IETF LDAPEXT Working Group   Roland Hedberg
Internet-Draft   Catalogix
Expires: January 12, 2000   July 12, 2000

Referrals in LDAP Directories
<draft-ietf-ldapext-refer-00.txt>

Status of this Memo

This Internet-Draft will expire on January 12, 2000.

Copyright Notice

Abstract

This document defines two reference attributes 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 and when responding to requests for which the directory server does not contain the requested object but may contain some knowledge of the location of the requested object.

1. 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 is based on the following basic assumptions:

- several naming domains
The usage of LDAP as a access protocol to other than X.500 servers has created islands of directory service systems containing one or more LDAP servers. Each of these islands are free to pick their own naming domain. And that they also do; some use the old country,organization, organizationalUnit naming scheme[X.521], some use the newer domain name based naming scheme but these two are in no way the only ones in use. The existence of several naming domains are in itself no real problem as long as they produce unique names for the objects in the directory.
Still naming schemes like the domain name based one, might easily create non-continues naming structures because some toplevel domain names might no find organizations that are interested and/or willing to manage them. Therefor tree transversal might not longer be possible except in parts of the whole tree.

- authoritive structure vs directory structure
In some instances even if a part of the tree is delegated to one organization, the organization doing the delegation might want to remain as the authority for the baseobject of the delegated tree.

- support for onelevel searches
At points in the tree where the responsibility for all or almost all of the children of a object is delegated to different organizations and resides in different directory servers a one-level search is not

very efficient if not supported by special facilities in the directory as such.

-- directory server discovery
LDAP servers that do not use dc nameing or are not registered with SRV records in the DNS are very hard to find.

This document defines a general method of representing knowledge information in LDAP directories, based on URIs.
Two types of knowledge reference are defined: refer and subRefer.

The key words "MUST", "SHOULD", and "MAY" used in this document are to be interpreted as described in [RFC2119].

2. Knowledge references

2.1 The refer attribute

```
( 1.2.752.17.1.100
  NAME 'refer'
  DESC 'URL reference'
  EQUALITY caseExactIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  USAGE distributedOperation )
```

The refer attribute type has IA5 syntax and is case sensitive.
It is multivalued. Values placed in the attribute MUST conform to the specification given for the labeledURI attribute as 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 a refer attribute.
If the URI contained in a refer attribute refers to an LDAP server, it must be in the LDAP URI format described in [RFC2255].

When returning a referral result, the server must not return the label portion of the labeledURI as part of the referral. Only the URI portion of the refer attributes should be returned.

The refer attribute can be further specified by the use of options as defined in section 4.1.5 of [RFC2251]. This document defines five options and their use. Future documents might defined other options.

The options defined are:
"me", "sup", "cross", "nssr" and "sub" .

'refer;me' is used to hold the reference of this server, and is always held in the root DSE

'refer;sup' is used to hold the reference of a server superior to this one in this global LDAP naming domain e.g. a server holding the dc=com, dc=se, or the c=se node. The 'refer;sup' is always held in the root DSE.

'refer;cross' indicates that this is a cross reference pointing to another naming context within or outside this global LDAP naming domain.

'refer;sub' indicates that this is a subordinate reference pointing to a subordinate naming context in this global LDAP naming domain.

'refer;nssr' indicates that this is a non-specific subordinate reference pointing to a subordinate naming context in this global LDAP naming domain.

3. Use of the knowledge attribute

Except when the manageDsaIT control (documented in section 6 of this document) is present in the operation request, the refer 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 refer 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 refer attribute minus any optional trailing whitespace and labels that might be present.

If the manageDsaIT control is not set, and an entry containing the ref attribute is 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 refer attribute.

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

4 Behaviour specification

4.1 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.

Case 1: The target entry is not held by the server and is superior to some entry held by the server.

If the server DSE contains a "refer;sup" attribute then the server will return an LDAPResult with the result code field set to referral, and the referral field set to contain the value(s) of the "refer;sup" attribute minus any optional trailing whitespace and labels that might be present.

Case 2: The target entry is not held by the server and is subordinate to some entry, held by the server, that contains a refer attribute.

The server will return an LDAPResult with the result code field set to referral, and the referral field set to contain the value(s) of the refer attribute minus any optional trailing whitespace and labels that might be present.

Case 3: The target entry is held by the server and contains a refer attribute without the 'nssr' option.

The server will return an LDAPResult with the result code field set to referral, and the referral field set to contain the value(s) of the refer attribute minus any optional trailing whitespace and labels that might be present.

Case 4: The target entry is not held by the server, and is not subordinate or superior to any object held by the server.

If the server contains a "refer;cross" attribute
in the root DSE with a baseobject that is either the same or
superior to the target entry then
the server will return an LDAPResult with the result code field set
to referral, and the referral field set to contain the value(s) of
these refer attributes minus any optional trailing whitespace and labels
that might be present.

4.2 Search evaluation

For search operations, once the base object has been found and determined NOT to contain a refer attribute without the 'nssr' option, the search may progress.

4.2.1 base-level

If the entry matches the filter and does NOT contain a refer attribute
it will be returned to the client as described in [RFC2251].
If the entry matches the filter contains a refer attribute without
the 'nssr' option it will be returned as a referral as described here.

If a matching entry contains a refer attribute and the URI contained in the refer attribute is NOT an LDAP URI [RFC2255], the server should return the URI value contained in the refer attribute of that entry in a SearchResultReference.

If a matching entry contains a refer attribute in the LDAP URI syntax, the server will return an SearchResultReference containing the value(s) of the refer attribute minus any optional trailing whitespace and labels that might be present. The URL from the refer attribute must be modified before it is returned by adding or substituting a "base" scope into the URL. If the URL does not contain a scope specifier, the "base" scope specifier must be added. If the URL does contain a scope specifier, the existing scope specifier must be replaced by the "base" scope.

4.2.2 One-level

Any entries matching the filter and one level scope that do NOT contain a refer attribute are returned to the client normally as described in [RFC2251]. Any entries matching the filter and one level scope that contains a refer attribute without the 'nssr' option must be returned as referrals as described here.

If a matching entry contains a refer attribute and the URI contained in the refer attribute is NOT an LDAP URI [RFC2255], the server should return the URI value contained in the refer attribute of that entry in a SearchResultReference.

If a matching entry contains a refer attribute in the LDAP URI syntax, the server will return an SearchResultReference containing the value(s) of the refer attribute minus any optional trailing whitespace and labels that might be present. The URL from the refer attribute must be modified before it is returned by adding or substituting a "base" scope into the URL. If the URL does not contain a scope specifier, the "base" scope specifier must be added. If the URL does contain a scope specifier, the existing scope specifier must be replaced by the "base" scope.

4.2.3 Subtree search evaluation

Any entries, held by the server, matching the filter and subtree scope that do NOT contain a refer attribute or contains a refer attribute with the 'nssr' option are returned to the client normally as described in [RFC2251]. Any entries matching the subtree scope and containing a refer attribute must be returned as referrals as described here.

If a matching entry contains a refer attribute and the URI contained in that attribute is NOT an LDAP URI [RFC2255], the server should return the URI value contained in the refer attribute of that entry in a SearchResultReference.

If a matching entry contains a refer attribute in the LDAP URI syntax, the server will return an SearchResultReference containing the value(s) of the refer attribute minus any optional trailing whitespace and labels that might be present.

N.B. in subtree search evaluation a entry containing a refer attribut with the 'nssr' option might appear twice in the result, first as a entry and then as a reference. A client following all references might therefore end up with a resultset containing two representations of the same entry, one from the server getting the original query and one from the server that the 'nssr' reference points to.

5. The referral object class

The referral object class is defined as follows.

```
( 1.2.752.17.2.10
  NAME 'referral'
  SUP top
  STRUCTURAL
  MAY ( refer ) )
```

The referral object class is a subclass of top and may contain the refer attribute. The referral object class should, in general, be used in conjunction with the extensibleObject object class to support the naming attributes used in the entry's distinguished name.

Servers must support the refer attributes through use of the referral object class. Any named reference must be of the referral object class and will likely also be of the extensibleObject object class to support naming and use of other attributes.

6. 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.

7. Superior Reference

This document defines two types of knowledge references that point to parts of the naming context that is above of beyone the part held by a server.
The 'sup' option when referring to a LDAP server that holds a naming context that is closer to the root of the same naming context and 'other' when referring to a LDAP server that holds a naming context that belongs to a different naming domain then the one the server belongs to.

Thus if the server receives a request for an operation where the target entry is a entry closer to the root than the naming context held the server and if the server holds a 'refer;sup' attribute in the DSE, then the server MUST return an LDAPResult with the result code field set to referral, and the referral field set to contain the value(s) of the 'refer;sub' attribute minus any optional trailing whitespace and labels that might be present.

On the other hand if the server receives a request for an operation where the target entry is a entry that belongs to a other naming domain and if there is any 'refer;other' attributes in the DSE with a base entry that belongs to the same naming domain as the target entry and is closer to the root then the target entry, then the server SHOULD return an LDAPResult with the result code field set to referral, and the referral field set to contain the value(s) of the 'refer;other' attribute minus any optional trailing hitespace and labels that might be present.

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,

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

[RFC2119]
S. Bradner, "Key Words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997. (Format: TXT=4723 bytes) (Also BCP0014) (Status: BEST CURRENT PRACTICE)

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

[RFC2255]
T. Howes, M. Smith, "The LDAP URL Format", RFC 2255, December, 1997. (Format: TXT=20685 bytes) (Status: PROPOSED STANDARD)

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

[X521]
ITU-T Rec. X.521, "---------------------", 1993.

12. Acknowledgements

This draft is heavily based on the previous drafts on knowledge references in LDAP written by Christopher Lukas, Tim Howes, Michael Roszkowski, Mark C. Smith, Mark Wahl and David Chadwick. Peter Valkenburg and Henny Bekker has also made valueable contributions.

13. Authors Address

Roland Hedberg
Catalogix
Dalsveien 53
0775 Oslo
Norway
EMail: Roland@catalogix.se

Appendix A

Example of usage.
Information stored in a server.

```
  dn:
  objectclass: referral
  refer;me: ldap://hostCAT/dc=cat,dc=se
  refer;sup: ldap://hostSE/dc=se
  refer;cross: ldap://hostNO/dc=no
  refer;cross: ldap://hostNL/c=nl

  dn: dc=cat,dc=se
  objectclass: domain
  dc: cat

  dn: dc=one,dc=cat,dc=se
  objectclass: extendedObject
  objectclass: referral
  refer;nssr: ldap://hostCAT1/dc=one,dc=cat,dc=se
  ou: one
  l: umea

  dc: dc=two,dc=cat,dc=se
  objectclass: referral
  objectclass: extendedObject
  refer;sub: ldap://hostCAT2/dc=two,dc=cat,dc=se

  dn: dc=three,dc=cat,dc=se
  objectclass: referral
  objectclass: extendedObject
  refer;cross: ldap://hostCAT3/dc=cat,dc=nl

  dc: dc=four,dc=cat,dc=se
  objectclass: domain
  objectclass: extendedObject
  ou: four
  l: umea
```

```
   ==========================================
      A number of descriptive cases
   ==========================================

   case 1: One-level search, target object on the server
     search
       baseobject: dc=cat,dc=se
       scope:      onelevel
       filter:     (objectclass=*)
       attributes: ou

     returns
       searchResultEntry {
         dn: dc=one,dc=cat,dc=se
         ou: one
       }
       searchResultReference {
         ldapurl: ldap://hostCAT2/dc=two,dc=cat,dc=se
       }
       searchResultReference {
         ldapurl: ldap://hostCAT3/dc=cat,dc=nl
       }
       searchResultEntry {
         dn: dc=four,dc=cat,dc=se
         ou: four
       }
       searchResultDone {
         resultCode: success
       }

   case 2: Subtree search, target object on the server
     search
       baseobject: dc=cat,dc=se
       scope:      subtree
       filter:     (objectclass=*)
       attributes: ou

     returns
       searchResultEntry {
         dn: dc=one,dc=cat,dc=se
         ou: one
       }
       searchResultReference {
         ldapurl: ldap://hostCAT1/dc=one,dc=cat,dc=se
       }
       searchResultReference {
         ldapurl: ldap://hostCAT2/dc=two,dc=cat,dc=se
       }
```

```
       searchResultReference {
         ldapurl: ldap://hostCAT3/dc=cat,dc=nl
       }
       searchResultEntry {
         dn: dc=four,dc=cat,dc=se
         ou: four
       }
       searchResultDone {
         resultCode: success
       }
```

case 3: base search, target entry contains a 'refer;nssr' attribute

```
     search
       baseobject: dc=one,dc=cat,dc=se
       scope:      base
       filter:     (objectclass=*)
       attributes: ou

     returns
       searchResultEntry {
         dn: dc=one,dc=cat,dc=se
         ou: four
       }
       searchResultDone {
         resultCode: success
       }
```

case 4: base search, target entry contains a 'refer;sub' attribute

```
     search
       baseobject: dc=two,dc=cat,dc=se
       scope:      base
       filter:     (objectclass=*)
       attributes: ou

     returns
       searchResultDone {
         resultCode: referral
         matchedDN: dc=two,dc=cat,dc=se
         referral: ldap://hostCAT2/dc=two,dc=cat,dc=se
       }
```

```
  case 5: one-level search, target entry contains a 'refer;nssr' attribute
     search
       baseobject: dc=one,dc=cat,dc=se
       scope:      onelevel
       filter:     (objectclass=*)
       attributes: ou

       searchResultDone {
         resultCode: referral
         matchedDN: dc=one,dc=cat,dc=se
         referral: ldap://hostCAT1/dc=one,dc=cat,dc=nu
       }

  case 6: Search on area above the baseobject of the server
     search
       baseobject: dc=pi,dc=se
       scope:      subtree
       filter:     (objectclass=*)
       attributes: ou

     returns
       searchResultDone {
         resultCode: referral
         matchedDN:  dc=se
         referral:   ldap://hostSE/dc=se
      }



  case 7: Search on area beyond, but not below the baseobject
        of the server
     search
       baseobject: o=surfnet,c=nl
       scope:      base
       filter:     (objectclass=*)

     returns
       searchResultDone {
         resultCode: referral
         matchedDN:  c=nl
         referral:   ldap://hostNL/c=NL
       }
```