              IMAP4 Extension to SEARCH Command for Controlling
                     What Kind of Information Is Returned

Status of This Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements. Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

   This document extends IMAP (RFC 3501) SEARCH and UID SEARCH commands
   with several result options, which can control what kind of
   information is returned. The following result options are defined:
   minimal value, maximal value, all found messages, and number of found
   messages.

Table of Contents

1. Introduction

[IMAPABNF] extended SEARCH and UID SEARCH commands with result
specifiers (also known as result options), which can control what
kind of information is returned.

A server advertising the ESEARCH capability supports the following
result options: minimal value, maximal value, all found messages,
and number of found messages. These result options allow clients to
get SEARCH results in more convenient forms, while also saving
bandwidth required to transport the results, for example, by finding
the first unseen message or returning the number of unseen or deleted
messages. Also, when a single MIN or a single MAX result option is
specified, servers can optimize execution of SEARCHes.

2. Conventions Used in This Document

In examples, "C:" and "S:" indicate lines sent by the client and
server, respectively.

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 [KEYWORDS].

3.   IMAP Protocol Changes

3.1. New SEARCH/UID SEARCH Result Options

The SEARCH/UID SEARCH commands are extended to allow for the
following result options:

MIN
    Return the lowest message number/UID that satisfies the SEARCH
    criteria.

    If the SEARCH results in no matches, the server MUST NOT
    include the MIN result option in the ESEARCH response; however,
    it still MUST send the ESEARCH response.

MAX
    Return the highest message number/UID that satisfies the SEARCH
    criteria.

    If the SEARCH results in no matches, the server MUST NOT
    include the MAX result option in the ESEARCH response; however,
    it still MUST send the ESEARCH response.

   ALL
      Return all message numbers/UIDs that satisfy the SEARCH
      criteria. Unlike regular (unextended) SEARCH, the messages are
      always returned using the sequence-set syntax. A sequence-set
      representation may be more compact and can be used as is in a
      subsequent command that accepts sequence-set. Note, the client
      MUST NOT assume that messages/UIDs will be listed in any
      particular order.

      If the SEARCH results in no matches, the server MUST NOT
      include the ALL result option in the ESEARCH response; however,
      it still MUST send the ESEARCH response.

   COUNT
      Return number of the messages that satisfy the SEARCH criteria.
      This result option MUST always be included in the ESEARCH
      response.

If one or more result options described above are specified, the
extended SEARCH command MUST return a single ESEARCH response
[IMAPABNF], instead of the SEARCH response.

An extended UID SEARCH command MUST cause an ESEARCH response with
the UID indicator present.

Note that future extensions to this document can allow servers to
return multiple ESEARCH responses for a single extended SEARCH
command. These extensions will have to describe how results from
multiple ESEARCH responses are to be amalgamated.

If the list of result options is empty, that requests the server to
return an ESEARCH response instead of the SEARCH response. This is
equivalent to "(ALL)".

   Example: C: A282 SEARCH RETURN (MIN COUNT) FLAGGED
                  SINCE 1-Feb-1994 NOT FROM "Smith"
               S: * ESEARCH (TAG "A282") MIN 2 COUNT 3
               S: A282 OK SEARCH completed

   Example: C: A283 SEARCH RETURN () FLAGGED
                  SINCE 1-Feb-1994 NOT FROM "Smith"
               S: * ESEARCH (TAG "A283") ALL 2,10:11
               S: A283 OK SEARCH completed

The following example demonstrates finding the first unseen message
as returned in the UNSEEN response code on a successful SELECT
command:

        Example: C: A284 SEARCH RETURN (MIN) UNSEEN
                    S: * ESEARCH (TAG "A284") MIN 4
                    S: A284 OK SEARCH completed

   The following example demonstrates that if the ESEARCH UID indicator
   is present, all data in the ESEARCH response is referring to UIDs;
   for example, the MIN result specifier will be followed by a UID.

        Example: C: A285 UID SEARCH RETURN (MIN MAX) 1:5000
                    S: * ESEARCH (TAG "A285") UID MIN 7 MAX 3800
                    S: A285 OK SEARCH completed

   The following example demonstrates returning the number of deleted
   messages:

        Example: C: A286 SEARCH RETURN (COUNT) DELETED
                    S: * ESEARCH (TAG "A286") COUNT 15
                    S: A286 OK SEARCH completed

3.2. Interaction with CONDSTORE extension

   When the server supports both the ESEARCH and the CONDSTORE
   [CONDSTORE] extension, and the client requests one or more result
   option described in section 3.1 together with the MODSEQ search
   criterion in the same SEARCH/UID SEARCH command, then the server MUST
   return the ESEARCH response containing the MODSEQ result option
   (described in the following paragraph) instead of the extended SEARCH
   response described in section 3.5 of [CONDSTORE].

   If the SEARCH/UID SEARCH command contained a single MIN or MAX result
   option, the MODSEQ result option contains the mod-sequence for the
   found message. If the SEARCH/UID SEARCH command contained both MIN
   and MAX result options and no ALL/COUNT option, the MODSEQ result
   option contains the highest mod-sequence for the two returned
   messages. Otherwise the MODSEQ result option contains the highest
   mod-sequence for all messages being returned.

   Example: The following example demonstrates how Example 15 from
   [CONDSTORE] would look in the presence of one or more result option:

        C: a1 SEARCH RETURN (MIN) MODSEQ "/flags/\\draft"
           all 620162338
        S: * ESEARCH (TAG "a1") MIN 2 MODSEQ 917162488
        S: a1 OK Search complete

        C: a2 SEARCH RETURN (MAX) MODSEQ "/flags/\\draft"
           all 620162338
        S: * ESEARCH (TAG "a2") MAX 23 MODSEQ 907162321

```
        S: a2 OK Search complete

        C: a3 SEARCH RETURN (MIN MAX) MODSEQ "/flags/\\draft"
            all 620162338
        S: * ESEARCH (TAG "a3") MIN 2 MAX 23 MODSEQ 917162488
        S: a3 OK Search complete

        C: a4 SEARCH RETURN (MIN COUNT) MODSEQ "/flags/\\draft"
            all 620162338
        S: * ESEARCH (TAG "a4") MIN 2 COUNT 10 MODSEQ 917162500
        S: a4 OK Search complete
```

4. Formal Syntax

   The following syntax specification uses the Augmented Backus-Naur
   Form (ABNF) notation as specified in [ABNF].

   Non-terminals referenced but not defined below are as defined by
   [IMAP4], [CONDSTORE], or [IMAPABNF].

   Except as noted otherwise, all alphabetic characters are case-
   insensitive. The use of upper or lowercase characters to define
   token strings is for editorial clarity only. Implementations MUST
   accept these strings in a case-insensitive fashion.

```
      capability         =/ "ESEARCH"

      search-return-data = "MIN" SP nz-number /
                           "MAX" SP nz-number /
                           "ALL" SP sequence-set /
                           "COUNT" SP number
                           ;; conforms to the generic
                           ;; search-return-data syntax defined
                           ;; in [IMAPABNF]

      search-return-opt  = "MIN" / "MAX" / "ALL" / "COUNT"
                           ;; conforms to generic search-return-opt
                           ;; syntax defined in [IMAPABNF]
```

   When the CONDSTORE [CONDSTORE] IMAP extension is also supported,
   the ABNF is updated as follows:

```
      search-return-data =/ "MODSEQ" SP mod-sequence-value
                           ;; mod-sequence-value is defined
                           ;; in [CONDSTORE]
```

5. Security Considerations

   In the general case, the IMAP SEARCH/UID SEARCH commands can be CPU
   and/or IO intensive, and are seen by some as a potential attack point
   for denial of service attacks, so some sites/implementations even
   disable them entirely. This is quite unfortunate, as SEARCH command
   is one of the best examples demonstrating IMAP advantage over POP3.

   The ALL and COUNT return options don't change how SEARCH is working
   internally; they only change how information about found messages is
   returned. MIN and MAX SEARCH result options described in this
   document can lighten the load on IMAP servers that choose to optimize
   SEARCHes containing only one or both of them.

   It is believed that this extension doesn't raise any additional
   security concerns not already discussed in [IMAP4].

6. IANA Considerations

   IMAP4 capabilities are registered by publishing a standards track RFC
   or an IESG-approved experimental RFC. The registry is currently
   located at <http://www.iana.org/assignments/imap4-capabilities>.

   This document defines the ESEARCH IMAP capability, which IANA added
   to the registry.

7. Normative References

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

   [IMAP4]   Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION
                4rev1", RFC 3501, March 2003.

   [ABNF]   Crocker, D. (Ed.) and P. Overell , "Augmented BNF for
                Syntax Specifications: ABNF", RFC 4234, October 2005.

   [IMAPABNF]   Melnikov, A. and C. Daboo, "Collected Extensions to IMAP4
                ABNF", RFC 4466, April 2006..

   [CONDSTORE] Melnikov, A. and S. Hole, "IMAP Extension for Conditional
                STORE", RFC 4551, June 2006.

8. Acknowledgments

   Thanks to Michael Wener, Arnt Gulbrandsen, Cyrus Daboo, Mark Crispin,
   and Pete Maclean for comments and corrections.

Authors' Addresses

    Alexey Melnikov
    Isode Limited
    5 Castle Business Village
    36 Station Road
    Hampton, Middlesex, TW12 2BX
    UK

    EMail: Alexey.Melnikov@isode.com

    Dave A. Cridland
    Inventure Systems Limited

    EMail: dave.cridland@inventuresystems.co.uk
    URL: http://invsys.co.uk/dave/

Full Copyright Statement

Intellectual Property

   The IETF takes no position regarding the validity or scope of any
   Intellectual Property Rights or other rights that might be claimed to
   pertain to the implementation or use of the technology described in
   this document or the extent to which any license under such rights
   might or might not be available; nor does it represent that it has
   made any independent effort to identify any such rights. Information
   on the procedures with respect to rights in RFC documents can be
   found in BCP 78 and BCP 79.

   Copies of IPR disclosures made to the IETF Secretariat and any
   assurances of licenses to be made available, or the result of an
   attempt made to obtain a general license or permission for the use of
   such proprietary rights by implementers or users of this
   specification can be obtained from the IETF on-line IPR repository at
   http://www.ietf.org/ipr.

   The IETF invites any interested party to bring to its attention any
   copyrights, patents or patent applications, or other proprietary
   rights that may cover technology that may be required to implement
   this standard. Please address the information to the IETF at
   ietf-ipr@ietf.org.

Acknowledgement