Internet Message Access Protocol (IMAP) - MOVE Extension

Abstract

This document defines an IMAP extension consisting of two new commands, MOVE and UID MOVE, that are used to move messages from one mailbox to another.

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 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6851.

Copyright Notice

Copyright (c) 2013 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.
1. Introduction

This document defines an IMAP [RFC3501] extension to facilitate moving messages from one mailbox to another. This is accomplished by defining a new MOVE command and extending the UID command to allow UID MOVE.

A move function is not provided in the base IMAP specification, so clients have instead had to use a combination of the COPY, STORE, and EXPUNGE commands to perform this very common operation.

Implementors have long pointed out some shortcomings with this approach. Because the moving of a message is not an atomic process, interruptions can leave messages in intermediate states. Because multiple clients can be accessing the mailboxes at the same time, clients can see messages in intermediate states even without interruptions. If the source mailbox contains other messages that are flagged for deletion, the third step can have the side effect of expunging more than just the set of moved messages. Additionally, servers with certain types of back-end message stores might have efficient ways of moving messages, which don’t involve the actual copying of data. Such efficiencies are often not available to the COPY/STORE/EXPUNGE process.

The MOVE extension is present in any IMAP implementation that returns "MOVE" as one of the supported capabilities to the CAPABILITY command.

2. 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 specified using ABNF [RFC5234].

Example lines prefaced by "C:" are sent by the client and ones prefaced by "S:" by the server.
3. MOVE and UID MOVE

3.1. MOVE Command

Arguments: sequence set
mailbox name

Responses: no specific responses for this command

Result: OK - move completed
NO - move error: can’t move those messages or to that name
BAD - command unknown or arguments invalid

3.2. UID MOVE Command

This extends the first form of the UID command (see [RFC3501], Section 6.4.8) to add the MOVE command defined above as a valid argument.

3.3. Semantics of MOVE and UID MOVE

The MOVE command takes two arguments: a message set (sequence numbers for MOVE, UIDs for UID MOVE) and a named mailbox. Each message included in the set is moved, rather than copied, from the selected (source) mailbox to the named (target) mailbox.

This means that a new message is created in the target mailbox with a new UID, the original message is removed from the source mailbox, and it appears to the client as a single action. This has the same effect for each message as this sequence:

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

Although the effect of the MOVE is the same as the preceding steps, the semantics are not identical: The intermediate states produced by those steps do not occur, and the response codes are different. In particular, though the COPY and EXPUNGE response codes will be returned, response codes for a STORE MUST NOT be generated and the \DELETED flag MUST NOT be set for any message.
Because a MOVE applies to a set of messages, it might fail partway through the set. Regardless of whether the command is successful in moving the entire set, each individual message SHOULD either be moved or unaffected. The server MUST leave each message in a state where it is in at least one of the source or target mailboxes (no message can be lost or orphaned). The server SHOULD NOT leave any message in both mailboxes (it would be bad for a partial failure to result in a bunch of duplicate messages). This is true even if the server returns a tagged NO response to the command.

Because of the similarity of MOVE to COPY, extensions that affect COPY affect MOVE in the same way. Response codes such as TRYCREATE (see [RFC3501], Section 6.4.7), as well as those defined by extensions, are sent as appropriate. See Section 4 for more information about how MOVE interacts with other IMAP extensions.

An example:

C: a UID MOVE 42:69 foo
S: * OK [COPYUID 432432 42:69 1202:1229]
S: * 22 EXPUNGE
S: (more expunges)
S: a OK Done

Note that the server may send unrelated EXPUNGE responses as well, if any happen to have been expunged at the same time; this is normal IMAP operation.

Implementers will need to read [RFC4315] to understand what UID EXPUNGE does, though full implementation of [RFC4315] is not necessary.

Note that moving a message to the currently selected mailbox (that is, where the source and target mailboxes are the same) is allowed when copying the message to the currently selected mailbox is allowed.

The server may send EXPUNGE (or VANISHED) responses before the tagged response, so the client cannot safely send more commands with message sequence number arguments while the server is processing MOVE or UID MOVE.

Both MOVE and UID MOVE can be pipelined with other commands, but care has to be taken. Both commands modify sequence numbers and also allow unrelated EXPUNGE responses. The renumbering of other messages in the source mailbox following any EXPUNGE response can be surprising and makes it unsafe to pipeline any command that relies on message sequence numbers after a MOVE or UID MOVE. Similarly, MOVE
cannot be pipelined with a command that might cause message
renumbering. See [RFC3501], Section 5.5, for more information about
ambiguities as well as handling requirements for both clients and
servers.

4. Interaction with Other Extensions

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

4.1. RFC 2087, QUOTA

The QUOTA extension (defined by [RFC2087]) may interact with MOVE on
some servers, in the sense that a MOVE command may succeed where COPY
would cause a quota overrun.

4.2. RFC 4314, Access Control List (ACL)

The ACL rights [RFC4314] required for MOVE and UID MOVE are the union
of the ACL rights required for UID STORE, UID COPY, and UID EXPUNGE.

4.3. RFC 4315, UIDPLUS

Servers supporting UIDPLUS [RFC4315] SHOULD send COPYUID in response
to a UID MOVE command. For additional information see Section 3 of
[RFC4315].

Servers implementing UIDPLUS are also advised to send the COPYUID
response code in an untagged OK before sending EXPUNGE or moved
responses. (Sending COPYUID in the tagged OK, as described in the
UIDPLUS specification, means that clients first receive an EXPUNGE
for a message and afterwards COPYUID for the same message. It can be
unnecessarily difficult to process that sequence usefully.)

4.4. RFC 5162, QRESYNC

The QRESYNC extension [RFC5162] states that the server SHOULD send
VANISHED rather than EXPUNGE in response to the UID EXPUNGE command.
The same requirement applies to MOVE, and a QRESYNC-enabled client
needs to handle both VANISHED and EXPUNGE responses to a UID MOVE
command.

If the server is capable of storing modification sequences for the
selected mailbox, it MUST increment the per-mailbox mod-sequence if
at least one message was permanently moved due to the execution of
the MOVE/UID MOVE command. For each permanently removed message, the
server MUST remember the incremented mod-sequence and corresponding
UID. If at least one message was moved, the server MUST send the
updated per-mailbox modification sequence using the HIGHESTMODSEQ
response code (defined in [RFC4551]) in the tagged or untagged OK
response.

When one or more messages are moved to a target mailbox, if the
server is capable of storing modification sequences for the mailbox,
the server MUST generate and assign new modification sequence numbers
to the moved messages that are higher than the highest modification
sequence of the messages originally in the mailbox.

4.5. IMAP Events in Sieve

MOVE applies to IMAP events in Sieve [RFC6785] in the same way as
COPY does. Therefore, MOVE can cause a Sieve script to be invoked
with the imap.cause set to "COPY". Because MOVE does not cause flags
to be changed, a MOVE command will not result in a script invocation
with the imap.cause set to "FLAG".

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", "sequence-set", and
"mailbox".

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   =/ "MOVE"

command-select =/ move

move          = "MOVE" SP sequence-set SP mailbox
uid            = "UID" SP (copy / fetch / search / store / move)

6. Security Considerations

MOVE does not introduce any new capabilities to IMAP, and this limits
the security impact. However, the transactional semantics of MOVE
may interact with specific implementations in ways that could have
unexpected consequences. For example, moving messages between
mailboxes under the quota root may require temporary suspension of
quota checking.

An additional area of concern is interaction with antispam,
antivirus, and other security scanning and auditing mechanisms.
Different mailboxes may have different security policies that could
interact with MOVE in complex ways. Scanning with updated rules may also be required when messages are moved even when the underlying policy has not changed.

MOVE does relieve a problem with the base specification, since client authors currently have to devise and implement complicated algorithms to handle partial failures of the STORE/COPY/EXPUNGE trio. Incomplete or improper implementation of these algorithms can lead to mail loss.

7. IANA Considerations

The IANA has added MOVE to the "IMAP 4 Capabilities" registry, <http://www.iana.org/assignments/imap4-capabilities>.

8. Acknowledgments

This document is dedicated to the memory of Mark Crispin, the inventor of the IMAP protocol, author of the IMAP protocol specification [RFC3501], and contributor to many other email specifications in the IETF.

An extension like this has been proposed many times, by many people. This document is based on several of those proposals, most recently that by Witold Krecicki. Witold, Benoit Claise, Adrien W. de Croy, Stephen Farrell, Bron Gondwana, Dan Karp, Christian Ketterer, Murray Kucherawy, Jan Kundrat, Barry Leiba, Alexey Melnikov, Kathleen Moriarty, Zoltan Ordogh, Pete Resnick, Timo Sirainen, Michael Slusarz, and others provided valuable comments.

9. References

9.1. Normative References


9.2. Informative References


Authors’ Addresses

Arnt Gulbrandsen
Schweppermannstr. 8
D-81671 Muenchen
Germany

Fax: +49 89 4502 9758
EMail: arnt@gulbrandsen.priv.no

Ned Freed (editor)
Oracle
800 Royal Oaks
Monrovia, CA 91016-6347
USA

EMail: ned+ietf@mrochek.com