Network Working Group   A. Herron, Microsoft
INTERNET DRAFT   T. Howes, Netscape
Expire in six months   M. Wahl, Critical Angle Inc
                                                             16 April, 1997

LDAP Control Extension for Server Side Sorting of Search Results
draft-ietf-asid-ldapv3-sorting-00.txt

1. Status of this Memo

2. Abstract

This document describes two LDAPv3 control extensions for server side sorting of search results. These controls allows a client to specify the attribute types and matching rules a server should use when returning the results to an LDAP search request. The controls may be useful when the LDAP client has limited functionality or for some other reason cannot sort the results but still needs them sorted. Other permissible controls on search operations are not defined in this extension.

The sort controls allow a server to return a result code for the sorting of the results that is independent of the result code returned for the search operation.

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

3. The Controls

3.1 Request Control

This control is included in the searchRequest message as part of the controls field of the LDAPMessage, as defined in Section 4.1.12 of [LDAPv3].

The controlType is set to "1.2.840.113556.1.4.473". The criticality MAY be either TRUE or FALSE (where absent is also equivalent to FALSE) at the client's option. The controlValue is an OCTET STRING, whose value is the BER encoding of a value of the following SEQUENCE:

```
     SortKeyList ::= SEQUENCE OF SEQUENCE {
                attributeType   AttributeType,
                orderingRule    [0] MatchingRuleId OPTIONAL,
                reverseOrder    [1] BOOLEAN DEFAULT FALSE }
```

The SortKeyList sequence is in order of highest to lowest sort key precedence.

Each attributeType should only occur in the SortKeyList once. If an attributeType is included in the sort key list multiple times, the server should return an error in the sortResult of unwillingToPerform.

3.2 Response Control

This control is included in the searchResultDone message as part of the controls field of the LDAPMessage, as defined in Section 4.1.12 of [LDAPv3].

The controlType is set to "1.2.840.113556.1.4.474". The criticality is FALSE (MAY be absent). The controlValue is an OCTET STRING, whose value is the BER encoding of a value of the following SEQUENCE:

```
     SortResult ::= SEQUENCE {
        sortResult  ENUMERATED {
            success                   (0), -- results are sorted
            operationsError           (1), -- server internal failure
            timeLimitExceeded         (3), -- timelimit reached before
                                           -- sorting was completed
            strongAuthRequired        (8), -- refused to return sorted
                                           -- results via insecure
                                           -- protocol
            adminLimitExceeded       (11), -- too many matching entries
                                           -- for the server to sort
            noSuchAttribute          (16), -- unrecognized attribute
                                           -- type in sort key
            inappropriateMatching    (18), -- unrecognized or inappro-
                                           -- priate matching rule in
                                           -- sort key
            insufficientAccessRights (50), -- refused to return sorted
                                           -- results to this client
            busy                     (51), -- too busy to process
            unwillingToPerform       (53), -- unable to sort
            other                    (80)
            },
      attributeType [0] AttributeType OPTIONAL }
```

4. Client-Server Interaction

The sortKeyRequestControl specifies one or more attribute types and matching rules for the results returned by a search request. The server SHOULD return all results for the search request in the order specified by the sort keys. If the reverseOrder field is set to TRUE, then the entries will be presented in reverse sorted order for the specified key.

There are six possible scenarios that may occur as a result of the sort control being included on the search request :

1 - **If the server does not support this sorting control and the client** specified TRUE for the control's criticality field, then the server MUST return unavailableCriticalExtension as a return code in the searchResultDone message and not send back any other results. This behavior is specified in section 4.1.12 of [LDAPv3].

2 - **If the server does not support this sorting control and the client** specified FALSE for the control's criticality field, then the server MUST ignore the sort control and process the search request as if it were not present. This behavior is specified in section 4.1.12 of [LDAPv3].

3 - **If the server supports this sorting control but for some reason** cannot sort the search results using the specified sort keys and the client specified TRUE for the control's criticality field, then the server SHOULD do the following: return unavailableCriticalExtension as a return code in the searchResultDone message; include the sortKeyResponseControl in the searchResultDone message, and not send back any search result entries.

4 - **If the server supports this sorting control but for some reason** cannot sort the search results using the specified sort keys and the client specified FALSE for the control's criticality field, then the server should return all search results unsorted and include the sortKeyResponseControl in the searchResultDone message.

5 - **If the server supports this sorting control and can sort the search** results using the specified sort keys, then it should include the sortKeyResponseControl in the searchResultDone message with a sortResult of success.

6 - **If the search request failed for any reason and/or there are no** searchResultEntry messages returned for the search response, then the server SHOULD omit the sortKeyResponseControl from the searchResultDone message.

The client application is assured that the results are sorted in the specified key order if and only if the result code in the sortKeyResponseControl is success. If the server omits the sortKeyResponseControl from the searchResultDone message, the client SHOULD assume that the sort control was ignored by the server.

The sortKeyResponseControl, if included by the server in the searchResultDone message, should have the sortResult set to either success if the results were sorted in accordance with the keys specified in the sortKeyRequestControl or set to the appropriate error code as to why it could not sort the data (such as noSuchAttribute or inappropriateMatching). Optionally, the server MAY set the attributeType to the first attribute type specified in the SortKeyList that was in error. The client SHOULD ignore the attributeType field if the sortResult is success.

The server may not be able to sort the results using the specified sort keys because it may not recognize one of the attribute types, the matching rule associated with an attribute type is not applicable, or none of the attributes in the search response are of these types. Servers may also restrict the number of keys allowed in the control, such as only supporting a single key.

Servers that chain requests to other LDAP servers should ensure that the server satisfying the client's request sort the entire result set prior to sending back the results.

5. Interaction with other search controls

When the sortKeyRequestControl control is included with the pagedResultsControl control as specified in [LdapPaged], then the server should send the searchResultEntry messages sorted according to the sort keys applied to the entire result set. The server should not simply sort each page, as this will give erroneous results to the client.

The sortKeyList must be present on each searchRequest message for the paged result. It also must not change between searchRequests for the same result set. If the server has sorted the data, then it SHOULD send back a sortKeyResponseControl control on every searchResultDone message for each page. This will allow clients to quickly determine if the result set is sorted, rather than waiting to receive the entire result set.

6. Security Considerations

Implementors and administrators should be aware that allowing sorting of results could enable the retrieval of a large number of records from a given directory service, irregardless of administrative limits set on the maximum number of records to return.

A client that desired to pull all records out of a directory service could use a combination of sorting and updating of search filters to retrieve all records in a database in small result sets, thus circumventing administrative limits.

This behavior can be overcome by the judicious use of permissions on the directory entries by the administrator and by intelligent implementations of administrative limits on the number of records retrieved by a client.

7. References

[LDAPv3]
Wahl, M, S. Kille and T. Howes, "Lightweight Directory Access Protocol (v3)", Internet Draft, February, 1997. Available as draft-ietf-asid-ldapv3-protocol-04.txt.

[Bradner97]
Bradner, Scott, "Key Words for use in RFCs to Indicate Requirement Levels", Internet Draft, January, 1997. Available as draft-bradner-key-words-03.txt.

[LdapPaged]
C. Weider, A. Herron, and T. Howes, "LDAP Control Extension for Simple Paged Results Manipulation", Internet Draft, February, 1997. Available as draft-ietf-asid-ldapv3-simplepaged-00.txt.

8. Author's Address

Andy Herron
Microsoft Corp.
1 Microsoft Way
Redmond, WA 98052
USA
andyhe@microsoft.com
+1 206 882-8080

Tim Howes
Netscape Communications Corp.
501 E. Middlefield Road
Mountain View, CA 94043
USA
howes@netscape.com
+1 415 937-2600

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