IETF LDAPEXT Working Group
INTERNET-DRAFT

David Watts
Data Connection Ltd.
Steve Orbell
Data Connection Ltd.
April 1998

Efficient Referral Chasing in LDAP Directories
<draft-watts-ldapext-x500-referrals-00.txt>

1. Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as ``work in progress.''

To view the entire list of current Internet-Drafts, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), ftp.nordu.net (Northern Europe), ftp.nis.garr.it (Southern Europe), munnari.oz.au (Pacific Rim), ftp.ietf.org (US East Coast), or ftp.isi.edu (US West Coast).

Distribution of this memo is unlimited. Editorial comments should be sent to the authors. Technical discussion should take place on the IETF IETF LDAP Extension Working Group mailing list <ietf-ldapext@netscape.com>.

2. Abstract

This document defines an extension to the LDAP URL format and a control on a LDAP search operation which, together, permit LDAP clients to chase LDAP referrals in a more efficient and more X.500-like manner.

3. Contents of this Document

The remaining sections in this document are as follows.

- Section 4 gives some background on distributed directories and referrals.

- Section 5 describes the limitations of LDAP referrals and explains why these limitations are unacceptable.

- Sections 6 and 7 formally define the new URL extension and the new search control.

- Sections 8 and 9 give the formal rules governing client and server behaviour with respect to the new extension and control.

- Sections 10 to 12 cover security, references and authors' addresses.

The key words "MUST", "MUST NOT", "SHOULD", and "MAY" used in this document are to be interpreted as described in [BRADNER97].

4. Background - Distributed Directories and Referrals

4.1 Centralised and Distributed Directories

In the LDAP/X.500 model, a directory may be centralised in a single server or distributed over several servers.

4.1.1 Centralised Directories

In the centralised model, the single server performs all operations, returning either results or errors.

There is no requirement to communicate with other servers, because they are outside the scope of the model.

4.1.2 Distributed Directories

In the distributed model, when a server receives an operation which it cannot satisfy (or can only satisfy partially), it can do one of two things.

- It can chain the request to a second server which can satisfy the request. This second server performs the operation and passes the results back to the first server. The first server merges those results with any which have been produced locally or returned from other chained requests, and passes the combined result back to the client.

  The client has no further work to perform.

- It can pass a continuation reference (i.e. a referral) back to the requestor, along with any results which have been produced locally. The client may then use the information in the referral to contact other servers and continue processing the operation.

  This requires the client, not only to contact the other servers, but also to perform server functions such as merging results, discarding any duplicates, referral loop detection, and managing size and time limits.

The latter behaviour is typical of unmanaged directories and of low-end servers which do not implement chaining.

The former behaviour is preferable in managed directories. It allows truly lightweight clients to be deployed and avoids the network and processing overhead of clients connecting to, and authenticating with, multiple servers.

4.2 LDAPv2

LDAPv2 [LDAPv2] defines a simplified version of the X.500 Directory Access Protocol (DAP). It permits access to a directory using a restricted set of ASN.1 encodings, and without requiring an OSI stack.

LDAPv2 thus circumvents some of the overhead associated with DAP and allows more lightweight directory access clients to be deployed.

Like DAP, LDAPv2 defines abstract operations for accessing a directory. LDAPv2 does not define either:

- a chaining mechanism (because LDAP is an access protocol)

- a referral mechanism (for simplicity, because LDAP is lightweight).

4.3 The Requirement for Referrals

Referrals are required in a distributed directory when the servers are incapable of chaining.

Even when servers are capable of chaining, referrals may still be useful. If a server tries to chain to a second server, but the second server is busy or unavailable, the first server should return a referral to the client to indicate that the operation could not be completed.

In the case where the server is capable of chaining, the client will not generally chase the returned referrals, since the server will already have tried to contact the referred-to server. The information in the referral may still be of use, for example to:

- indicate the partial nature of the results to the human user

- chase the referral later (by which time the referred-to server might no longer be unavailable/busy).

4.4 LDAPv3

One of the ways in which LDAPv3 [LDAPv3] extends LDAPv2 is by defining a referral mechanism. Referrals are LDAP URLs, defined in [URL] to contain the following fields. All fields are optional.
- name/address/port of the referred-to server
- DN of the base object of Search (or other operation)
- attributes to be returned from Search
- scope of Search (base-object/one-level/whole-subtree)
- filter for Search
- extensions

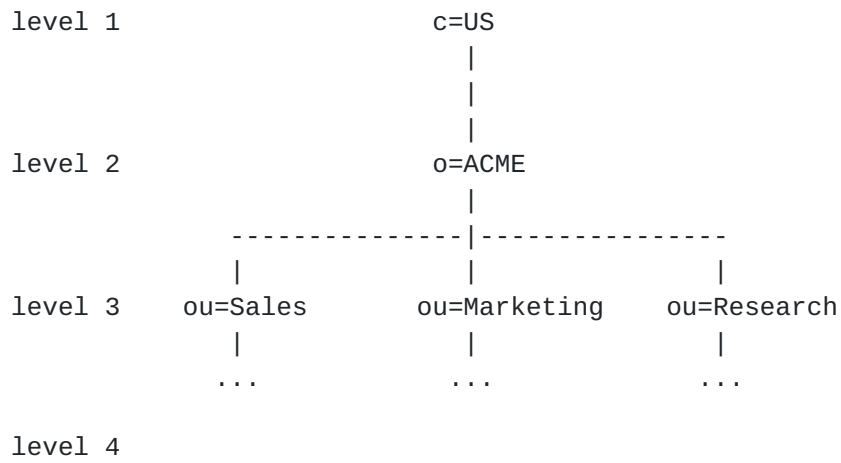
5. Limitations of LDAPv3 Referrals

LDAP URLs as defined in [URL] suffer from the following limitations.
- They can be inefficient.
- They are incompatible with X.500 referrals.

5.1 Inefficient Referrals

A common topology for managing a distributed directory is to divide the naming contexts amongst the cooperating servers in a strictly hierarchical manner.

Example - Distributed Directory for ACME Corp



2 servers hold the directory for ACME Corp:
- server A contains levels 1 and 2 (i.e. US, ACME)
- server B contains level 3 (Sales, Marketing, Research) and subordinate levels

Servers A and B hold knowledge of the sections of the DIT held by each other.

If server A is unable to chain to server B, then a LDAPv3
whole-subtree Search from o=ACME,c=US will return:
- an entry for ACME (if it matches the filter)
- a referral to server B, to search from ou=Sales,o=ACME,c=US
- a referral to server B, to search from ou=Marketing,o=ACME,c=US
- a referral to server B, to search from ou=Research,o=ACME,c=US

Thus when the client chases the referrals to continue the search
in server B, it has to chase 3 different referrals, even though
they are referrals to the same server at sibling nodes.

It would be far more efficient to have a single referral to server B,
with semantic

"search from all the sibling nodes just below o=ACME,c=US"

Chasing three referrals instead of one is not terribly inefficient, but
in real-life directory deployments, the number of subordinates at each
level may be much larger - perhaps several thousand.

5.2 Incompatibility with X.500 referrals

X.500 referrals are defined in X.518 [X518]. In addition to the
information contained in a LDAP URL, they contain a field
nameResolutionPhase.

- nameResolutionPhase 'not completed' indicates that the referral
  contains the DN of entry from which to continue the search.

- nameResolutionPhase 'completed' indicates that the referral contains
  the DN of PARENT of the entry from which to continue the search.

In the ACMECorp example in 5.1, if server A is unable to chain to
server B, then it will return:

- an entry for ACME (if it matches the filter)

- a single referral to server B, to search from all the naming contexts
  which are immediately subordinate to o=ACME,c=US
  (i.e. with nameResolutionPhase = 'completed')

If this X.500 referral is converted to a LDAP referral

ldap://serverb/o=ACME,c=US

and the client converts this to a search operation to be passed
to server B, then the nameResolutionPhase = 'completed' information
will be lost.

Server B will therefore attempt to continue the search from o=ACME,c=US (rather than each of its subordinates). Server B, when it follows the Find DSE procedure in [X518] section 18, will either chain or refer back to server A. This will then lead to a referral loop between servers A and B.

Note that this problem cannot be overcome by 'munging' the X.500 referral when it comes to convert it to LDAP. The information which would be required to produce usable LDAP referrals:

```
ldap://serverb/ou=Sales,o=ACME,c=US
ldap://serverb/ou=Marketing,o=ACME,c=US
ldap://serverb/ou=Research,o=ACME,c=US
```

(i.e. the RDNs) is not contained in the X.500 referral.

This information may not even be available to the server which is converting from the X.500 referral to the LDAP referral.

- The X.500 referral may have been produced by a different server.

- The knowledge of server B may be contained in an NSSR, rather than in 3 subordinate references.

Thus LDAP referrals are incompatible with X.500 referrals. This means that LDAPv3 cannot be mapped to a strict subset of the X.500 Directory Abstract Service, and thus fails to meet one of its principal design goals ([LDAPv3] section 3.1).

5.3 Summary

LDAP referrals are inefficient in certain distributed configurations, and are incompatible with X.500 referrals.

These limitations are related because, as shown above, the extra information contained in X.500 referrals overcomes the inefficiency inherent in LDAP referrals.

6. The subcontexts URL Extension

This section defines a LDAP URL extension. The extension indicates that the URL does not refer to the entry named in the URL, but instead refers to all the context prefix entries which are immediately subordinate to the entry named in the URL.

The extension type is "subcontexts". The extension value is absent. The extension is critical.

eg:

```
ldap://serverb/o=ACME,c=US????!subcontexts
```

The subcontexts URL extension is returned within a LDAP URL which is part of a SearchResultReference, as defined in section 4.5.2 of [LDAPv3].

7. The Subordinate Contexts Control

A client may specify the following control when issuing a search request.

This control is included in the searchRequest message as part of the controls field of the LDAPMessage, as defined in Section 4.1.12 of [LDAPv3].

The control type is 1.2.826.0.1.1578918.2.1.2.1. The control SHOULD be marked as critical. There is no value - the controlValue field is absent.

The ASN.1 production for Control defined in [LDAPv3] is:

```
Control ::= SEQUENCE {
        controlType             LDAPOID,
        criticality             BOOLEAN DEFAULT FALSE,
        controlValue            OCTET STRING OPTIONAL }
```

The Subordinate Contexts Control is therefore encoded as follows:

```
3020
    041B 312E322E3832362E302E312E313537383931382E322E312E322E31
    0101 FF
```

8. Client Behaviour

The subcontexts URL extension is critical. Therefore, if a client does not support the subcontexts extension, it MUST NOT process the URL.

Clients which support the subcontexts extension MAY choose to process the URL or MAY choose not to process it. If the client chooses to process the URL, the search operation which is passed to the referred-to server MUST include the Subordinate Contexts Control.

9. Server Behaviour

There are two aspects to the Server behaviour:
- Returning the subcontexts URL extension
- Processing the Subordinate Contexts Control.

9.1 Returning the subcontexts URL extension

If a server is processing a Search and has found the base object, but does not hold the entire search area and is unable to chain, then each of the LDAP URLs which it returns may take one of the following forms.

- The DN refers to the entry from which to continue the search, and the subcontexts URL extension is absent.

  (This is the format defined in [URL].)

- The DN refers to the parent of all the context prefix entries from which to continue the search, and the subcontexts URL extension is present.

The formal rules governing the presence of the subcontexts URL extension are as follows.

- The subcontexts URL extension MUST NOT be returned in a URL which is returned from a non-Search operation, or in a URL for a Search operation where the server has been unable to find the base object.

  i.e. the subcontexts URL extension MUST NOT be present in a URL which is contained in a referral error.

- The subcontexts URL extension MAY be returned in a URL for a Search operation where the server has found the base object.

  i.e. the subcontexts URL extension MAY be returned in a URL which is contained in a SearchResultReference.

- If a server is unable to complete a search, and the search needs to be continued at many sibling context prefix nodes, all of which are stored in the same server, then the server SHOULD return a single URL with the subcontexts extension (in preference to many URLs without the subcontexts extension).

  This allows clients to chase the referrals more efficiently.

9.2 Processing the Subordinate Contexts Control

The Subordinate Contexts Control is critical. Therefore, if a server does not support it, it MUST NOT perform the operation, and MUST instead return the resultCode unsupportedCriticalExtension.

Servers which support the Subordinate Contexts Control MUST continue the Search from all of the entries which are context prefix subordinates of the baseObject. If there are no context prefix entries which are immediately subordinate to the baseObject, then the server MUST return the result code invalidReference.

10. Security Considerations

The server decision to return a referral containing the DN of the parent, rather than the DN of the entry itself, involves disclosing less, rather than more, information to the client.

There are therefore no security considerations over and above those involved in the decision to return a LDAP URL as considered in [URL].

11. References

[BRADNER97]
S. Bradner, "Key Words for use in RFCs to Indicate Requirement Levels", Internet Draft, draft-bradner-key-words-03.txt, January 1997.

[LDAPv2]
W . Yeong, T. Howes, S. Kille, "Lightweight Directory Access Protocol", RFC 1777, March, 1995.

[LDAPv3]
M. Wahl, T. Howes, S. Kille, "Lightweight Directory Access Protocol (v3)", Internet Draft draft-ietf-asid-ldapv3-protocol-09.txt, November, 1997.

[URL]
T. Howes, "The LDAP URL Format", Internet draft draft-ietf-asid-ldapv3-url-04.txt, August 1997.

[X518]
ITU-T Rec. X.518, "The Directory: Procedures for Distributed Operation", 1993.

12. Authors' Addresses

David Watts
Data Connection Ltd.
100 Church Street
Enfield
Middlesex
England
EN2 6BQ
EMail: djw@datcon.co.uk

Steve Orbell
Data Connection Ltd.
100 Church Street
Enfield
Middlesex
England
EN2 6BQ
EMail: so@datcon.co.uk