

The IMAP SEARCH=INTHREAD and THREAD=REFS Extensions  
[draft-ietf-morg-inthread-01.txt](#)

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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>.

This Internet-Draft expires in January 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Simplified BSD License.

## Abstract

The SEARCH=INTHREAD extension extends the IMAP SEARCH command to operate on threads as well as individual messages. Other commands which search are implicitly extended. This allows clients to perform searches such as "find all threads that mention schemata", rather than just "find all single messages that mention schemata".

The THREAD=REFS extension provides a threading algorithm using (almost) only the References header field for use with the IMAP THREAD command.

## 1. Conventions Used in This Document

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 [[RFC2119](#)].

Formal syntax is defined by [[RFC5234](#)].

Example lines prefaced by "C:" are sent by the client and ones prefaced by "S:" by the server. The five characters [...] means that something has been elided.

## 2. Overview

This document defines two related extensions.

The THREAD=REFS extension defined a fairly pure References-based (see [[RFC5322](#)] [section 3.6.4](#)) threading algorithm for use with the THREAD command (see [[RFC5256](#)]) and with SEARCH=INTHREAD.

An IMAP server (see [[RFC3501](#)]) that supports the THREAD=REFS extension MUST announce THREAD=REFS as capabilities. The THREAD=REFS extension adds no new commands and responses, only a new thread algorithm.

The SEARCH=INTHREAD extension extends the IMAP SEARCH command to operate on threads as well as individual messages. Other commands which search are implicitly extended. SEARCH=INTHREAD requires that servers implement THREAD=REFS too.

An IMAP server that supports SEARCH=INTHREAD announces both SEARCH=INTHREAD and THREAD=REFS as capabilities. The SEARCH=INTHREAD extension adds no new commands and responses, but adds two new search-keys, INTHREAD and MESSAGEID, and one search return option,



THREAD=REFS.

### **3. New Search Keys etc.**

This document defines a new search key which finds all messages in a thread where at least one message matches another (specified) search key, and a helper which matches a message given its message-id.

#### **3.1. The INTHREAD Search Key**

INTHREAD takes one argument, which is another search key.

If the argument matches a message, then INTHREAD matches all the message in the same thread as that message.

This command finds all messages in an entire thread concerning the meetings where fizzle was discussed:

```
C: a UID SEARCH INTHREAD (SUBJECT meeting BODY fizzle)
```

This command threads all threads containing at least one message from fred@example.com:

```
C: a UID THREAD REFS utf-8 INTHREAD FROM <fred@example.com>
```

#### **3.2. The MESSAGEID Search Key**

The MESSAGEID search key takes a single argument, and matches a message if that message's message-id is the same as the argument.

This command finds all in the same thread as <432.123.321@example.com>:

```
C: a UID SEARCH INTHREAD MESSAGEID
   "<4321.1234.321@example.com>"
```

Note that in the past, some message-ids contained needless quoting. Up to around 2001, a message might have the following Message-ID:

```
Message-ID: <"432.123.321"@example.com>
```

A reply might remove the unnecessary quotes:

```
References: <432.123.321@example.com>
```



Although such message-id quoting is still permitted, the author of this document has not found evidence of such quoting since 2001. This document does not make any recommendation about how to handle quoted left-hand-sides.

### **3.3. The THREAD=\* Search Return Option(s)**

The THREAD=\* search return options enables the client to select which threading algorithm the server uses when processing INTHREAD as part of a SEARCH command. If THREAD=\* isn't specified, then the default for the SEARCH command is THREAD=REFS.

When the server processes a THREAD command, it uses the algorithm specified by the client.

I can't think of a good example (or use case) for a non-default use of this.

## **4. The THREAD=REFS Thread Algorithm**

The THREAD=REFS thread algorithm is defined as the part of THREAD=REFERENCES (see [[RFC5256](#)]) which concerns itself with the References, In-Reply-To and Message-ID fields. THREAD=REFS ignores Subject.

THREAD=REFS sorts threads by the most recent INTERNALDATE in each thread, replacing THREAD=REFERENCES step (4). This means that when a new message arrives, its thread becomes the latest thread. (Note that while threads are sorted by arrival date, messages within a thread are sorted by sent date, just as for THREAD=REFERENCES.)

This document defines THREAD=REFS because THREAD=REFERENCES is too inclusive for the INTHREAD search key. For example, independent threads that happen to have the same subject field (such as "Agenda for Friday's meeting", "Web site updated" or "Message delivery failed") are grouped into one thread by THREAD=REFERENCES.

It is explicitly permitted for the server to persistently store threading information, even if this causes the server to return different information than it would otherwise. This can happen if the first messages in a thread are deleted, for example.

## **5. Formal Syntax**

The following syntax specification uses the Augmented Backus-Naur



Form (ABNF) notation as specified in [RFC5234]. [RFC3501] defines the non-terminals "capability", "search-key" and "string", [RFC4466] defines "search-return-opt", [RFC5256] defines "thread-alg".

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

```
capability    =/ "SEARCH=INTHREAD" / "THREAD=REFS"

search-key    =/ "INTHREAD" SP search-key / "MESSAGEID" SP string

thread-alg    =/ "REFS"

search-return-opt =/ "THREAD=" thread-alg
```

## 6. Security Considerations

This document is believed not to have any security implications.

## 7. IANA Considerations

The IANA is requested to add SEARCH=INTHREAD and THREAD=REFS to the list of IMAP extensions,  
<http://www.iana.org/assignments/imap4-capabilities>.

## 8. Acknowledgements

The name THREAD=REFS was suggested by Cyrus Daboo. Dave Cridland, Alexey Menikov and particularly Timo Sirainen have helped with the document.

## 9. Normative References

- [RFC2119] Bradner, "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), Harvard University, March 1997.
- [RFC3501] Crispin, "Internet Message Access Protocol - Version 4rev1", [RFC 3501](#), University of Washington, June 2003.
- [RFC4466] Melnikov, Daboo, "Collected Extensions to IMAP4 ABNF", [RFC 4466](#), Isode Ltd., April 2006.





- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", [RFC 5234](#), January 2008.
- [RFC5256] Crispin, Murchison, "INTERNET MESSAGE ACCESS PROTOCOL - SORT AND THREAD EXTENSIONS", [RFC 5256](#), Panda Programming, June 2008.
- [RFC5322] Resnick, "Internet Message Format", [RFC 5322](#), Qualcomm, October 2008.

#### **10. Author's Address**

Arnt Gulbrandsen  
Schweppermannstr. 8  
D-81671 Muenchen  
Germany

Email: [arnt@gulbrandsen.priv.no](mailto:arnt@gulbrandsen.priv.no)



(RFC Editor: Please delete everything after this point)

## **11. Open Issues**

I removed THREADROOT and THREADLEAF. Put them back in? I didn't actually implement them (it's now 14 months since I implemented INTTHREAD), so I dropped them from the draft now, on the theory that they're insufficiently justified.

I left the THREAD= search return option, but I haven't implemented it either. I don't see any use cases for anything other than the default. (At the moment there isn't an example, which is really another way of saying "no use cases".) I want to drop it.

It's not clear to me that the MESSAGEID search-key is worth the bother. HEADER Message-Id "<foo@bar>" searches for a substring and MESSAGEID "<foo@bar>" for a complete message-id. In a sample of a half-million recent messages, none of the message-ids contained embedded greater-than or less-than signs, so it's hard to imagine any practical effects of treating the two searches identically.

## **12. Changes since -00**

- The IANA asked me to specify the IANA registry exactly
- Boilerplate updates - IETF Trust and so on.

### Changes since -01

- Added THREADROOT, THREADLEAF and MESSAGEID
- Fixed the typo

### Changes since -02

- Specified thread algorithm per-command, generally using a search return option.
- Defined THREADROOT and THREADLEAF robustly.
- Required that the server implement THREAD=REFS if it implements SEARCH=INTTHREAD.
- Use In-Reply-To as THREAD=REFERENCES, since Timo prefers that and I don't mind.
- Use Date as T=R does. Hm? Good idea?



Changes since -03

- Boilerplate updates for 5377 and blah

Changes since -03

- Sort threads by the most recent date in each thread, so that new messages arriving makes a thread new again.

Changes since d-g-i-i-04

- Define "most recent thread" by arrival date, not sender date. Suits typical client use better. When reading a thread, you want to read messages as ordered by the senders, but the most recent thread should be the one which arrived in your mailbox most recently.
- Rename to be a WG draft.

Changes since -00

- I had a bit of an SCM disaster with my laptop, desktop and git. I hope I resolved it well. My apologies if dropped something.
- Better wording for the INTHREAD search key
- Message-id turned into an IMAP string.
- Elaborated a little on message-id equality. The text leaves it up to the server whether it will normalize. (The last program to generate quotes in message-ids was, AFAICT, procmail/formail, and it quit around 2001. The changelog does not say why. I asked Philip and he can't remember either. Up to 2001 some respondents normalized procmail's message-ids and some (most?) didn't.)
- Remove THREADROOT and THREADLEAF. Note my desire to remove THREAD= and perhaps MESSAGEID.

