IETF LDAPEXT Working Group   Tim Howes
INTERNET-DRAFT   Netscape Communications Corp.
                                                              Mark Wahl
                                                    Critical Angle, Inc.
                                                             March 1998

Referrals and Knowledge References in LDAP Directories
<draft-ietf-ldapext-referral-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 learn the current status of any Internet-Draft, please check the ``1id-abstracts.txt'' listing contained in the Internet- Drafts Shadow Directories on ds.internic.net (US East Coast), nic.nordu.net (Europe), ftp.isi.edu (US West Coast), or munnari.oz.au (Pacific Rim).

Distribution of this memo is unlimited. Editorial comments should be sent to the authors. Technical discussion should take place on the IETF LDAPEXT mailing list (ietf-ldapext@netscape.com).

This draft is a revision of a draft formerly published as draft-ietf-asid-ldapv3-referral-00.txt.

2. Abstract

This document defines a "ref" attribute and associated "referral" object class for representing generic knowledge information in LDAP directories [RFC2251]. The attribute uses URIs [RFC1738] to represent knowledge, enabling LDAP and non-LDAP services alike to be referenced. 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 servers maintained in another server, used to link servers and services together.

This document defines a general method of representing knowledge information in LDAP directories, based on URIs.

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

4. The ref attribute type

This section defines the ref attribute type for holding general knowledge reference information.

```
( 2.16.840.1.113730.3.1.34 NAME 'ref' DESC 'URL reference'
  EQUALITY caseExactIA5Match SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  USAGE distributedOperation )
```

The ref attribute type has IA5 syntax and is case sensitive. The ref attribute is multivalued. Values placed in the attribute MUST conform to the specification given for the labeledURI attribute defined in [RFC2079]. The labeledURI specification defines a format that is a URI, optionally followed by whitespace and a label. This document does not make use of the label portion of the syntax. Future documents MAY enable new functionality by imposing additional structure on the label portion of the syntax as it appears in the ref attribute.

5. Use of the ref attribute

Three uses for the ref attribute are defined in this document. Other uses of the ref attribute MAY be defined in subsequent documents, or by bilateral agreement between cooperating clients and servers.

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

If the manageDsaIT control is not set, and the entry named in a request contains the ref attribute, and the entry is not the root DSE, the server returns an LDAPResult with the resultCode field set to "referral" and the referral field set to contain the value(s) of the ref attribute.

If the manageDsaIT control is not set, and an entry containing the ref attribute is otherwise in the scope of a one level or subtree search request, the server returns a SearchResultReference for each such entry containing the value(s) of the entry's ref attribute.

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

The following sections define three uses for the ref attribute.

5.1. Named reference

This use of the ref attribute is similar to the subordinate reference concept found in X.500 [X500]. It is used to facilitate distributed name resolution or search across multiple servers. The ref attribute appears in an entry named in the referencing server. The value of the ref attribute points to the corresponding entry maintained in the referenced server.

While the distinguished name in a value of the ref attribute is typically that of an entry in a naming context below the naming context held by the referencing server, it is permitted to be the distinguished name of any entry. If the ref attribute is multi-valued all the DNs in the values of the ref attribute SHOULD have the same value. It is the responsibility of clients to not loop repeatedly if a naming loop is present in the directory. Administrators SHOULD avoid configuring naming loops using referrals.

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. Clients MAY perform more sophisticated loop detection, for example not chasing the same URI twice.

If an entry containing the ref 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.

5.1.1. Examples

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

```
       |------------------------------------------------------------|
       |                    Server A                                |
       | dn: o=abc,c=us                 dn: o=xyz,c=us              |
       | ref: ldap://hostB/o=abc,c=us  ref: ldap://hostD/o=xyz,c=us |
       | ref: ldap://hostC/o=abc,c=us  objectclass: referral        |
       | objectclass: referral                                      |
       |____________________________________________________________|

  |---------------------| |---------------------| |---------------------|
  |       Server B      | |       Server D      | |      Server C       |
  | dn: o=abc,c=us      | | dn: o=xyz,c=us      | | dn: o=abc,c=us      |
  | o: abc              | | o: xyz              | | o: abc              |
  | other attributes... | | other attributes... | | other attributes... |
  |_____________________| |_____________________| |_____________________|
```

In this example, Server A holds references for two entries: "o=abc,c=us" and "o=xyz,c=us". For the "o=abc,c=us" entry, Server A holds two references, one to Server B and one to Server C. The entries referenced are replicas of each other. For the "o=xyz,c=us" entry, Server A holds a single reference to the entry contained in Server D.

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

5.1.1.1. Subtree search from a superior naming context

If a client requests a subtree search of "c=us", then in addition to any entries in the "c=us" naming context which match the filter, Server A will also return two continuation references. One of the continuation references will be for "o=abc,c=us", and the other continuation reference will be for "o=xyz,c=us".

The order in which the continuation references are returned, and the order of LDAP URI values in each continuation reference, are not standardized. One possible response might be:

```
... SearchResultEntry responses ...

SearchResultReference {
 ldap://hostB/o=abc,c=us
 ldap://hostC/o=abc,c=us
}

SearchResultReference {
 ldap://hostD/o=xyz,c=us
}

SearchResultDone "success"
```

5.1.1.2. One level search from an immediately superior object

If the client requests a one level search of "c=us", then in addition to any entries in the "c=us" naming context which match the filter, Server A will also return two continuation references, as in the previous example. One possible response might be:

```
        ... SearchResultEntry responses ...

        SearchResultReference {
         ldap://hostB/o=abc,c=us??base
         ldap://hostC/o=abc,c=us??base
        }

        SearchResultReference {
         ldap://hostD/o=xyz,c=us??base
        }

        SearchResultDone "success"
```

Note the inclusion of the "base" scope in the returned URL continuation references. This is required to maintain the one-level search semantics.

5.1.1.3. Other operations

If the client requests an operation in which the base or target entry has a ref attribute, then the server returns an LDAPResult with the resultCode field set to referral and the referral field set to the value(s) of the ref attribute. If the operation is a search, the referring server does not return any SearchResultEntry or SearchResultReference before the SearchResultDone.

For example, if the client had issued a subtree search of "o=abc,c=us", the server would return

```
        SearchResultDone "referral" {
         ldap://hostB/o=abc,c=us
         ldap://hostC/o=abc,c=us
        }
```

Similarly, if the client had issued a modify of "o=xyz,c=us", the server would return

```
        ModifyResponse "referral" {
         ldap://hostD/o=xyz,c=us
        }
```

5.2. Superior Reference

This use of the ref attribute is similar to the superior reference concept found in X.500 [X500]. An LDAP server's root DSE MAY contain the "ref" attribute. The values of the ref attribute in the root DSE that are LDAP URIs SHOULD NOT contain any dn part, just the host name and optional port number.

When the server receives an operation for which the base or target entry of the request is not contained in or subordinate to any naming context held by the server, the server will return an LDAPResult with the resultCode set to "referral", and with the referral field filled in using the values from the "ref" attribute from the root DSE.

5.3. Unnamed reference

This use of the ref attribute is similar to the nonspecific subordinate reference concept found in X.500 [X500]. It goes beyond this concept to facilitate distributed searching or indexing across multiple servers. The ref attribute is used to name an entry in the referencing server. The reference entry may contain other attributes used to select the reference during searching.

A multi-valued ref attribute MAY indicate the locations of different resources all associated with the same LDAP entity. The following example illustrates the use of the ref attribute to indicate two unnamed references.

```
   |------------------------------------------------------------------|
   |                       Server A                                   |
   | dn: ref=ldap://hostB/o=abc,c=us  dn: ref=ldap://hostC/o=xyz,c=us |
   | cn: babs                         cn: babs                        |
   | cn: gern                         cn: bob                         |
   | cn: bob                                                          |
   |__________________________________________________________________|

  |-----------------------------| |-----------------------------|
  |          Server B           | |          Server C           |
  | dn: o=abc,c=us              | | dn: o=xyz,c=us              |
  | o: abc                      | | o: xyz                      |
  | other attributes...         | | other attributes...         |
  |                             | |                             |
  | dn: cn=babs,o=abc,c=us      | | dn: cn=babs,o=xyz,c=us      |
  | cn: babs                    | | o: xyz                      |
  | other attributes...         | | other attributes...         |
  |                             | |                             |
  | dn: cn=gern,o=abc,c=us      | | dn: cn=bob,o=xyz,c=us       |
  | cn: gern                    | | cn: bob                     |
  | other attributes...         | | other attributes...         |
  |                             | |_____________________________|
  | dn: cn=bob,o=abc,c=us       |
  | cn: bob                     |
  | other attributes...         |
  |_____________________________|
```

In this example Server A contains two unnamed references to servers B and C. The unnamed reference entries have additional cn attribute values which may be used during a search operation to select the reference for return to a client.

6. The referral object class

The referral object class is defined as follows.

```
( 2.16.840.1.113730.3.2.6 NAME 'referral' SUP top STRUCTURAL
  MAY ( ref $ * ) )
```

The referral object class is a subclass of top and may contain the referral attribute. It is a structural object class. The referral object class may also contain any other attribute, as indicated by the "*" in the MAY portion of the definition. This is required to support the naming attributes used in the entry's distinguished name.

Servers MAY support the ref attribute through use of the referral object class. Servers MAY also support the ref attribute as an operational attribute 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 "ref" attribute to be treated as normal entries, allowing clients to read and modify these entries.

This control is not needed if the entry containing the referral attribute is one used for directory administrative purposes, such as the root DSE, or the server change log entries. Operations on these entries never cause referrals or continuation references to be returned.

8. Relationship to X.500 Knowledge References

The X.500 standard defines several types of knowledge references, used to bind together different parts of the X.500 namespace. In X.500, knowledge references can be associated with a set of unnamed entries (e.g., a reference, associated with an entry, to a server containing the descendants of that entry).

This creates a potential problem for LDAP clients resolving an LDAPv3 URL referral referring to an LDAP directory back-ended by X.500. Suppose the search is a subtree search, and that server A holds the base object of the search, and server B holds the descendants of the base object. The behavior of X.500(1993) subordinate references is that the base object on server A is searched, and a single continuation reference is returned pointing to all of the descendants held on server B.

An LDAP URL only allows the base object to be specified. It is not possible using standard LDAP URLs to indicate a search of several entries whose names are not known to the server holding the superior entry.

X.500 solves this problem by having two fields, one indicating the progress of name resolution and the other indicating the target of the search. In the above example, name resolution would be complete by the time the query reached server B, indicating that it should not refer the request.

This document does not address this problem. This problem will be addressed in separate documents which define the changes to the X.500 distribution model and LDAPv3 extensions to indicate the progress of name resolution.

9. 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.

10. 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