

---

Stream: Internet Engineering Task Force (IETF)  
RFC: [9394](#)  
Updates: [4731](#), [5267](#)  
Category: Standards Track  
Published: June 2023  
ISSN: 2070-1721  
Authors: A. Melnikov    A. P. Achuthan    V. Nagulakonda    L. Alves  
              Isode                      Yahoo!                      Yahoo!

# RFC 9394

## IMAP PARTIAL Extension for Paged SEARCH and FETCH

---

### Abstract

The PARTIAL extension of the Internet Message Access Protocol (see RFCs 3501 and 9051) allows clients to limit the number of SEARCH results returned, as well as to perform incremental (paged) searches. This also helps servers to optimize resource usage when performing searches.

This document extends the PARTIAL SEARCH return option originally specified in RFC 5267. It also clarifies some interactions between RFC 5267 and RFCs 4731 and 9051.

This document updates RFCs 4731 and 5267.

### Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc9394>.

### Copyright Notice

Copyright (c) 2023 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

## Table of Contents

1. Introduction and Overview	3
2. Document Conventions	3
3. The PARTIAL Extension	3
3.1. Incremental SEARCH and Partial Results	3
3.2. Interaction between PARTIAL, MIN, MAX, and SAVE SEARCH Return Options	5
3.3. Extension to UID FETCH	6
3.4. Use of "PARTIAL" and "CONDSTORE" IMAP Extensions Together	6
4. Formal Syntax	7
5. Security Considerations	8
6. IANA Considerations	9
6.1. Changes/Additions to the IMAP Capabilities Registry	9
7. References	9
7.1. Normative References	9
7.2. Informative References	10
Acknowledgments	10
Authors' Addresses	10

## 1. Introduction and Overview

This document defines an extension to the Internet Message Access Protocol [RFC3501] [RFC9051] for performing incremental searches and fetches. This extension is compatible with both IMAP4rev1 [RFC3501] and IMAP4rev2 [RFC9051]. This extension uses IMAP extensibility rules defined in [RFC4466].

The PARTIAL extension of the Internet Message Access Protocol allows clients to limit the number of SEARCH results returned, as well as to perform incremental (paged) searches. This also helps servers to optimize resource usage when performing searches.

This document extends the PARTIAL SEARCH return option originally specified in RFC 5267. It also clarifies some interactions between RFC 5267 and RFCs 4731 and 9051.

## 2. Document Conventions

In protocol examples, this document uses a prefix of "C: " to denote lines sent by the client to the server and "S: " for lines sent by the server to the client. Lines prefixed with "// " are comments explaining the previous protocol line. These prefixes and comments are not part of the protocol. Lines without any of these prefixes are continuations of the previous line, and no line breaks are present in the protocol unless specifically mentioned.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Other capitalized words are IMAP key words [RFC3501] [RFC9051] or key words from this document.

## 3. The PARTIAL Extension

An IMAP server advertises support for the PARTIAL extension by including the "PARTIAL" capability in the CAPABILITY response / response code.

### 3.1. Incremental SEARCH and Partial Results

The PARTIAL SEARCH return option causes the server to provide in an ESEARCH response [RFC4731] [RFC9051] a subset of the results denoted by the sequence range given as the mandatory argument. The first result (message with the lowest matching Unique Identifier (UID)) is 1; thus, the first 500 results would be obtained by a return option of "PARTIAL 1:500" and the second 500 by "PARTIAL 501:1000". This intentionally mirrors message sequence numbers.

It is also possible to direct the server to start the SEARCH from the latest matching (with the highest UID) message. This can be done by prepending "-" to the index. For example, -1 is the last message, -2 is next to the last, and so on. Using this syntax helps server implementations to optimize their SEARCHes.

A single command **MUST NOT** contain more than one PARTIAL or ALL search return option; that is, either one PARTIAL, one ALL, or neither PARTIAL nor ALL is allowed.

For SEARCH results, the entire list of results **MUST** be ordered in mailbox order -- that is, in UID or message sequence number order.

In cases where a PARTIAL SEARCH return option references results that do not exist by using a range that starts or ends higher (or lower) than the current number of results, the server returns the results that are in the set. This yields a PARTIAL return data item that has, as payload, the original range and a potentially missing set of results that may be shorter than the extent of the range. If the whole range references results that do not exist, a special value "NIL" is returned by the server instead of the sequence set.

Clients need not request PARTIAL results in any particular order. Because mailboxes may change, clients might wish to use PARTIAL in combination with UPDATE (see [RFC5267](#)) if the server also advertises the "CONTEXT=SEARCH" capability, especially if the intent is to walk a large set of results; however, these return options do not interact -- the UPDATE will provide notifications for all matching results.

```
// Let's assume that the A01 SEARCH without PARTIAL would return
// 23764 results.
C: A01 UID SEARCH RETURN (PARTIAL -1:-100) UNDELETED
  UNKEYWORD $Junk
S: * ESEARCH (TAG "A01") UID PARTIAL (-1:-100 ...)
// 100 most recent results in set syntax elided.
S: A01 OK Completed.
```

```
// Let's assume that the A02 SEARCH without PARTIAL would return
// 23764 results.
C: A02 UID SEARCH RETURN (PARTIAL 23500:24000) UNDELETED
  UNKEYWORD $Junk
C: A03 UID SEARCH RETURN (PARTIAL 1:500) UNDELETED
  UNKEYWORD $Junk
C: A04 UID SEARCH RETURN (PARTIAL 24000:24500) UNDELETED
  UNKEYWORD $Junk
S: * ESEARCH (TAG "A02") UID PARTIAL (23500:24000 ...)
// 264 results in set syntax elided;
// this spans the end of the results.
S: A02 OK Completed.
S: * ESEARCH (TAG "A03") UID PARTIAL (1:500 ...)
// 500 results in set syntax elided.
S: A03 OK Completed.
S: * ESEARCH (TAG "A04") UID PARTIAL (24000:24500 NIL)
// No results are present; this is beyond the end of the results.
S: A04 OK Completed.
```

### 3.2. Interaction between PARTIAL, MIN, MAX, and SAVE SEARCH Return Options

This section only applies if the server advertises the "PARTIAL" IMAP capability or "CONTEXT=SEARCH" [RFC5267], together with "ESEARCH" [RFC4731] and/or IMAP4rev2 [RFC9051].

The SAVE result option doesn't change whether the server would return items corresponding to PARTIAL SEARCH result options.

As specified in Section 3.1, it is an error to specify both the PARTIAL and ALL result options in the same SEARCH command.

When the SAVE result option is combined with the PARTIAL result option and none of the MIN/MAX/COUNT result options are present, the corresponding PARTIAL is returned, and the "\$" marker would contain references to all messages returned by the PARTIAL result option.

When the SAVE and PARTIAL result options are combined with the MIN or MAX result option and the COUNT result option is absent, the corresponding PARTIAL result and MIN/MAX are returned (if the SEARCH result is not empty), and the "\$" marker would contain references to all messages returned by the PARTIAL result option together with the corresponding MIN/MAX message.

If the SAVE and PARTIAL result options are combined with both the MIN and MAX result options and the COUNT result option is absent, the PARTIAL, MIN, and MAX result options are returned (if the SEARCH result is not empty), and the "\$" marker would contain references to all messages returned by the PARTIAL result option together with the MIN and MAX messages.

If the SAVE and PARTIAL result options are combined with the COUNT result option, the PARTIAL and COUNT result options are returned, and the "\$" marker would always contain references to all messages found by the SEARCH or UID SEARCH command.

Table 1 summarizes additional requirements for ESEARCH server implementations described in this section.

Note regarding Table 1: "[m]" means optional "MIN" and/or "MAX".

Combination of Result Options	"\$" Marker Value
SAVE PARTIAL	PARTIAL
SAVE PARTIAL MIN	PARTIAL & MIN
SAVE PARTIAL MAX	PARTIAL & MAX
SAVE PARTIAL MIN MAX	PARTIAL & MIN & MAX
SAVE PARTIAL COUNT [m]	all found messages

Table 1

### 3.3. Extension to UID FETCH

The PARTIAL extension also extends the UID FETCH command with a PARTIAL FETCH modifier. The PARTIAL FETCH modifier has the same syntax as the PARTIAL SEARCH result option. The presence of the PARTIAL FETCH modifier instructs the server to only return FETCH results for messages in the specified range. It is useful when the sequence-set (first) parameter in the UID FETCH command includes an unknown number of messages.

```
// Returning information for the last 3 messages in the UID range
C: 10 UID FETCH 25900:26600 (UID FLAGS) (PARTIAL -1:-3)
S: * 12888 FETCH (FLAGS (\Seen) UID 25996)
S: * 12889 FETCH (FLAGS (\Flagged \Answered) UID 25997)
S: * 12890 FETCH (FLAGS () UID 26600)
S: 10 OK FETCH completed

// Returning information for the first 5 messages in the UID range
C: 11 UID FETCH 25900:26600 (UID FLAGS) (PARTIAL 1:5)
S: * 12591 FETCH (FLAGS (\Seen) UID 25900)
S: * 12592 FETCH (FLAGS (\Flagged) UID 25902)
S: * 12593 FETCH (FLAGS (\Answered) UID 26310)
S: * 12594 FETCH (FLAGS () UID 26311)
S: * 12595 FETCH (FLAGS (\Answered) UID 26498)
S: 11 OK FETCH completed
```

### 3.4. Use of "PARTIAL" and "CONDSTORE" IMAP Extensions Together

This section is informative.

The PARTIAL FETCH modifier can be combined with the CHANGEDSINCE FETCH modifier [\[RFC7162\]](#).

```
// Returning information for the last 30 messages in the UID range
// that have any flags/keywords modified since MODSEQ 98305
C: 101 UID FETCH 25900:26600 (UID FLAGS
   ) (PARTIAL -1:-30 CHANGEDSINCE 98305)
S: * 12888 FETCH (FLAGS (\Flagged \Answered
   ) MODSEQ (98306) UID 25997)
S: * 12890 FETCH (FLAGS ( ) MODSEQ (98312) UID 26600)
S: 101 OK FETCH completed
```

The above example causes the server to first select the last 30 messages and then only return flag changes for a subset of those messages that have MODSEQ higher than 98305.

Note that the order of PARTIAL and CHANGEDSINCE FETCH modifiers in the UID FETCH command is not important, i.e., the above example can also use the "UID FETCH 25900:26600 (UID FLAGS) (CHANGEDSINCE 98305 PARTIAL -1:-30)" command and it would result in the same responses.

## 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 [IMAP4rev1 \[RFC3501\]](#) or [IMAP4rev2 \[RFC9051\]](#).

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

```
SP                = <Defined in RFC 5234>
MINUS             = "-"

capability        =/ "PARTIAL"
                  ;; <capability> from [RFC3501].

modifier-partial  = "PARTIAL" SP partial-range

partial-range-first = nz-number ":" nz-number
                  ;; Request to search from oldest (lowest UIDs) to
                  ;; more recent messages.
                  ;; A range 500:400 is the same as 400:500.
                  ;; This is similar to <seq-range> from [RFC3501]
                  ;; but cannot contain "*".

partial-range-last  = MINUS nz-number ":" MINUS nz-number
                  ;; Request to search from newest (highest UIDs) to
                  ;; oldest messages.
                  ;; A range -500:-400 is the same as -400:-500.

partial-range      = partial-range-first / partial-range-last

search-return-opt  =/ modifier-partial
                  ;; All conform to <search-return-opt> from
                  ;; [RFC4466] and [RFC9051].

search-return-data =/ ret-data-partial

ret-data-partial   = "PARTIAL"
                  SP "(" partial-range SP partial-results ")"
                  ;; <partial-range> is the requested range.

partial-results    = sequence-set / "NIL"
                  ;; <sequence-set> from [RFC3501].
                  ;; NIL indicates that no results correspond to
                  ;; the requested range.

tagged-ext-simple  =/ partial-range-last

fetch-modifier     =/ modifier-partial
                  ;; <fetch-modifier> from [RFC4466].
```

## 5. Security Considerations

This document defines an additional IMAP4 capability. As such, it does not change the underlying security considerations of IMAP4rev1 [RFC3501] and IMAP4rev2 [RFC9051]. The authors and reviewers believe that no new security issues are introduced with these additional IMAP4 capabilities.



This document defines an optimization that can reduce both the amount of work performed by the server and the amount of data returned to the client. Use of this extension is likely to cause the server and the client to use less memory than when the extension is not used. However, as this is going to be new code in both the client and the server, rigorous testing of such code is required in order to avoid introducing new implementation bugs.

## 6. IANA Considerations

### 6.1. Changes/Additions to the IMAP Capabilities Registry

IMAP4 capabilities are registered by publishing a Standards Track or IESG-approved Informational or Experimental RFC. The registry is currently located at <<https://www.iana.org/assignments/imap-capabilities>>.

IANA has added the PARTIAL extension to the "IMAP Capabilities" registry with RFC 9394 as the reference.

## 7. References

### 7.1. Normative References

- [ABNF] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/info/rfc5234>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, DOI 10.17487/RFC3501, March 2003, <<https://www.rfc-editor.org/info/rfc3501>>.
- [RFC4466] Melnikov, A. and C. Daboo, "Collected Extensions to IMAP4 ABNF", RFC 4466, DOI 10.17487/RFC4466, April 2006, <<https://www.rfc-editor.org/info/rfc4466>>.
- [RFC4731] Melnikov, A. and D. Cridland, "IMAP4 Extension to SEARCH Command for Controlling What Kind of Information Is Returned", RFC 4731, DOI 10.17487/RFC4731, November 2006, <<https://www.rfc-editor.org/info/rfc4731>>.
- [RFC5267] Cridland, D. and C. King, "Contexts for IMAP4", RFC 5267, DOI 10.17487/RFC5267, July 2008, <<https://www.rfc-editor.org/info/rfc5267>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [RFC9051]** Melnikov, A., Ed. and B. Leiba, Ed., "Internet Message Access Protocol (IMAP) - Version 4rev2", RFC 9051, DOI 10.17487/RFC9051, August 2021, <<https://www.rfc-editor.org/info/rfc9051>>.

## 7.2. Informative References

- [RFC7162]** Melnikov, A. and D. Cridland, "IMAP Extensions: Quick Flag Changes Resynchronization (CONDSTORE) and Quick Mailbox Resynchronization (QRESYNC)", RFC 7162, DOI 10.17487/RFC7162, May 2014, <<https://www.rfc-editor.org/info/rfc7162>>.

## Acknowledgments

This document was motivated by the Yahoo! team and their questions about best client practices for dealing with large mailboxes.

The authors of this document would like to thank the following people, who provided useful comments or participated in discussions of this document: Timo Sirainen and Barry Leiba.

This document uses a lot of text from RFC 5267. Thus, the work of the RFC 5267 authors -- Dave Cridland and Curtis King -- is appreciated.

## Authors' Addresses

### Alexey Melnikov

Isode Limited

Email: [alexey.melnikov@isode.com](mailto:alexey.melnikov@isode.com)

URI: <https://www.isode.com>

### Arun Prakash Achuthan

Yahoo!

Email: [arunprakash@myyahoo.com](mailto:arunprakash@myyahoo.com)

### Vikram Nagulakonda

Yahoo!

Email: [nvikram\\_imap@yahoo.com](mailto:nvikram_imap@yahoo.com)

### Luis Alves

Email: [luis.alves@lafaspot.com](mailto:luis.alves@lafaspot.com)