Rationale for changes:
i) we need to be able to store other sorts of knowledge reference such as cross references, that were not catered for in the original draft.
ii) we need to describe more fully how name resolution works, as this was not described in the original draft.
iii) there seems to be little point in making the knowledge ref a labelled URI, since we don't make full use of the URI features, so a simpler format is proposed. Replication or retrieval of an attribute on its own is not very useful (it is context-less), so we usually need to know the DN of the entry from which the attribute was obtained, in order to have useful information. Therefore it is quite acceptable that a knowledge reference comprises a combination of the DN of the entry and the attribute it holdings. This allows us to simplify the syntax of the knowledge ref substantially (which cant be bad).
When an entry is copied (replicated) both the DN of the entry and its attributes are replicated at the same time, so we can still replicate simplified knowledge references in this way. This is why the proposed sharedRef attribute does not need to be a URI, it can simply be a pointer to the server which holds the naming context named by the DN of the entry in which the sharedRef attribute is held. (Q. Do you really think that ftp and http URIs will be used as knowledge refs by LDAP servers? I did not think so, so I removed the protocol field If I am wrong, we can add back the protocol field.)

IETF LDAPEXT Working Group
INTERNET-DRAFT

David Chadwick
University of Salford
An adaption of an earlier ID by
Tim Howes
Netscape Communications Corp.
Mark Wahl
Critical Angle, Inc.
4 July 1999

Referrals and Knowledge References in LDAP Directories
<draft-ietf-ldapext-knowledge-00.txt>

1. Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

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."

The list of current Internet-Drafts can be accessed at

http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Distribution of this document is unlimited. Please send comments to the authors or the LDAPEXT mailing list, ietf-ldapext@netscape.com.

This draft is a revision of a draft formerly published as draft-ietf-ldapext-referral-00.txt.

This draft expires 4 January 2000

2. Abstract

This document defines two reference attributes and an associated "reference" object class for representing knowledge information in LDAP directories [RFC2251]. The object class can be used to construct entries in an LDAP directory containing references to other directories or services. This document also defines procedures directory servers should follow when supporting these schema elements.

3. Background and intended usage

The broadening of interest in LDAP directories beyond their use as front ends to X.500 directories has created a need to represent knowledge information in a more general way. Knowledge information is information about one or more LDAP servers maintained in another LDAP server, used to link servers and services together. Two types of knowledge reference are defined:

sharedRefs, which are knowledge references about other LDAP servers, both within and without the current global LDAP naming domain, and consequently may be copied between LDAP servers in this global LDAP naming domain, and
privateRefs, which are knowledge references only intended for use by the holding server, and should not be copied to other LDAP servers.

This document draws on the following concept:
A global LDAP naming domain is a set of one or more LDAP servers, that may be autonomously or collectively managed, and fully or partially interconnected, but which have all been allocated their LDAPDNs according to one or more non-conflicting naming schemes which guarantee that each entry in the global LDAP naming domain has an unambiguous LDAPDN (excluding of course replicated entries which will have the same DN !) Thus LDAP servers that use the DC naming scheme are within one global LDAP naming domain. LDAP servers within the Nameflow-Paradise service, that use the non-conflicting country name scheme (based on X.521) are also part of this same LDAP naming domain.

The key words "MUST", "SHOULD", and "MAY" used in this document are to be interpreted as described in [BRADNER97].

4. Knowledge references

Knowledge references are composed of two components:

a) the DN of the entry (DSE) in which they are stored, and

b) a knowledge reference attribute which contains a host component (host name and optional port number) and a reference type field.

The DN of the entry in which knowledge references are stored is also the name of the context prefix entry of the naming context in the referred to server.

Two knowledge reference attribute types are defined, sharedRefs and privateRefs.

4.1 The sharedRef attribute type

This section defines the sharedRef attribute type for holding general knowledge reference information, that may be shared between servers in this global LDAP naming domain.

```
( 2.16.840.1.113730.3.1.x NAME 'sharedRef' DESC 'a shareable knowledge
reference'
  EQUALITY caseExactIA5Match SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  USAGE distributedOperation )
```

The sharedRef attribute type has IA5 syntax and is case sensitive. The sharedRef attribute is multivalued. Each value represents a different LDAP server holding (usually a copy of) the same naming context. Values placed in the attribute MUST conform to values of <sharedRef> according to the following syntax

<sharedRef> ::= <host>'/'<type>

<host> is the standard host and optional port syntax of a URL.

<type> ::= 'cross' | 'sub' | 'nssr' | 'external'

'cross' indicates that this is a cross reference pointing to another naming context within this global LDAP naming domain.
'sub' indicates that this is a subordinate reference pointing to a subordinate naming context in this global LDAP naming domain.
'nssr' indicates that this is a non-specific subordinate reference pointing to a subordinate naming context in this global LDAP naming domain.
Note nssr can be deleted if we agree that we will not support this.
'external' indicates that this is a knowledge reference pointing to an LDAP server in a different global LDAP naming domain, that possibly holds quite different information about entries in the referenced naming context, to servers within this global LDAP naming domain.

When a client is trying to retrieve information from a particular naming context, it should contact each host of type external, as well as at least one host of type cross or subr [or nssr].

4.2 The privateRef attribute type

This section defines the privateRef attribute type for holding general knowledge reference information, that is specific to a server and may not be shared between servers.

```
( 2.16.840.1.113730.3.1.y NAME 'privateRef' DESC 'a private reference'
  EQUALITY caseExactIA5Match SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  USAGE dSAOperation )
```

The privateRef attribute type has IA5 syntax and is case sensitive. The privateRef attribute is multivalued. Values placed in the attribute MUST conform to values of <privateRef> according to the following syntax

```
<privateRef> ::= <host>'/'<type>
```

<host> is the standard host optional port syntax of a URL.

```
<type> ::= 'me' | 'sup' | 'supplier' | 'consumer'
```

'me' is used to hold the host address of this server, and is always held in the root DSE
'sup' is used to hold the host address of a server superior to this one in this global LDAP naming domain e.g. a server holding the dc=com node, or the c=gb node. The 'sup' privateRef is always held in the root DSE.
'supplier' is used to hold the address of the server that is providing this replicated naming context
'consumer' is used to hold the address of a server that is being provided with a copy of this naming context.

5. Use of the knowledge attributes

Except when the manageDsaIT control (documented in section 7 of this document) is present in the operation request, the knowledge attributes are not visible to clients, except when its value is returned in referrals or continuation references.

If the manageDsaIT control is not set, and the entry named in a request is either the same as or is subordinate to the DSE holding a sharedRef attribute, the server returns an LDAPResult with the resultCode field set to "referral" and the referral field set to contain the value(s) of the host component of the sharedRef attribute and the LDAPDN in the request.

If the manageDsaIT control is not set, and an entry containing a sharedRef attribute is otherwise in the scope of a one level or subtree search request, the server returns a SearchResultReference for each such entry and the field is set to contain the value(s) of the host component of the sharedRef attribute and the DN of the DSE holding the sharedRef.

When the manageDsaIT control is present in a request, the server will treat an entry containing a knowledge reference attribute as an ordinary entry, and the knowledge reference attribute as an ordinary attribute, and the server will not return referrals or continuation references corresponding to the knowledge reference attributes.

The following sections define three uses for the knowledge reference attributes,

namely: name resolution for any operation, one level search evaluation and full subtree search evalution.

5.1. Name resolution for any operation

Clients SHOULD perform at least simple "depth-of-referral count" loop detection by incrementing a counter each time a new set of referrals is received. (The maximum value for this count should be twice the number of RDNs in the target object less one, to allow for ascending and descending the DIT.) Clients MAY perform more sophisticated loop detection, for example not chasing the same referral twice.

If the client requests an operation in which the target entry is below an entry (DSE) holding a shareRef attribute, and this entry (DSE) is a leaf entry in the current server, then the server will return an LDAPResult with the result code field set to referral, and the referral field set as follows:

Host component is copied from the sharedRef attribute in the leaf DSE
LDAPDN is copied from the operation.

If the client requests an operation in which the target entry is below an entry (DSE) holding a shareRef attribute, but this DSE is not a leaf entry in the current server, but the server knows that the target entry is not within a local naming context, then the server will return an LDAPResult with the result code field set to referral, and the referral field set as follows:

Host component is copied from the sharedRef attribute of the lowest non-leaf DSE above the target entry
LDAPDN is copied from the operation

If the client requests an operation in which the target entry is not present in the current server, and not below an entry held in the current server, then the server will return an LDAPResult with the result code field set to referral, and the referral field set as follows:

Host component is copied from the privateRef attribute of type 'sup' in the root DSE
LDAPDN is copied from the operation

5.2 One-level Search evaluation

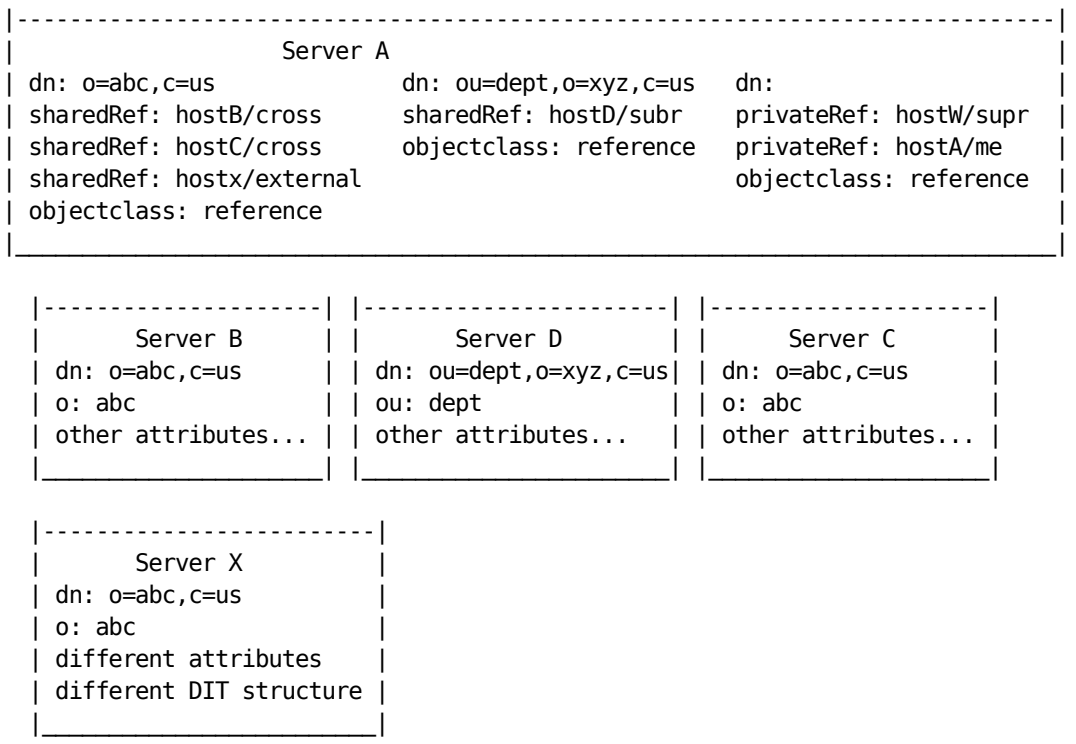
If an entry containing a sharedRef attribute is immediately subordinate to the base object named in a one level search request, then the referring server MUST include a scope of "base" in any LDAP URIs returned in the corresponding SearchResultReference. Host is copied from the sharedRef attribute and the LDAPDN is copied from the name of the entry holding the sharedRef attribute.

5.3 Full Subtree Search Evalution

If an entry containing a sharedRef attribute is within the scope of a subtree search evaluation, the server returns a SearchResultReference for each such entry set to contain the value(s) of the host component of the sharedRef attribute and the DN of the DSE holding the sharedRef.

5.4 Example

A multi-valued sharedRef attribute MAY be used to indicate different locations for the same resource and different locations for a similarly named though different resource. An example configuration illustrating the use of the sharedRef attribute in this capacity is provided below.



In this example, Server A holds the naming context o=xyz,c=us, and a subordinate reference to the subordinate naming context ou=dept,o=xyz,c=us. It holds a superior reference to hostW (probably holding the c=us naming context, although it is not necessary to know the name of the superior naming context for name resolution), as well as its own reference. It also holds two cross references and an external reference to the o=abc,c=us naming context. This implies that hostB and hostC hold copies of the same naming context information, but that hostX holds different information for this naming context, probably structured differently, and holding different attributes in the entries.

In the following protocol interaction examples, the client has contacted Server A. Server A holds the naming context "o=xyz,c=us".

5.4.1. Subtree search from a superior naming context

If a client requests a subtree search of "o=xyz,c=us", then in addition to any entries in the "o=xyz,c=us" naming context which match the filter, Server A will also return a continuation reference for "ou=dept,o=xyz,c=us".

One possible response might be:

```
        ... SearchResultEntry responses ...

        SearchResultReference {
```

```
         ldap://hostD/ou=dept,o=xyz,c=us
        }

        SearchResultDone "success"
```

6. The reference object class

The reference object class is defined as follows.

```
( 2.16.840.1.113730.3.2.z NAME 'reference' SUP top AUXILIARY
  MAY ( sharedRef | privateRef ) )
```

The reference object class is a subclass of top and may contain either knowleldge reference attributes. It is an auxiliary object class.

Servers MAY support the knowledge reference attributes through use of the reference object class. Servers MAY also support the knowledge reference attributes as operational attributes in any entry, or through use of other object classes.

7. The manageDsaIT control

A client MAY specify the following control when issuing a search, compare, add, delete, modify, or modifyDN request.

The control type is 2.16.840.1.113730.3.4.2. The control SHOULD be marked as critical. There is no value; the controlValue field is absent.

This control causes entries with the knowledge reference attributes to be treated as normal entries, allowing clients to read and modify these entries.

8. Security Considerations

This document defines mechanisms that can be used to "glue" LDAP (and other) servers together. The information used to specify this glue information should be protected from unauthorized modification. If the server topology information itself is not public information, the information should be protected from unauthorized access as well.

9. References

[RFC1738]
Berners-Lee, T., Masinter, L., and McCahill, M., "Uniform Resource Locators (URL)", RFC 1738, CERN, Xerox Corporation, University of Minnesota, December 1994, <URL:ftp://ds.internic.net/rfc/rfc1738.txt>

[RFC2251]
M. Wahl, T. Howes, S. Kille, "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997. 1997.

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

[X500]
    ITU-T Rec. X.501, "The Directory: Models", 1993.

[RFC2079]
    M. Smith, "Definition of an X.500 Attribute Type and an Object Class to Hold Uniform Resource Identifiers (URIs)", RFC 2079, January 1997.

11. Author's Address

Tim Howes
Netscape Communications Corp.
501 E. Middlefield Rd.
Mountain View, CA 94043
USA
EMail: howes@netscape.com

Mark Wahl
Critical Angle Inc.
4815 W Braker Lane #502-385
Austin, TX 78759
USA
EMail: M.Wahl@critical-angle.com

David Chadwick
University of Salford
The Crescent
Salford
M5 4WT
England
Email D.W.Chadwick@iti.salford.ac.uk