

IMAP REPLACE Extension
draft-ietf-extra-imap-replace-00

Abstract

This document defines an IMAP extension which can be used to replace an existing message in a message store with a new message. Message replacement is a common operation for clients that automatically save drafts or notes as a user composes them.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on February 9, 2019.

Copyright Notice

Copyright (c) 2018 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 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.

Table of Contents

1.	Conventions Used in This Document	2
2.	Overview	2
3.	REPLACE and UID REPLACE	3
3.1.	Advertising Support for REPLACE	3
3.2.	REPLACE Command	3
3.3.	UID REPLACE Command	4
3.4.	Semantics of REPLACE and UID REPLACE	5
3.5.	IMAP State Diagram Impacts	6
4.	Interaction with other extensions	6
4.1.	RFC 4314 , ACL	6
4.2.	RFC 4469 , CATENATE	6
4.3.	RFC 4315 , UIDPLUS	8
4.4.	RFC 6785 , IMAP Events in Sieve	8
4.5.	RFC 7162 , CONDSTORE/QRESYNC	8
4.6.	draft-ietf-extra-imap-objectid , OBJECTID	8
5.	Formal Syntax	8
6.	Security Considerations	9
7.	IANA Considerations	9
8.	Acknowledgements	9
9.	References	9
9.1.	Normative References	9
9.2.	Informative References	10
	Author's Address	10

[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.

[2.](#) Overview

This document defines an IMAP [[RFC3501](#)] extension to facilitate replacing an existing message with a new one. This is accomplished by defining a new REPLACE command and extending the UID command to allow UID REPLACE.

Since there is no replace function in the base IMAP specification, clients have instead had to use a combination of three separate commands issued in serial fashion; APPEND, STORE, EXPUNGE. Pipelining of these three commands is not recommended since failure

of any individual command should prevent subsequent commands from being executed lest the original message version be lost.

Because of the non-atomic nature of the existing sequence, interruptions can leave messages in intermediate states which can be seen and acted upon by other clients. Such interruptions can also strand older revisions of messages, thereby forcing the user to manually clean up multiple revisions of the same message in order to avoid wasteful quota consumption. Additionally, the existing sequence can fail on APPEND due to an over-quota condition even though the subsequent STORE/EXPUNGE would free up enough space for the newly revised message. And finally, server efficiencies may be possible with a single logical message replacement operation as compared to the existing APPEND/STORE/EXPUNGE sequence.

In its simplest form, the REPLACE command is a single-command encapsulation of APPEND, STORE +flags \DELETED and UID EXPUNGE for a message, except that it avoids any of the quota implications or intermediate states associated with the 3 command sequence. In handling a REPLACE command, a server MUST NOT generate a response code for the STORE +flags \DELETED portion of the sequence. Additionally, servers supporting the REPLACE command MUST NOT infer any inheritance of content, flags, or annotations from the message being replaced. Finally, the replaced and replacing messages SHOULD NOT be present in the mailbox at the same time.

3. REPLACE and UID REPLACE

3.1. Advertising Support for REPLACE

Servers that implement the REPLACE extension will return "REPLACE" as one of the supported capabilities in the CAPABILITY command response.

3.2. REPLACE Command

Arguments: message sequence number
 mailbox name
 OPTIONAL flag parenthesized list
 OPTIONAL date/time string
 message literal

Responses: no specific responses for this command

Result: OK - replace completed
 NO - replace error; can't remove specified message
 or can't add new message content
 BAD - command unknown or arguments invalid

Example:

```
C: A003 REPLACE 4 Drafts (\Seen \Draft) {312}
S: + Ready for literal data
C: Date: Thu, 1 Jan 2015 00:05:00 -0500 (EST)
C: From: Fritz Schmidt <fritz.ze@example.org>
C: Subject: happy new year !!
C: To: miss.mitzzy@example.org
C: Message-Id: <B238822388-0100000@example.org>
C: MIME-Version: 1.0
C: Content-Type: TEXT/PLAIN; CHARSET=US-ASCII
C:
C: Just saw the best fireworks show. Wish you were here.
C:
S: * APPENDUID 1 2000
S: * 5 EXISTS
S: * 4 EXPUNGE
S: A003 OK Replace completed
```

3.3. UID REPLACE Command

This extends the first form of the UID command (see [\[RFC3501\]](#) [Section 6.4.8](#)) to add the REPLACE command defined above as a valid argument. This form of REPLACE uses a UID rather than sequence number as its first parameter.

Example:

```
C: A004 UID REPLACE 2000 Drafts (\Seen \Draft) {350}
S: + Ready for literal data
C: Date: Thu, 1 Jan 2015 00:06:00 -0500 (EST)
C: From: Fritz Schmidt <fritz.ze@example.org>
C: Subject: happy new year !!
C: To: miss.mitzzy@example.org
C: Message-Id: <B238822389-0100000@example.org>
C: MIME-Version: 1.0
C: Content-Type: TEXT/PLAIN; CHARSET=US-ASCII
C:
C: Just saw the best fireworks show. Wish you were here.
C: Hopefully next year you can join us.
C:
S: * APPENDUID 1 2001
S: * 5 EXISTS
S: * 4 EXPUNGE
S: A004 OK Replace completed
```


3.4. Semantics of REPLACE and UID REPLACE

The REPLACE and UID REPLACE commands take five arguments: a message identifier, a named mailbox, an optional parenthesized flag list, an optional message date/time string, and a message literal. The message literal will be appended to the named mailbox, and the message specified by the message identifier will be removed from the selected mailbox. These operations will appear to the client as a single action. This has the same effect as the following sequence:

1. APPEND
2. [UID] STORE +FLAGS.SILENT \DELETED
3. UID EXPUNGE

In the cited sequence, the quota implications of the APPEND are evaluated within the context of the pending EXPUNGE so that only the net quota consumption is considered. Additionally, the EXPUNGE portion of the sequence only applies to the specified message, not all messages flagged as \Deleted.

Although the effect of REPLACE is identical to the steps above, the semantics are not identical; similar to MOVE [[RFC6851](#)], the intermediate states produced do not occur, and the response codes are different. In particular, the response codes for APPEND and EXPUNGE will be returned while those for the STORE operation MUST NOT be generated.

When an error occurs while processing REPLACE or UID REPLACE, the server MUST NOT leave the selected mailbox in an inconsistent state; any untagged EXPUNGE response MUST NOT be sent until all actions are successfully completed.

While it may be common for the named mailbox argument to match the selected mailbox for the common use case of replacing a draft, the REPLACE extension intentionally does not require the two to be the same. As an example, it's possible to use the REPLACE command to replace a message in the \Drafts special-use mailbox with a message in the \Sent special-use mailbox following message submission.

Because of the similarity of REPLACE to APPEND, extensions that affect APPEND affect REPLACE in the same way. Response codes such as TRYCREATE (see [[RFC3501](#)] [Section 6.3.11](#)), along with those defined by extensions, are sent as appropriate. See [Section 4](#) for more information about how REPLACE interacts with other IMAP extensions.

3.5. IMAP State Diagram Impacts

Unlike the APPEND command which is valid in the authenticated state, the REPLACE and UID REPLACE commands MUST only be valid in the selected state. This difference from APPEND is necessary since REPLACE operates on message sequence numbers.

4. Interaction with other extensions

This section describes how REPLACE interacts with some other IMAP extensions.

4.1. [RFC 4314](#), ACL

The ACL rights [[RFC4314](#)] required for UID REPLACE are the union of the ACL rights required for UID STORE and UID EXPUNGE in the current mailbox, and APPEND in the target mailbox.

4.2. [RFC 4469](#), CATENATE

Servers supporting both REPLACE and CATENATE [[RFC4469](#)] MUST support the additional append-data and resp-text-code elements defined the Formal Syntax section of [RFC4469](#) in conjunction with the REPLACE command. When combined with CATENATE, REPLACE can become a quite efficient way for message manipulation.

Example:

User composes message and attaches photo

```
-----
C: A010 APPEND Drafts (\Seen \Draft) {1201534}
S: + Ready for literal data
C: Date: Thu, 1 Jan 2015 00:10:00 -0500 (EST)
C: From: Fritz Schmidt <fritz.ze@example.org>
C: Message-ID: <B238822388-0100003@example.org>
C: MIME-Version: 1.0
C: Content-Type: multipart/mixed;
C:         boundary="-----030305060306060609050804"
C:
C: -----030305060306060609050804
C: Content-Type: text/plain; charset=utf-8; format=flowed
C: Content-Transfer-Encoding: 7bit
C:
C: Here is picture from the fireworks
C:
C: Yours...
C: Fritz
C:
C: -----030305060306060609050804
C: Content-Type: image/jpeg;
C:         name="Fireworks.jpg"
C: Content-Transfer-Encoding: base64
C: Content-Disposition: attachment;
C:         filename="Fireworks.jpg"
C:
C: <large base64 encoded part goes here>
C:
C: -----030305060306060609050804--
S: A010 OK [APPENDUID 1 3002] APPEND complete
```

User completes message with To: and Subject: fields

```
-----
C: A011 UID REPLACE 3002 Drafts CATENATE (TEXT {71}
S: + Ready for literal data
C: To: Mitzy <miss.mitzy@example.org>
C: Subject: My view of the fireworks
C: URL "/Drafts/;UID=3002")
S: * APPENDUID 1 3003
S: * 5 EXISTS
S: * 4 EXPUNGE
S: A011 OK REPLACE completed
```

Brandt

Expires February 9, 2019

[Page 7]

4.3. [RFC 4315](#), UIDPLUS

Servers supporting both REPLACE and UIDPLUS [[RFC4315](#)] SHOULD send APPENDUID in response to a UID REPLACE command. For additional information see [section 3 of RFC4315](#). Servers implementing REPLACE and UIDPLUS are also advised to send the APPENDUID response code in an untagged OK before sending the EXPUNGE or replaced responses. (Sending the APPENDUID in the tagged OK, as described in the UIDPLUS specification means that the client first receives an EXPUNGE for a message and afterwards APPENDUID for the new message. It can be unnecessarily difficult to process that sequence usefully.)

4.4. [RFC 6785](#), IMAP Events in Sieve

REPLACE applies to IMAP events in Sieve [[RFC6785](#)] in the same way that APPEND does. Therefore, REPLACE can cause a Sieve script to be invoked with the `imap.cause` set to "APPEND". Because the intermediate state of `STORE +FLAGS.SILENT \DELETED` is not exposed by REPLACE, no action will be taken that results in a `imap.cause` of FLAG.

4.5. [RFC 7162](#), CONDSTORE/QRESYNC

Servers implementing both REPLACE and CONDSTORE/QRESYNC [[RFC7162](#)] MUST treat the message being replaced as if it were being removed with a UID EXPUNGE command. Sections [3.2.9](#) and [3.2.10](#) of [RFC 7162](#) are particularly relevant for this condition.

4.6. [draft-ietf-extra-imap-objectid](#), OBJECTID

Servers implementing both REPLACE and OBJECTID MUST return different EMAILIDs for both the replaced and replacing messages. The only exception to this is the case outlined in OBJECTID [section 5.1](#) when the server detects that both messages' immutable content are identical.

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", "command-select", "mailbox", and "sequence-number". [[RFC4466](#)] defines the non-terminal "append-message".

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      =/ "REPLACE"

command-select =/ replace
replace        = "REPLACE" SP seq-number SP mailbox append-message
uid            = "UID" SP (copy / fetch/ search / store / move /
                        replace)
```

6. Security Considerations

This document is believed to add no security problems beyond those that may already exist with the base IMAP specification.

7. IANA Considerations

The IANA is requested to add REPLACE to the "IMAP 4 Capabilities" registry, <http://www.iana.org/assignments/imap4-capabilities>.

8. Acknowledgements

The author would like to thank the participants of IMAPEXT with particular thanks to Arnt Gulbrandsen, Alexey Melkinov, Chris Newman, and Bron Gondwana for their specific contributions.

9. References

9.1. Normative References

- [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>>.
- [RFC4314] Melnikov, A., "IMAP4 Access Control List (ACL) Extension", [RFC 4314](#), DOI 10.17487/RFC4314, December 2005, <<https://www.rfc-editor.org/info/rfc4314>>.
- [RFC4315] Crispin, M., "Internet Message Access Protocol (IMAP) - UIDPLUS extension", [RFC 4315](#), DOI 10.17487/RFC4315, December 2005, <<https://www.rfc-editor.org/info/rfc4315>>.
- [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>>.

- [RFC4469] Resnick, P., "Internet Message Access Protocol (IMAP) CATENATE Extension", [RFC 4469](#), DOI 10.17487/RFC4469, April 2006, <<https://www.rfc-editor.org/info/rfc4469>>.
- [RFC5234] 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>>.
- [RFC6785] Leiba, B., "Support for Internet Message Access Protocol (IMAP) Events in Sieve", [RFC 6785](#), DOI 10.17487/RFC6785, November 2012, <<https://www.rfc-editor.org/info/rfc6785>>.
- [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>>.

[9.2.](#) Informative References

- [RFC6851] Gulbrandsen, A. and N. Freed, Ed., "Internet Message Access Protocol (IMAP) - MOVE Extension", [RFC 6851](#), DOI 10.17487/RFC6851, January 2013, <<https://www.rfc-editor.org/info/rfc6851>>.

Author's Address

Stuart Brandt
Verizon
22001 Loudoun County Parkway
Ashburn, VA 20147
USA

Email: stujenerin@aol.com

