Network Working Group
Request for Comments: 2696
Category: Informational

C. Weider
A. Herron
A. Anantha
Microsoft
T. Howes
Netscape

September 1999

LDAP Control Extension for Simple Paged Results Manipulation

Status of this Memo

Copyright Notice

1. Abstract

This document describes an LDAPv3 control extension for simple paging of search results. This control extension allows a client to control the rate at which an LDAP server returns the results of an LDAP search operation. This control may be useful when the LDAP client has limited resources and may not be able to process the entire result set from a given LDAP query, or when the LDAP client is connected over a low-bandwidth connection. Other operations on the result set are not defined in this extension. This extension is not designed to provide more sophisticated result set management.

The key words "MUST", "SHOULD", and "MAY" used in this document are to be interpreted as described in [bradner97].

2. The Control

This control is included in the searchRequest and searchResultDone messages as part of the controls field of the LDAPMessage, as defined in Section 4.1.12 of [LDAPv3]. The structure of this control is as follows:

```
pagedResultsControl ::= SEQUENCE {
        controlType     1.2.840.113556.1.4.319,
        criticality     BOOLEAN DEFAULT FALSE,
        controlValue    searchControlValue
}
```

The searchControlValue is an OCTET STRING wrapping the BER-encoded version of the following SEQUENCE:

```
realSearchControlValue ::= SEQUENCE {
        size            INTEGER (0..maxInt),
                                -- requested page size from client
                                -- result set size estimate from server
        cookie          OCTET STRING
}
```

3. Client-Server Interaction

An LDAP client application that needs to control the rate at which results are returned MAY specify on the searchRequest a pagedResultsControl with size set to the desired page size and cookie set to the zero-length string. The page size specified MAY be greater than zero and less than the sizeLimit value specified in the searchRequest.

If the page size is greater than or equal to the sizeLimit value, the server should ignore the control as the request can be satisfied in a single page. If the server does not support this control, the server MUST return an error of unsupportedCriticalExtension if the client requested it as critical, otherwise the server SHOULD ignore the control. The remainder of this section assumes the server does not ignore the client's pagedResultsControl.

Each time the server returns a set of results to the client when processing a search request containing the pagedResultsControl, the server includes the pagedResultsControl control in the searchResultDone message. In the control returned to the client, the size MAY be set to the server's estimate of the total number of entries in the entire result set. Servers that cannot provide such an estimate MAY set this size to zero (0). The cookie MUST be set to an empty value if there are no more entries to return (i.e., the page of search results returned was the last), or, if there are more entries to return, to an octet string of the server's choosing,used to resume the search.

The client MUST consider the cookie to be an opaque structure and make no assumptions about its internal organization or value. When the client wants to retrieve more entries for the result set, it MUST

send to the server a searchRequest with all values identical to the initial request with the exception of the messageID, the cookie, and optionally a modified pageSize. The cookie MUST be the octet string on the last searchResultDone response returned by the server. Returning cookies from previous searchResultDone responses besides the last one is undefined, as the server implementation may restrict cookies from being reused.

The server will then return the next set of results from the whole result set. This interaction will continue until the client has retrieved all the results, in which case the cookie in the searchResultDone field will be empty, or until the client abandons the search sequence as described below. Once the paged search sequence has been completed, the cookie is no longer valid and MUST NOT be used.

A sequence of paged search requests is abandoned by the client sending a search request containing a pagedResultsControl with the size set to zero (0) and the cookie set to the last cookie returned by the server. A client MAY use the LDAP Abandon operation to abandon one paged search request in progress, but this is discouraged as it MAY invalidate the client's cookie.

If, for any reason, the server cannot resume a paged search operation for a client, then it SHOULD return the appropriate error in a searchResultDone entry. If this occurs, both client and server should assume the paged result set is closed and no longer resumable.

A client may have any number of outstanding search requests pending, any of which may have used the pagedResultsControl. A server implementation which requires a limit on the number of outstanding paged search requests from a given client MAY either return unwillingToPerform when the client attempts to create a new paged search request, or age out an older result set. If the server implementation ages out an older paged search request, it SHOULD return "unwilling to perform" if the client attempts to resume the paged search that was aged out.

A client may safely assume that all entries that satisfy a given search query are returned once and only once during the set of paged search requests/responses necessary to enumerate the entire result set, unless the result set for that query has changed since the searchRequest starting the request/response sequence was processed. In that case, the client may receive a given entry multiple times and/or may not receive all entries matching the given search criteria.

4. Example

The following example illustrates the client-server interaction between a client doing a search requesting a page size limit of 3. The entire result set returned by the server contains 5 entries.

Lines beginning with "C:" indicate requests sent from client to server. Lines beginning with "S:" indicate responses sent from server to client. Lines beginning with "--" are comments to help explain the example.

```
-- Client sends a search request asking for paged results
-- with a page size of 3.
C: SearchRequest + pagedResultsControl(3,"")
-- Server responds with three entries plus an indication
-- of 5 total entries in the search result and an opaque
-- cooking to be used by the client when retrieving subsequent
-- pages.
S: SearchResultEntry
S: SearchResultEntry
S: SearchResultEntry
S: SearchResultDone + pagedResultsControl(5, "opaque")
-- Client sends an identical search request (except for
-- message id), returning the opaque cooking, asking for
-- the next page.
C: SearchRequest + PagedResultsControl(3, "opaque")
-- Server responds with two entries plus an indication
-- that there are no more entries (null cookie).
S: SearchResultEntry
S: SearchResultEntry
S: SearchResultDone + pagedResultsControl(5,"")
```

5. Relationship to X.500

For LDAP servers providing a front end to X.500 (93) directories, the paged results control defined in this document may be mapped directly onto the X.500 (93) PagedResultsRequest defined in X.511 [x500]. The size parameter may be mapped onto pageSize. The cookie parameter may be mapped onto queryReference. The sortKeys and reverse fields in the X.500 PagedResultsRequest are excluded.

6. Security Considerations

Server implementors should consider the resources used when clients send searches with the simple paged control, to ensure that a client's misuse of this control does not lock out other legitimate operations.

Servers implementations may enforce an overriding sizelimit, to prevent the retrieval of large portions of a publically-accessible directory.

Clients can, using this control, determine how many entries match a particular filter, before the entries are returned to the client. This may require special processing in servers which perform access control checks on entries to determine whether the existence of the entry can be disclosed to the client.

7. References

[LDAPv3] Wahl, M., Howes, T. and S. Kille, "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997.

[Bradner97] Bradner, S., "Key Words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

8. Authors' Addresses

Chris Weider
Microsoft Corp.
1 Microsoft Way
Redmond, WA 98052
USA

Phone: +1 425 882-8080
EMail: cweider@microsoft.com

Andy Herron
Microsoft Corp.
1 Microsoft Way
Redmond, WA 98052
USA

Phone: +1 425 882-8080
EMail: andyhe@microsoft.com

Anoop Anantha
Microsoft Corp.
1 Microsoft Way
Redmond, WA 98052
USA

Phone: +1 425 882-8080
EMail: anoopa@microsoft.com

Tim Howes
Netscape Communications Corp.
501 E. Middlefield Road
Mountain View, CA 94043
USA

Phone: +1 415 937-2600
EMail: howes@netscape.com

9. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.