang-over from X.500 where it worked because only OIDs were used on the wire. LDAP servers and clients do not always match up the synonyms, which can cause confusion. Always use the shortest name.

Note the suggested length limit on the syntax of favouriteDrink. This is a 'minimum maximum' and all it means is that servers and clients must allow the value to be at least that size.

cn inherits most of its characteristics from 'name'

Open Source Open Training

Syntaxes

- Data-types
  - directoryString
  - DN
  - generalizedTime
  - IA5String
  - telephoneNumber
  - postalAddress
- Always referred to by OID
- Text almost always UTF-8

Most text attributes in LDAP allow the full UTF-8 character set. A few have a more restrictive definition, e.g. e-mail addresses must be in IA5 (ASCII)

Syntaxes must be compiled into the clients and servers so it is normally not practical to add new ones.

Open Source Open Training

Matching Rules

- Operations to be used in searches
  - caseExactMatch
  - caseIgnoreMatch
  - caseIgnoreSubstringsMatch
  - caseIgnoreOrderingMatch
  - telephoneNumberMatch
  - Many more...
- Beware! Not all implemented!

DSAs will only permit searches if the definition of the attribute being matched includes the relevant matching rule. This can be really annoying, as some obvious combinations are missing – probably due to an oversight by the author of the original schema. This can sometimes be worked around in the client by using a special search form that includes the OID of the matching rule to be used (see RFC4515 for details).

Worse still, some DSAs implement a very limited subset of the matching rules. In one design job I had to work around a big-name server that would not do greater-than/less-than comparisons on integers.

Open Source Open Training

Object classes

- Define the *type* of the entry
- List permitted and required attributes
- Three types:
  - Structural
  - Auxiliary
  - Abstract
- Inheritance is supported

Once an entry has been created, its **structural** object class cannot be changed.
Auxiliary classes can be added and removed at any time.
Abstract classes are very rare: you will never need to declare one.

Open Source Open Training

Objectclass definitions

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

The set of attributes used in a class can seem very arbitrary. The early (X.500 1988) classes were defined before there was much practical experience of directory services, and many of the others “grew like topsy” during the PARADISE project.

However silly they may seem, **do not modify the standard definitions** – you can easily add your own classes, and there may be hidden assumptions in server and client code that would break if you changed something they were expecting to see.

Open Source Open Training

Objectclass rules

- STRUCTURAL class of entry cannot be changed after creation
- Entry cannot inherit from two *different* structural classes
  - *person, organizationalPerson, inetOrgPerson* is OK
  - *inetOrgPerson, pilotPerson* is not

person, *organizationalPerson*, and *inetOrgPerson* are a single inheritance chain, so an *inetOrgPerson* object is also a *person* object by definition. These are all structural classes.

pilotPerson is also a structural class but it inherits directly from *top* so it clashes with the other classes.

This rule and the division into STRUCTURAL, ABSTRACT, and AUXILIARY classes came in from X.500(1993) so many older schema definitions were caught by it. Early LDAP servers did not enforce any schema rules but later ones do, so people still run into problems when they upgrade their software.

Open Source Open Training

OIDs

- Object Identifier – a unique “name”
- X.500 uses these in protocol
- LDAP *prefers* human-readable names
- 0.9.2342.19200300.100.1.5
- Infinitely extendable
- Various registries and allocation rules
  - 1.2.826.0.1.<UK company number>

It is fairly easy to get an allocation under the “enterprises” arc from IANA.

Easier still in some countries as standards like the UK's BS7453 effectively pre-allocate OIDs to registered companies. (The example given above is for companies registered in England or Wales)

For organizations registered as a company under the provisions of the Companies Act 1985, section 6.1.1 of BS7453 states that the company "is deemed to be assigned" an object identifier that starts with 1 for England and Wales, 2 for Scotland, and 3 for Northern Ireland, and has as its next component "the integer value of the numeric component of the registered number".

Once you have an OID you can assign what you like beneath it.

For LDAP purposes the main thing is to make sure that the OIDs you use are unique. Their structure does not have any inherent meaning.

Harald Alvestrand maintains a useful registry of OIDs:

http://www.alvestrand.no/objectid/

Open Source Open Training

Data model summary

- Tree of entries – the DIT
- Attribute-value data in entries
- Schema rules define what can/must be present

Open Source Open Training

Exercise 2

- Install GUI browser
- Browse DIT
- Simple searches
- Browse schema

Open Source Open Training

3: LDAP Operations

Open Source Open Training

LDAP operations

- Bind
- Search
- Add
- Delete
- Modify
- Compare
- Abandon
- Extended

There is no “read” operation – this function is included in Search
There are a few others, such as “Modify DN” which are not used very much – see RFC4511 for the full details

Controls can be applied to most operations to change their behaviour. A particularly useful one allows the operation to be performed *as if bound as some other user*. This is governed by server policy, and can be useful with trusted portal applications.

Open Source Open Training

Bind

- Authenticates to the server
- “Simple”: DN and password
- SASL
  - userID and credentials
    - Kerberos
    - DIGEST_MD5
  - external (e.g. client certificate)

LDAPv3 allows anonymous operations without Bind.
LDAP 'usernames' are DNs. They often correspond to entries in the DIT, but they do not have to.
One username that usually does not have a corresponding entry is known as the *rootdn* and it has powers similar to the Unix *root* account.

Open Source Open Training

Search

- Base – specifies starting point in DIT
- Scope – how far to look
  - base object, single level, subtree
- Filter – what to look for
- Attribute list – what to return
- Options – limits on size, time etc

Open Source Open Training

Search filter examples

- (sn=Smith)
- (cn=Andrew*)
- (cn=and*w*fi*y)
- (objectClass=*)
- (&(objectclass=acct)(uid=zb42))
- (&(objectclass=person)(|(cn=*fred*)(sn=*fred*)(drink=*fred*)))

Don't OR too many things together! If you include just one that the DSA does not index, the performance drops badly and the DSA may refuse to perform the search.

Search filters are supposed to be enclosed in parentheses. Most implementations cope if you omit them, but it is better to use the correct syntax..

Open Source Open Training

Search results

- Zero or more entry names
- Possibly some attributes for each entry
- Status code
  - Success
  - Various failures
  - Size limit exceeded
  - Admin limit exceeded
- Operational attributes can be requested

Finding zero entries is not an error!

If no attributes are requested, search will return all the “user” attributes that the requester is allowed to see.

Operational attributes include things like creation date, last modified date, last modified by, etc... You must ask for these explicitly if you want them.

Directories do not often return “permission denied” codes – they are more likely to behave as if the data you asked for simply does not exist.

It is possible to get some entries as well as an error code. If you exceed a limit then it is up to the server to decide whether to return anything at all.

Open Source Open Training

Modify

- Add/delete/change attribute-value pairs
- Accepts a list of changes
- The only atomic operation
- Modify/replace whole attributes or specified values
  - To specify values there must be a matching rule for the attribute

Beware of reading entries and then sending back entire modified attributes – if the attribute has 100,000 values this could take a long time! (e.g. big groups)

Open Source Open Training

Add

- Add one directory entry
- Entry must conform to schema
- Parent entry must exist
  (unless adding a suffix entry)
- Bulk adds usually start from LDIF file

Open Source Open Training

LDIF

- LDAP Data Interchange Format
- RFC2849
- Transfer complete entries / subtrees
- Specify attribute-level modifications
- Delete entries
- Portable format
  - backup
  - data transfer between DSAs

```
dn: uid=ac000000,dc=example,dc=org
objectclass: inetOrgPerson
cn: Jason Medland
sn: Medland
mail: ac000000@example.org
telephoneNumber: +44 1234 567000

dn: uid=bl2345,dc=example,dc=org
changetype: delete

dn: uid=ma736353,dc=example,dc=org
changetype: modify
add: postaladdress
postaladdress: 25 The Street $ The Town $ Linuxshire
-
replace: mail
mail: ma5@example.org
-
```

Open Source Open Training

LDIF Example

```
dn: dc=people,dc=example,dc=org
objectclass: organizationalUnit
objectclass: dcObject
ou: People
dc: people

dn: uid=qr00042,dc=people,dc=example,dc=org
objectclass: inetOrgPerson
objectclass: person
cn: Fiona Pinnington
sn: Pinnington
uid: qr00042
mail: qr00042@example.org
telephoneNumber: +44 1234 567000
userPassword: secret
```

Where an LDIF file defines objects at several levels in the tree it is important to place the higher-level ones before objects that are logically 'below' them.

Open Source Open Training

Command-line tools

- One tool for each LDAP operation:
  - ldapsearch
  - ldapadd
  - ldapmodify
  - ldapdelete
  - ldappasswd
- All can bind as specified ID

Open Source Open Training

Command-line examples

```
ldapsearch -x -b dc=example,dc=org \
      sn=smith

ldapsearch -x \
  -D cn=root,dc=example,dc=org \
  -w secret \
  -b dc=example,dc=org \
  -s sub \
  '(&(sn=smith)(mail=*@example.org))'

ldapadd -x \
  -D cn=root,dc=example,dc=org \
  -y file-with-password \
  -f data.ldif
```

Most command-line option flags are common to LDAP tools from all suppliers, but some of the more recent flags vary as they were introduced after the various code-bases forked from the original University of Michigan code.

Open Source Open Training

Exercise 3

- Load data from LDIF
- Modify data from GUI

Open Source Open Training

4: Authentication and Authorisation

Open Source Open Training

Authentication using LDAP

- Normal process:
  - Bind anonymously or with fixed ID
  - Search for user entry (uid=username)
  - Bind as that entry with supplied password
- Alternative:
  - Bind directly using SASL

Note that in both cases the supplied password will usually be sent to the LDAP server. This transaction should be protected cryptographically.

Open Source Open Training

Authorisation using LDAP

- Authorisation normally expressed as group membership
- LDAP group is an entry
- Members represented by DN values of *member* attribute

```
dn: cn=Web Editors,ou=groups,dc=example,dc=org
objectclass: groupOfNames
cn: Web Editors
member: uid=qr0042,dc=people,dc=example,dc=org
member: uid=xa0003,dc=people,dc=example,dc=org
```

An application will get the user's DN during the authorisation process. It can easily check whether that DN is listed as a member of a particular group before giving access to controlled facilities. This can be done using either the *search* or *compare* operations. If *search* is used, take care not to request the return of the *member* attribute as it may have many thousands of values.

Open Source Open Training

POSIX passwd data in LDAP

- RFC2307

```
ajf:x:1234:1234:Andrew Findlay:/home/ajf:/bin/bash
```

```
objectclass: inetOrgPerson
objectclass: posixAccount
cn: Andrew Findlay
sn: Findlay
uid: ajf
uidNumber: 1234
gidNumber: 1234
homeDirectory: /home/ajf
gecos: Andrew Findlay
userPassword: {SSHA}MCbiTYMHrt6GSReXxZ6dHzNviiUEE/xR
```

The original RFC2307 has some flaws, but is widely supported.
Newer systems may conform to 'rfc2307bis' which can be found in the OpenLDAP source distribution as `draft-howard-rfc2307bis-xx.txt`
There are still problems with this schema: it mixes account management and password management and it is not completely POSIX compliant. Efforts to improve it are ongoing. There is also a competitor called DBIS that appeared in 2014.

Open Source Open Training

POSIX group data in LDAP

- RFC2307

```
objectClass: posixGroup
cn: dialout
gidNumber: 16
memberUid: ajf
memberUid: bjc
memberUid: mtr
```

Important: existing LDAP data conforming to RFC2307 may not load into a directory server conforming to RFC2307bis without modification. The definitions of some attribute types and object classes have changed in incompatible ways. For example, *posixGroup* is now an auxiliary class rather than a structural one.

Note that *memberUid* is case-sensitive, where *uid* itself is not.

Open Source Open Training

Exercise 4

- Simple authentication
- Groups using DNs
- Using RFC2307
  - Passwd data
  - Groups using UIDs

Open Source Open Training

More LDAP Topics

- TLS
- Replication
- Distributed DIT
- DIT Design
- Access Control
- Client-side programming

Open Source Open Training

LDAP Basics

Andrew Findlay
Skills 1st Ltd

November 2015
andrew.findlay@skills-1st.co.uk

Open Source Open Training

Creative Commons

This course was originally written by Andrew Findlay of Skills 1st Ltd. If you modify it you should add your name here and indicate what you have done.
The roadworks image comes from SVG SILH and is licensed under CC0.