

INTERNET MESSAGE ACCESS PROTOCOL - THREAD 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>.

To learn the current status of any Internet-Draft, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ds.internic.net (US East Coast), nic.nordu.net (Europe), ftp.isi.edu (US West Coast), or munnari.oz.au (Pacific Rim).

A revised version of this draft document 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 the server-based threading extension to the IMAP4rev1 protocol. This extension provides substantial performance

improvements for IMAP clients which offer threaded views.

A server which supports this extension indicates this with more or more capability names consisting of "THREAD-" followed by a supported threading algorithm name as described in this document. This provides for future upwards-compatible extensions.

Extracted Subject Text

Threading algorithms use 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 determining the actual "base 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 THREAD.

All servers and disconnected clients MUST use exactly this algorithm when threading. Otherwise there is potential for a user to get inconsistent 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.THREAD. THREAD Command

Arguments: threading algorithm
 charset specification
 searching criteria (one or more)

Data: untagged responses: THREAD

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

The THREAD command is a variant of SEARCH with threading semantics for the results. Thread has two arguments before the searching criteria argument; a threading algorithm, and the searching charset. Note that unlike SEARCH, the searching charset argument is mandatory.

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

The THREAD 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 THREAD response, threaded according to the specified threading algorithm. Unlike SEARCH, if no messages match the searching criteria in a THREAD command, no untagged THREAD response is returned.

The defined threading algorithms are as follows:

ORDEREDSUBJECT

The ORDEREDSUBJECT threading algorithm is also referred to as "poor man's threading." The searched messages are sorted by subject and then by sent date, equivalent to a "SORT (SUBJECT DATE)". The messages are then split into separate threads, with each thread containing messages with the same extracted subject text. Finally, the threads are sorted by the sent date

of the first message in the thread.

Example: C: A283 THREAD ORDEREDSUBJECT UTF-8 SINCE 5-AUG-1999
S: * THREAD (146 151)(144 145)(155)(147 148 152
167)(182)(181)(149)(154)(153)(164 170)(156)
(158 161 162)(157 160 163)(159)(183 185)(165)
(166)(168)(169)(171)(172)(173)(186)(174)(150)
(175)(176)(177 179 184)(178)(180)(187)
S: A283 OK THREAD completed
C: A284 THREAD ORDEREDSUBJECT US-ASCII TEXT "gewp"
S: A284 OK THREAD completed

Note: The line breaks in the first client response are for editorial clarity and do not appear in a real THREAD response.

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.THREAD. THREAD Response

Data: one or more threads

The THREAD response occurs as a result of a THREAD or UID THREAD command. It contains one or more threads. A thread consists of a parenthesized list of thread members. Thread members consist of one or more message numbers until the thread splits into multiple sub-threads, at which point the thread nests into multiple sub-threads. There is no limit to the nesting of threads.

The messages numbers refer to those messages that match the search criteria. For THREAD, these are message sequence numbers; for UID THREAD, these are unique identifiers.

Example: S: * THREAD (2)(3 6 (4 23)(44 7 96))

In this example, there are two threads. The first thread consists only of message 2. The second thread consists of the messages 3 and 6, after which it splits into two subthreads; the first of which contains messages 4 and 23, and the second of which contains messages 44, 7, and 96.

```

2
    , - 4 - 23
3 - 6 -<
    ` - 44 - 7 - 96

```


Formal Syntax of THREAD commands and Responses

```
thread-data      = "THREAD" SPACE 1*thread-list
thread-list      = "(" thread-members / thread-nested ")"
thread-members   = nz-number *(SP nz-number) [SP thread-nested]
thread-nested    = 2*thread-list
thread           = ["UID" SPACE] "THREAD" SP thread-algorithm
                  SP search-charset 1*(SP search-key)
thread-algorithm = "ORDEREDSUBJECT" / atom
```

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 threaded according to the "minimum sorting collation algorithm". All implementations of THREAD MUST implement

the minimum sorting collation algorithm.

In the minimum sorting collation algorithm, the 26 Latin alphabetic characters are sorted in a case-insensitive fashion; that is, "A" and "a" 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.

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 Avenue NE
Seattle, WA 98105-4527

Phone: (206) 543-5762

E-Mail: MRC@CAC.Washington.EDU

