LDAP Partial Entry Control

1. Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026 [1].

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. Technical discussion of this document will take place on the IETF LDAP Extensions Working Group mailing list <ietf-ldapext@netscape.org>. Please send editorial comments directly to the authors <gvithalprasad@novell.com> <sharipriya@novell.com>. This draft expires July, 2003.

2. Abstract

This document defines a mechanism to augment the LDAPv3 [RFC2251] search operation through an LDAP search control for sending a partial search entry to an LDAP client as result of a search operation in the SearchResultEntry LDAP PDU. This mechanism lets the LDAP client specify size and time limits to be applied to the server-side processing of a single SearchResultEntry.

This document defines two LDAP controls required to support the above requirement.

3. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [2].

4. Introduction

An LDAP v3 search returns the LDAP client all the values for the requested attributes along with the DN for each entry matching the search filter, in a single SearchResultEntry message. The SearchResultEntry sent as a part of the search response is defined
in [3] and repeated here for reference:

SearchResultEntry ::= SEQUENCE {
  ObjectName LDAPDN,
  Attributes PartialAttributeList }

PartialAttributeList ::= SEQUENCE OF SEQUENCE {
   type AttributeDescription
   vals SET OF AttributeValue }

There is currently no way for the server to break up the entry or for the client to ask the server to return the information for a single entry, in multiple smaller messages based on a client-specified size and time limit.

While this is not a problem for smaller entries, this can become a problem when huge multi-valued attributes are handled in an entry. Some of these may be constructed dynamically by the LDAP server like dynamic groups [DYNGRP], or may be statically stored by an underlying X.500 or other database backend.

In such cases, though the server would have processed some values of the entry, the server would not be able to send the SearchResultEntry message until all the values are processed. This poses a problem for clients who wish to receive portions of a search result entry as it becomes available. This could be either because it does not want to handle a large data set or it may want to process the available data while the server is returning more.

Removing this limitation may also help the LDAP server implementation save resources while processing entries having very large numbers of attribute values.

This document attempts to address this problem by introducing a LDAP search control, which allows the client to specify time and/or size limits for the entries that are returned by the LDAP server. If the LDAP server fails to process a complete entry within the specified limits, it will send a partial entry in SearchResultEntry to the client with a response control indicating that this is a partial response. The LDAP server will then typically send the remaining entry in subsequent SearchResultEntry responses, each time obeying the client specified limits, until the complete entry is returned to the client.

This is helpful for clients when retrieving the results of a search with less latency, especially when individual entry sizes are quite large or where the attribute values are computed at run-time. One such scenario is the listing of a member attribute of a dynamic group, based on the criteria specified in the memberQueryURL attribute [4]. Here, the computation of all the values of an attribute could take a long time depending on the criteria, but a few values of the attribute may be available in a relatively short time and can be sent back to the client as a partial entry.

5. PartialEntryRequest

This control is included in the SearchRequest message as part of the controls field of the LDAPMessage as defined in [LDAPv3]. The controlType is set to OID <ToBeDone>. The control value is an OCTET STRING whose value is the BER encoding (section 5.1 [RFC2251]) of
the following sequence:

PartialEntryRequestValue ::= SEQUENCE {
  sizeLimit       [0]     INTEGER(0..maxInt) OPTIONAL,
  timeLimit       [1]     INTEGER(0..maxInt) OPTIONAL }

‘sizelimit’ indicates the maximum number of attribute values of a
single entry that the server should try to return in one
SearchResultEntry message.

‘timeLimit’ indicates the maximum number of seconds since the search
request or since the previous SearchResultEntry message within which
the server should send another searchResultEntry message to the
client.

If one or both the fields are present, then a searchResultEntry will
be sent when any one of the conditions happen.

In the absence of both the limits, the server may use any internally
configured defaults for the sizelimit and timeLimit per entry.

If the sizelimit and timeLimit are set to 0, a partial entry will
not be returned when the processing of the entry is in progress, but
only if the server has processed part of the entry and then failed
to process the remainder. The client can use this to indicate to the
server that it should return the partial entry, if the entry
retrieval fails, and the server could only retrieve a portion of the
entry and not on account of the expiry of a timelimit or sizelimit.
The client can also use this to indicate that the server should set
the PartialEntryResponse control on the SearchResultEntry message
while returning a partial entry. This could be useful in cases where
a dynamically constructed entry spans across servers in a
distributed directory e.g. a dynamic group whose scope spans

6. PartialEntryResponse

This control is included in the SearchResultEntry message as part of
the controls field of the LDAPMessage as defined in [LDAPv3]. The
controlType is set to OID <ToBeDone>. The controlValue is an OCTET
STRING whose value is the BER (section 5.1 [RFC2251]) encoding of
the following sequence:

PartialEntryResponseValue ::= SEQUENCE {
  sequenceNumber    INTEGER(0..maxInt),
  ResultStatus ENUMERATED {
    entryDone    (0),
    entryCont    (1),
    entryIncomplete (2)
  },
  splitAttributes [0]   AttributeDescriptionList OPTIONAL,
  errorMessage    [1]   LDAPString OPTIONAL }

‘sequenceNumber’ indicates the ordering relationship between several
partial entry responses which represent a single directory entry
being returned. The sequenceNumber assigned by the server to partial entries would start from 0 and count up indicating their ordering relationship. This will help the client assort the several partial entries, possibly interspersed with other entries sent as a part of the search response, into a single composite entry.

‘entryDone’ indicates that the entry is completed in this SearchResultEntry message after being sent in multiple SearchResultEntry messages.

‘entryCont’ indicates that a part of the entry is sent in this message and more results are coming.

‘entryIncomplete’ means that the server has sent a part of the entry, but it will not be able to send the remainder due to some error. If the error is fatal, as decided by the server, the server will return the result in the ldapResult. If it is non-fatal the LDAP resultCode will be success, and ResultStatus will be set to ‘entryIncomplete’ in PartialEntryResponseValue. In that case, ‘errorMessage’ may contain the reason for the failure. ‘errorMessage’ can also be used by the server in other failure cases to give a textual message pertinent to the error encountered by the server. However, this should not be used by the client implementers for programmatic decisions.

The ‘splitAttributes’ field MAY be used by the server to indicate the attributes on which the server split the entry, to return partial list of values, when ‘entryCont’ is set in ResultStatus. In the absence of this field, the last attribute type occurring in the SearchResultEntry message is assumed to be the one. The server MUST not return the ‘splitAttributes’ field, if only one attribute was returned in the SearchResultEntry message.

If the complete set of values for the entry has been sent in one SearchResultEntry message, no PartialEntryResponse will be sent.

The server MUST not split in the middle of an attribute value, even if timeLimit is exceeded. The sizeLimit will not exceed in the middle of a value because the sizeLimit is expressed in terms of the number of attribute values. Hence, huge attribute values (like a jpegPhoto value) will not be affected by timeLimit or sizeLimit.

If the server is unable to obtain even one value for an entry within the ‘timeLimit’ interval, it MUST not return any SearchResultEntry to the client at that time, but MUST return a SearchResultEntry as soon as the first value is retrieved.

7. Protocol Examples

7.1 PartialEntryResult control with entryDone and entryCont

This example shows an example of how this control could be used to get a list of members of a groupOfNames split into multiple SearchResultEntry messages. Let’s say this is our groupOfNames entry:

dn: cn=Administrators,dc=example,dc=net
cn: Administrators
member: cn=admin1,dc=example,dc=net
We would set our search base to "cn=Administrators,dc=example,dc=net", filter to "(objectclass=*)", use base scope (to restrict it to this entry) and send the PartialEntryRequest control with sizelimit as 2 along with the search request with the requested attribute as "member". The resulting set would look like this:

```
dn: cn=Administrators,dc=example,dc=net
member: cn=admin1,dc=example,dc=net
member: cn=admin2,dc=example,dc=net
control: { <OID> 0 entryCont }
```

```
dn: cn=Administrators,dc=example,dc=net
member: cn=admin3,dc=example,dc=net
member: cn=admin4,dc=example,dc=net
control: { <OID> 1 entryCont }
```

```
dn: cn=Administrators,dc=example,dc=net
member: cn=admin5,dc=example,dc=net
member: cn=admin6,dc=example,dc=net
control: { <OID> 2 entryCont }
```

```
dn: cn=Administrators,dc=example,dc=net
member: cn=admin7,dc=example,dc=net
member: cn=admin8,dc=example,dc=net
control: { <OID> 3 entryDone }
```

8. Discussion on entry level atomicity

When the attribute values of an LDAP entry are returned in one SearchResultEntry message, the client can assume that the entry information thus returned is atomic, and internally consistent. By breaking an entry into multiple SearchResultEntry messages, it MAY not be guaranteed by a server that the resulting entry information, as returned in multiple SearchResultEntry messages are atomic and internally consistent. The clients should understand this limitation.

An abandon operation may stop the search operation in the middle of a PartialEntry Response, and thus a SearchResultEntry message with an entryDone or entryIncomplete MAY never be sent for an entry. The client should understand this change in semantics.

9. Note to Implementers

Server implementers are advised to have a minimum value for size and time limits per entry based on search transaction heuristics. These limits MAY be configurable. This will allow the server to not honor a request for a very small size or time limit within which a sensible read or search transaction is not possible. An adminLimitExceeded error could be returned in such a case.
Both client and server implementers MUST be aware that using this control could potentially result in a very large set of search results responses per entry.

Client implementers should note that the 'timeLimit' may not be very accurately followed by the LDAP server because of the way in which the search is implemented. The client may also not get the SearchResultEntry messages within the 'timeLimit' interval because of network latencies. The clients should not expect a very accurate implementation of the 'timeLimit' functionality.

10. Interactions with other controls

The PartialEntryRequest control can be used with other controls like the SSS and VLV. While the VLV control splits the results at the entry boundaries, the PartialEntryRequest control will split the entries at value boundaries. So there is no potential conflict between these controls.

11. Security Considerations

The server implementation should be guarded against a malicious client who could hog the server resources by requesting very small values of a large entry like a dynamic group with a huge set of values for the member attribute. By requesting one or a low no of attribute values at a time a client could potentially tie up the LDAP server with a search request for a very long time. Refer section 7 about details of how this can be avoided.

12. References

1 [RFC2026]

2 [RFC2119]
Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997

3 [RFC2251]

4 [DYNGRP]

5 [DUPENT]
Sermersheim, J, "LDAP control for duplicate entry representation
13. Acknowledgement

The authors would like to thank Jim Sermersheim of Novell who reviewed this document and gave his valuable suggestions.

14. Authors’ Addresses

Haripriya S.
Novell Inc.
49/1 & 49/3, Garvebhavi Palya,
7th Mile, Hosur Road,
Bangalore 560068
India
Phone: +91-80-5731858 Extn: 2121
Email: sharipriya@novell.com

Vithalprasad Gaitonde
Novell Inc.
49/1 & 49/3, Garvebhavi Palya,
7th Mile, Hosur Road,
Bangalore 560068
India
Phone: +91-80-5731858 Extn: 2014
Email: gvithalprasad@novell.com