Network Working Group M. Crispin
INTERNET-DRAFT: IMAP SORT University of Washington
Document: internet-drafts/draft-ietf-imapext-sort-00.txt February 2000

INTERNET MESSAGE ACCESS PROTOCOL - SORT EXTENSION

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC 2026.

This document is an Internet-Draft. 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.

A revised version of this document will be submitted to the RFC editor as an Informational Document for the Internet Community.

A revised version of this draft document, describing an expanded version of this protocol extension, will be submitted to the RFC editor as a Proposed Standard for the Internet Community.

Discussion and suggestions for improvement are requested, and should be sent to ietf-imapext@IMC.ORG. This document will expire before 5 July 2000. Distribution of this memo is unlimited.

Abstract

This document describes an experimental server-based sorting extension to the IMAP4rev1 protocol, as implemented by the University of Washington's IMAP toolkit. This extension provides substantial performance improvements for IMAP clients which offer sorted views.

A server which supports this extension indicates this with a capability name of "SORT". Client implementations SHOULD accept any

capability name which begins with "SORT" as indicating support for the extension described in this document. This provides for future upwards-compatible extensions.

At the time of this document was written, the IMAP Extensions Working Group (IETF-IMAPEXT) was considering upwards-compatible additions to the SORT extension described in this document, tenatively called the SORT2 extension.

Extracted Subject Text

The "SUBJECT" SORT criteria uses a version of the subject which has specific subject artifacts of deployed Internet mail software removed. Due to the complexity of these artifacts, the above syntax is ambiguous. The following procedure is followed to determing the actual "base subject" which is used to sort by subject:

(1) Remove all trailing text of the subject that matches the subj-trailer ABNF, repeat until no more matches are possible.

(2) Remove all prefix text of the subject that matches subj-leader.

(3) If there is prefix text of the subject that matches subj-blob, and removing that prefix leaves a non-empty subj-base, then remove the prefix text.

(4) Repeat (2) and (3) until no matches remain.

(5) Convert any RFC 2047 encoded-words in the remaining subj-base to UTF-8.

(6) The resulting text is the "base subject" used in the SORT.

All servers and disconnected clients MUST use exactly this algorithm when sorting by subject. Otherwise there is potential for a user to get inconsistant results based on whether they are running in connected or disconnected IMAP mode.

Additional Commands

This command is an extension to the IMAP4rev1 base protocol.

The section header is intended to correspond with where it would be located in the main document if it was part of the base specification.

6.3.SORT. SORT Command

Arguments: sort program
charset specification
searching criteria (one or more)

Data: untagged responses: SORT

Result: OK - sort completed
NO - sort error: can't sort that charset or criteria
BAD - command unknown or arguments invalid

The SORT command is a variant of SEARCH with sorting semantics for the results. Sort has two arguments before the searching criteria argument; a parenthesized list of sort criteria, and the searching charset.

Note that unlike SEARCH, the searching charset argument is mandatory. The US-ASCII and UTF-8 charsets MUST be implemented. All other charsets are optional.

There is also a UID SORT command which corresponds to SORT the way that UID SEARCH corresponds to SEARCH.

The SORT command first searches the mailbox for messages that match the given searching criteria using the charset argument for the interpretation of strings in the searching criteria. It then returns the matching messages in an untagged SORT response, sorted according to one or more sort criteria.

If two or more messages exactly match according to the sorting criteria, these messages are sorted according to the order in which they appear in the mailbox. In other words, there is an implicit sort criterion of "sequence number".

When multiple sort criteria are specified, the result is sorted in the priority order that the criteria appear. For example, (SUBJECT DATE) will sort messages in order by their subject text;

and for messages with the same subject text will sort by their sent date.

Untagged EXPUNGE responses are not permitted while the server is responding to a SORT command, but are permitted during a UID SORT command.

The defined sort criteria are as follows. Refer to the Formal Syntax section for the precise syntactic definitions of the arguments.

ARRIVAL
Internal date and time of the message. This differs from the ON criteria in SEARCH, which uses just the internal date.

CC
Mailbox part of the first "cc" address.

DATE
Sent date and time from the Date: header, adjusted by time zone. This differs from the SENTON criteria in SEARCH, which uses just the date and not the time, nor adjusts by time zone.

FROM
Mailbox part of the "From" address.

REVERSE
Followed by another sort criterion, has the effect of that criterion but in reverse order.

SIZE
Size of the message in octets.

SUBJECT
Extracted subject text.

TO
Mailbox part of the first "To" address.

```
Example:    C: A282 SORT (SUBJECT) UTF-8 SINCE 1-Feb-1994
            S: * SORT 2 84 882
            S: A282 OK SORT completed
            C: A283 SORT (SUBJECT REVERSE DATE) UTF-8 ALL
            S: * SORT 5 3 4 1 2
            S: A283 OK SORT completed
            C: A284 SORT (SUBJECT) US-ASCII TEXT "not in mailbox"
            S: A284 OK SORT completed
```

Additional Responses

This response is an extension to the IMAP4rev1 base protocol.

The section heading of this response is intended to correspond with where it would be located in the main document.

7.2.SORT. SORT Response

Data: one or more numbers

The SORT response occurs as a result of a SORT or UID SORT command. The number(s) refer to those messages that match the search criteria. For SORT, these are message sequence numbers; for UID SORT, these are unique identifiers. Each number is delimited by a space.

Example: S: * SORT 2 3 6

Formal Syntax of SORT commands and Responses

```
   sort-data       = "SORT" *(SP nz-number)

   sort            = ["UID" SPACE] "SORT" SP
                     "(" sort-criterion *(SP sort-criterion) ")"
                     SP search_charset 1*(SP search_key)

   sort-criterion  = ["REVERSE" SPACE] sort-key

   sort-key        = "ARRIVAL" / "CC" / "DATE" / "FROM" / "SIZE" /
                     "SUBJECT" / "TO"
```

The following syntax describes subject extraction rules (1)-(5):

```
   subject         = *(subj-leader / subj-blob) subj-base *subj-trailer

   subj-repeat     = "[" 1*DIGIT "]"

   subj-refwd      = ("re" / ("fw" ["d"])) [subj-repeat] ":"

   subj-leader     = subj-refwd / WSP

   subj-blob       = "[" 1*BLOBCHAR "]" *WSP

   subj-trailer    = "(fwd)" / WSP

   subj-base       = NONWSP *([WSP] NONWSP)

   BLOBCHAR        = %x01-5c / %x5e-7f
                     ; any CHAR except ']'

   NONWSP          = %x01-08 / %x0a-1f / %x21-7f
                     ; any CHAR other than WSP
```

Security Considerations

Security issues are not discussed in this memo.

Internationalization Considerations

By default, strings are sorted according to the "minimum sorting collation algorithm". All implementations of SORT MUST implement the minimum sorting collation algorithm.

In the minimum sorting collation algorithm, the Basic Latin alphabetics (U+0041 to U+005A uppercase, U+0061 to U+007A lowercase) are sorted in a case-insensitive fashion; that is, "A" (U+0041) and "a" (U+0061) are treated as exact equals. All other characters are sorted according to their octet values, as expressed in UTF-8. No attempt is made to treat composed characters specially, or to do case-insensitive comparisons of composed characters.

> Note: this means, among other things, that the composed characters in the Latin-1 Supplement are not compared in what would be considered an ISO 8859-1 "case-insensitive" fashion. Case comparison rules for characters with diacriticals differ between languages; the minimum sorting collation does not attempt to deal with this at all. This is reserved for other sorting collations, which may be language-specific.

Other sorting collations, and the ability to change the sorting collation, will be defined in a separate document dealing with IMAP internationalization.

It is anticipated that there will be a generic Unicode sorting collation, which will provide generic case-insensitivity for alphabetic scripts, specification of composed character handling, and language-specific sorting collations. A server which implements non-default sorting collations will modify its sorting behavior according to the selected sorting collation.

Non-English translations of "Re" or "Fw"/"Fwd" are not specified for removal in the extracted subject text process. By specifying that only the English forms of the prefixes are used, it becomes a simple display time task to localize the prefix language for the user. If, on the other hand, prefixes in multiple languages are permitted, the result is a geometrically complex, and ultimately unimplementable, task. In order to improve the ability to support non-English display in Internet mail clients, only the English form of these prefixes should be transmitted in Internet mail messages.

Author's Address

Mark R. Crispin
Networks and Distributed Computing
University of Washington
4545 15th Aveneue NE
Seattle, WA 98105-4527

Phone: (206) 543-5762

EMail: MRC@CAC.Washington.EDU