Selected LDAP Attributes

Version 1.3

10 July 2008

Andrew Findlay

Skills 1st Ltd

Synopsis

This document lists attributes that are commonly used in directory entries. The standard definition is given, followed by a commentary on the use and limitations of the attribute.

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

1 Attributes

Most directory data is held in the form of attribute-value pairs. The set of attributes that can appear in a given entry is set by its *objectClass*. Some of the attributes are mandatory, but most are optional. Other attributes can be added by extending the schema if necessary.

Most attributes can have multiple values associated with them. No order of preference is implied where there is more than one value.

Most attributes require values to be expressed in the UTF-8 encoding, though some are more restrictive.

Length limits are defined for some attributes. Note that the usable size may be less if multi-byte characters are in use. Note also that the limits are defined in the standards as *minimum acceptable limit values* – thus an LDAP implementation must allow values of at least the size specified but is not required to limit the size at all.

In the sections that follow, the name of the attribute is given in ***bold italics*** along with an expansion where necessary. The description from the defining standard then appears in *italics*, followed by explanatory notes in normal text.

Note that the precision of attribute definitions in the standards varies widely from one to the next. This is because many of the attributes were 'borrowed' from other postal and telecommunication standards, and many more were created on-the-fly when required in the early pilot implementations of X.500.

2 Name Attributes

Most name attributes are derived from the *name* supertype, which specifies a size limit of 32768 characters for each value and also defines the available matching rules and character set (UTF-8).

Directory searches on name attributes are not case-sensitive, though the case of input data is preserved. This has one awkward side-effect: it is quite hard to correct an entry that is spelled correctly but has incorrect case (e.g. to change sn=Mcleod to sn=McLeod). Searches may specify substring matching, but cannot normally specify greater-than or less-than filters.

2.1 cn (Common Name)

> *This is the X.500 commonName attribute, which contains a name of an object. If the object corresponds to a person, it is typically the person's full name.*

cn is often multi-valued, to aid searches in the directory, e.g.:

```
Brian Stanley Walker
Mr B S Walker
Paddy Walker
```

X.520(1988) contains an example suggesting that Common Names might extend to the very full and formal: 'Mr Robin Lachlan McLeod Bsc(Hons) CEng MIEE' though it is not clear how useful this might be in practice.

cn is mandatory in person entries, and is one of the most important attributes as it is commonly specified in 'white pages' searches.

Note that some information appearing in *cn* will also appear in other attributes, particularly *sn*, *givenName*, *personalTitle*, and *displayName*. CN may hold several variants of a name: this is important to support searching, particularly where names can include characters that are not available on all keyboards, e.g. this Polish name:

```
    cn: Kerstina Krzyža ska
    cn: Kerstina Krzyzanska
```

In this case, the CN attribute holds the “real” name and also a simplified form that can be typed without attention to diacritical marks. The simplified CN makes searching easier, and the real name is available for display in the *displayName*, *sn*, and *givenName* attributes.

2.2 sn (Surname)

> *This is the X.500 surname attribute, which contains the family name of a person.*

sn is mandatory in person entries and can hold multiple values. The standards betray a Western cultural bias here, and I suggest that where a person does not have distinct 'given' and 'family' names, their entire name should be placed in both *cn* and *sn* attributes.

Note that multi-part surnames such as *van der Vorst* should be handled as a single value of *sn*. The use of multiple values may be appropriate where someone has recently changed their name, but this is risky as many applications assume a single value for the *sn* attribute and may behave unpredictably in the presence of multiple values. Name-changes are better handled by adding values to the *cn* attribute.

The value stored in the *sn* attribute should be the “correct” form with all diacritical marks as used by the person themselves.

2.3 givenName

> *The givenName attribute is used to hold the part of a person's name which is not their surname nor middle name.*

A less-commonly used attribute. Provides better normalisation of data than *cn* and costs little to populate so worth using in most cases. As with *sn*, should be the “correct” form including diacritical marks where appropriate.

2.4 initials

> *The initials attribute contains the initials of some or all of an individuals names, but not the surname(s).*

The definition is rather unclear, so if this attribute is to be used it will require a local policy to enforce a common format (use of spaces, dots etc).

2.5 personalTitle

> *The Personal Title attribute type specifies a personal title for a person. Examples of personal titles are "Ms", "Dr", "Prof" and "Rev".*

This comes from RFC1274 and has a length limit of 256 characters. It allows multiple values, but where an individual has multiple titles it may be best to place them all in one value, e.g. 'Eur Ing Dr'.

2.6 generationQualifier

> *The generationQualifier attribute contains the part of the name which typically is the suffix, as in "IIIrd".*

Not commonly used, but might be necessary when dealing with Americans. In most cases it is probably sufficient to incorporate this information in the *cn* and *displayName* attributes.

2.7 displayName

> *When displaying an entry, especially within a one-line summary list, it is useful to be able to identify a name to be used. Since other attribute types such as 'cn' are multivalued, an additional attribute type is needed. Display name is defined for this purpose.*

This is a single-valued attribute. It is strongly recommended that this attribute be filled in with the preferred form of name (which should also appear as a value of *cn*). All appropriate diacritical marks should be included.

Note that where RFC3866 language tags are in use it is possible to have one value per language.

3 Descriptive Attributes

3.1 jpegPhoto

> *Used to store one or more images of a person using the JPEG File Interchange Format [JFIF].*

Size limit 250000 bytes. If this attribute is used, it tends to dominate the size of an entry and can have an impact on performance so it should be kept as small as possible.

3.2 description

> *This attribute contains a human-readable description of the object.*

Size limit 1024 characters.

4 Communication Attributes

4.1 preferredLanguage

> *Used to indicate an individual's preferred written or spoken language. This is useful for international correspondence or human-computer interaction. Values for this attribute type MUST conform to the definition of the Accept-Language header field defined in [RFC2068] with one exception: the sequence "Accept-Language" ":" should be omitted. This is a single valued attribute type.*

The attribute is single-valued, but RFC2068 allows it to contain a list of languages with optional preference values for each. The values follow a convention defined in RFC1766, based on ISO639 "Code for the representation of names of languages" and ISO3166 alpha-2 country codes.

4.2 mail (RFC 822 mail address)

> *RFC 1274 uses the longer name 'rfc822Mailbox' and syntax OID of 0.9.2342.19200300.100.3.5. All recent LDAP documents and most deployed LDAP implementations refer to this attribute as 'mail' and define the IA5 String syntax using using the OID 1.3.6.1.4.1.1466.115.121.1.26*

This is the normal electronic-mail address, which must be given in full RFC822 format (e.g. andrew.findlay@skills-1st.co.uk). Note that the attribute can have multiple values and no order of preference is implied. Note also that in keeping with RFC822 the attribute only allows the IA5 character set. I would advise limiting this to a single value in practice: where an individual has more than one e-mail address it is probably useful to put each in a separate entry so that other attributes can help the user to select the right one to use.

Size limit 256 characters.

4.3 telephoneNumber

> *Telephone numbers are recommended in X.520 to be in international form, as described in E.123 [15].*
>
> *Example: +1 512 305 0280*

Telephone numbers follow the *printableString* syntax so a fairly wide character set is accepted. This allows things like:

```
+44 207 123456 x9876
```

A local convention will be needed if extension numbers are to be permitted.

Telephone numbers should **always** be stored in international form, without any '(0)' insertions or similar. It is the job of the user interface to present the number to the user in the most useful way for that person.

Note that there is also a *homeTelephoneNumber* attribute with similar characteristics.

Size limit 32 characters.

4.4 facsimileTelephoneNumber

Similar to *telephoneNumber* above, but with added optional parameters to describe the capabilities of the fax machine.

4.5 mobile

> *The Mobile Telephone Number attribute type specifies a mobile telephone number associated with a person. Attribute values should follow the agreed format for international telephone numbers: i.e., "+44 71 123 4567".*

No size limit is given in the standards, but it would be reasonable to assume 32 characters in line with *telephoneNumber* above.

4.6 pager

> *The Pager Telephone Number attribute type specifies a pager telephone number for an object. Attribute values should follow the agreed format for international telephone numbers: i.e., "+44 71 123 4567".*

No size limit is given in the standards, but it would be reasonable to assume 32 characters in line with *telephoneNumber* above.

4.7 postalAddress

> *(From X.520(1988)) ... address information required for the physical delivery of postal messages ...*
>
> *... limited to 6 lines of 30 characters each, including a Postal Country Name. Normally the information contained in such an address could include an addressee's name, street address, city, state or province, postal code, and possibly a Post Office Box number depending on the requirements of the named object.*

The implication here is that this attribute should fit onto a small address label or be usable directly on a letter to be posted in a window-envelope. It might be a tall order to fit everything into the space allowed.

Most of the information that could go into *postalAddress* also shows up in individual attributes with more relaxed size limits. It may be possible to construct usable postal addresses from these other attributes, though human intervention will often be required.

This attribute may have multiple values, but it is strongly suggested that only one be provided. In most cases it is better to use *postalAddress* rather than the individual component attributes. There is a similar attribute called *homePostalAddress*.

For a UK-based directory, the *postalAddress* should be directly usable on letters to be posted in the UK, and should preferably be usable over the whole world if possible. Thus, if addressing someone in Greece, the early part of the address should be in Greek script, with only the country given in Roman characters.

4.8 c (Country Name)

> *This attribute contains a two-letter ISO 3166 country code*

This attribute may only carry a single value. Note that there is also a *friendlyCountryName* attribute which can be used to hold one or more human-readable country names. In most cases it is better to use the two-letter ISO3166 code in the *c* attribute and leave 'user friendly' presentation to the user-interface.

4.9 l (Locality Name)

> *This attribute contains the name of a locality, such as a city, county or other geographic region.*

This attribute is derived from the *name* superclass so it inherits the length limit of 32768 characters.

4.10 st (State or Province Name)

> *This attribute contains the full name of a state or province*

This attribute is derived from the *name* superclass so it inherits the length limit of 32768 characters.

4.11 street (Street Address)

> *This attribute contains the physical address of the object to which the entry corresponds, such as an address for package delivery.*
>
> *e.g. 'Arnulfstraße 60'*

Length limit 128 characters.

4.12 postalCode

> *The postalCode attribute specifies the postal code of the named object. If this attribute value is present it will be part of the object's postal address.*

Length limit 40 characters. Odd that this is permitted to be longer than a single line in a postal address!

It is often useful to use this attribute even though the same information may appear in the *postalAddress* attribute. Post codes are useful keys for other services like maps and address databases.

4.13 postOfficeBox

> *... specifies the Post Office Box by which the object will receive physical postal delivery. If present, the attribute value is part of the object's postal address.*

Length limit 40 characters.

4.14 physicalDeliveryOffice

> *The Physical Delivery Office Name attribute specifies the name of the city, village, etc. where a physical delivery office is situated.*

Length limit 128 characters.

4.15 buildingName

> *The Building Name attribute type specifies the name of the building where an organisation or organisational unit is based.*

Length limit 256 characters.

4.16 roomNumber

> *The Room Number attribute type specifies the room number of an object. Note that the commonName attribute should be used for naming room objects.*

Length limit 256 characters.

5 Organisational Attributes

Note that the object class name uses the American spelling: *organization*

5.1 o (Organisation Name)

> *... a string chosen by the organisation (e.g. "Scottish Telecommunications plc") Any variants should be associated with the named organisation as separate and alternative attribute values.*

The use of multiple values is appropriate in entries describing organisations, but may not be as relevant when using an organisation attribute in a person entry.

Length limit 64 characters in X.520. RFC2256 redefines *o* under the *name* superclass, so in LDAP it gets a limit of 32768 characters.

5.2 ou (Organisational Unit Name)

> *The Organisational Unit Name attribute type specifies an organisational unit. When used as a component of a directory name it identifies an organisational unit with which the named object is affiliated.*
>
> *The designated organisational unit is understood to be part of an organisation designated by an Organisation Name attribute. It follows that if an Organisational Unit Name attribute is used in a directory name, it must be associated with an Organisation Name attribute.*
>
> *An attribute value for Organisational Unit Name is a string chosen by the organisation of which it is part (e.g. ou="Technology Division").*

> *Note that the commonly used abbreviation “TD” would be a separate and alternative attribute value.*

Length limit 64 characters in X.520. RFC2256 redefines *ou* under the *name* superclass, so in LDAP it gets a limit of 32768 characters.

5.3 businessCategory

> *This attribute describes the kind of business performed by an organization.*

From X.520. Length limit 128 characters.

5.4 departmentNumber

> *Code for department to which a person belongs. This can also be strictly numeric (e.g., 1234) or alphanumeric (e.g., ABC/123).*

Can be used as a more rigidly-controlled alternative to *ou*. No length limit is specified.

5.5 title

> *The title attribute type specifies the designated position or function of the object within an organisation.*
>
> *Example: title=”Manager, Distributed Applications”*

Length limit is 64 characters in X.520. RFC2256 redefines *title* under the *name* superclass, so in LDAP it gets a limit of 32768 characters.

Note that this is the *job title*. Personal titles like Mrs and Dr come under personalTitle - see section 2.5 above.

5.6 owner

> *The* owner *attribute type specifies the name of some object which has some responsibility for the associated object. The value is a Distinguished Name*

From X.520

This attribute can be useful to trigger Access Control Lists, allowing individuals extra power over entries that they 'own'.

5.7 uniqueIdentifier

> *The Unique Identifier attribute type specifies a "unique identifier" for an object represented in the Directory. The domain within which the identifier is unique, and the exact semantics of the identifier, are for*

> *local definition. For a person, this might be an institution-wide payroll number. For an organisational unit, it might be a department code.*

Length limit 256 characters. See also *employeeNumber* below.

Often useful to provide an opaque naming attribute for an entry.

5.8 organizationalStatus

> *The Organisational Status attribute type specifies a category by which a person is often referred to in an organisation. Examples of usage in academia might include undergraduate student, researcher, lecturer, etc.*
>
> *A Directory administrator should probably consider carefully the distinctions between this and the title and userClass attributes.*

Length limit 256 characters. See also the *employeeType* attribute below.

5.9 employeeNumber

> *Numeric or alphanumeric identifier assigned to a person, typically based on order of hire or association with an organization. Single valued.*

This attribute may only hold a single value. No size limit is specified in the standard.

5.10 employeeType

> *Used to identify the employer to employee relationship. Typical values used will be "Contractor", "Employee", "Intern", "Temp", "External", and "Unknown" but any value may be used.*

Can be multi-valued. No size limit is specified in the standard.

5.11 secretary

> *The Secretary attribute type specifies the secretary of a person. The attribute value for Secretary is a distinguished name.*

This is a pointer to another entry in the directory.

5.12 manager

> *The Manager attribute type specifies the manager of an object represented by an entry.*

This allows organisational hierachies to be modelled in the directory. The value is a distinguished name.

5.13 seeAlso

> *The See Also attribute specifies names of other objects which may be other aspects (in some sense) of the same real world object.*
>
> *An attribute value for See Also is a Distinguished Name.*

This is a pointer to another entry in the directory. It might be used in a personal entry to point to an 'organisational role' entry for example. There is a similar attribute called *roleOccupant* which points back from the role to the actual person. Thus, a railway company might have entries like this in the directory:

```
Entry cn=Fat Controller, o=The Railway, c=sx
cn: Fat Controller
roleOccupant: cn=Sir Topham Hat, o=The Railway, c=sx
title: Director
...

Entry cn=Sir Topham Hat, o=The Railway, c=sx
sn: Hat
cn: Sir Topham Hat
seeAlso: cn=Fat Controller, o=The Railway, c=sx
```

6 Authentication Attributes

LDAP standards define a large set of attributes for describing computer accounts and for the storage of X.509 certificates. Most of these are omitted here, leaving only the basic attributes used in a simple authentication scheme.

Authorisation and access-control attributes are also outside the scope of this section.

6.1 uid (User Identifier)

> *The Userid attribute type specifies a computer system login name.*

Length limit 256 characters. Note that this is not a UID in the Unix sense: it is a username. If Unix account data is to be held in an LDAP format then *uidNumber* should be used for the numeric UID.

6.2 userPassword

> *Passwords are stored using an Octet String syntax and are not encrypted. Transfer of cleartext passwords are strongly discouraged*

> *where the underlying transport service cannot guarantee confidentiality and may result in disclosure of the password to unauthorized parties.*

The definition quoted here comes from RFC2256. However, RFC2307 expands the definition to permit a range of hashed passwords to be stored. Note that hashed passwords cannot be used with all authentication mechanisms so the choice is not straightforward.

This attribute can hold multiple values, each of which is limited to 128 characters.

Access-control is normally set on this attribute to prevent anyone from reading it (including directory managers and the owner of the entry).

7 Other standard attributes

7.1 co (Friendly Country Name)

> *The Friendly Country Name attribute type specifies names of countries in human readable format.*

From RFC1274. This attribute can hold multiple values. No length limit is specified.

This attribute was introduced to make the top level of the DIT more understandable to a person using a simple browser interface. The values are intended for human consumption, so it should always be used with the *c* attribute described in section 4.8 above.

7.2 labeledURI

> *Uniform Resource Locators (URLs) are being widely used to specify the location of Internet resources. There is an urgent need to be able to include URLs in directories that conform to the LDAP and X.500 information models, and a desire to include other types of Uniform Resource Identifiers (URIs) as they are defined.*

From RFC2079. This allows Web references to be attached to any object in the directory, along with a brief description. RFC2079 requires the use of ASCII or the T.61 character set, though in practice it should be possible to use UTF-8 for the label part.

7.3 member

The *member* attribute is used in entries defining groups. It has Distinguished Name syntax, so each value is effectively a pointer to another entry in the directory. This is a multi-valued attribute.