

EXTRA
Internet-Draft
Intended status: Standards Track
Expires: October 1, 2018

B. Gondwana, Ed.
FastMail
March 30, 2018

**IMAP Extension for unique identifiers
draft-ietf-extra-imap-uniqueid-00**

Abstract

This document adds new properties to IMAP mailboxes and messages to allow clients to more efficiently re-use cached data for resources which have changed location on the server.

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 October 1, 2018.

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.	Introduction	2
2.	Conventions Used In This Document	3
3.	CAPABILITY Identification	3
4.	STATUS Command and Response Extensions	3
5.	FETCH Command and Response Extensions	4
6.	SEARCH Command Extension	7
7.	Response Codes	7
8.	Formal syntax	7
9.	Implementation considerations	7
10.	Future considerations	8
11.	IANA Considerations	8
12.	Security Considerations	9
13.	TODO	9
14.	Changes	9
14.1.	01	9
15.	Acknowledgments	9
16.	Normative References	9
	Author's Address	10

[1.](#) Introduction

IMAP stores are often used by many clients, which each cache information locally about the server state so that they don't need to download anything again. [[RFC3501](#)] defines that a mailbox can be uniquely referenced by its name and UIDVALIDITY, and a message within that mailbox can be uniquely referenced by its mailbox (name + UIDVALIDITY) and UID.

Further, [[RFC4315](#)] defines a COPYUID response which allows a client which copies messages between folders to know the mapping between the UIDs in the source and destination mailboxes, and hence update its local cache.

So a client which copies (or [[RFC6851](#)] moves) messages or renames folders can update its local cache, but any other client connected to the same store can not know with certainty that the messages are identical, and so will re-download everything.

This extension adds new properties to a message (EMAILID) and mailbox (MAILBOXID) which allow a client to quickly identify messages or mailboxes which have been renamed by another client.

This extension also adds an optional thread identifier (THREADID) to messages, which can be used by the server to indicate messages which it has identified to be related.

2. Conventions Used In This Document

In examples, "C:" indicates lines sent by a client that is connected to a server. "S:" indicates lines sent by the server to the client.

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 [\[RFC2119\]](#) when they appear in ALL CAPS. These words may also appear in this document in lower case as plain English words, absent their normative meanings.

3. CAPABILITY Identification

IMAP servers that support this extension MUST include "UNIQUEID" in the response list to the CAPABILITY command.

4. STATUS Command and Response Extensions

This extension defines one new status data item for the STATUS command and response:

MAILBOXID A unique identifier for the mailbox. This identifier SHOULD be retained when the mailbox is renamed. This identifier MUST NOT be identical if the mailbox does not meet the invariants for a mailbox with the same name and uidvalidity as a mailbox previously reported to have this UIDVALIDITY. A server MUST NOT return two mailboxes with the same MAILBOXID.

The value of the MAILBOXID is an opaque string of 1..255 bytes in length. The MAILBOXID is server assigned and read-only.

The server MAY choose to create a MAILBOXID value in a way that does not survive RENAME, (e.g. a digest of mailboxname + uidvalidity could be used as a "MAILBOXID" and it would be legal, though of course clients would not get the full benefits of this extension from such a server).

Example:


```
C: 3 create foo
S: 3 OK Completed
C: 4 create bar
S: 4 OK Completed
C: 5 status foo (mailboxid uidvalidity)
S: * STATUS foo (UIDVALIDITY 1521472287 MAILBOXID 7uijf0bg4yeo51a7)
S: 5 OK Completed
C: 6 status bar (mailboxid uidvalidity)
S: * STATUS bar (UIDVALIDITY 1521472288 MAILBOXID u8vhi0uy16v5k99p)
S: 6 OK Completed
C: 7 rename foo renamed
S: * OK rename foo renamed
S: 7 OK Completed
C: 8 status renamed (mailboxid uidvalidity)
S: * STATUS renamed (UIDVALIDITY 1521472287 MAILBOXID 7uijf0bg4yeo51a7)
S: 8 OK Completed
C: 9 status bar (mailboxid uidvalidity)
S: * STATUS bar (UIDVALIDITY 1521472288 MAILBOXID u8vhi0uy16v5k99p)
S: 9 OK Completed
```

When the LIST-STATUS IMAP capability [[RFC5819](#)] is also available, the STATUS command can be combined with the LIST command to further improve efficiency. This way, the unique ids of many mailboxes can be queried with just one LIST command.

Example:

```
C: 10 list "" "*" return (status (mailboxid))
S: * LIST (\HasNoChildren) "." INBOX
S: * STATUS INBOX (MAILBOXID 2l3g6iiw520ruqo1mcspbfv2)
S: * LIST (\HasNoChildren) "." bar
S: * STATUS bar (MAILBOXID 0vfkeq1wzgse343kegel6glk)
S: * LIST (\HasNoChildren) "." renamed
S: * STATUS renamed (MAILBOXID 4jbgi9zgfnc9gtwu7qmkye45)
S: 10 OK Completed (0.001 secs 3 calls)
```

5. FETCH Command and Response Extensions

This extension defines two additional FETCH items on messages:

EMAILID A server allocated opaque string value (1..255 bytes) which uniquely identifies the content of a single message. That is the exact bytes of the [RFC822](#) FETCH item. The server **MUST NOT** return the same EMAILID for two different sets of bytes. The server **SHOULD** return the same EMAILID for the same set of bytes.

The server **SHOULD** retain the same INTERNALDATE for messages with the same EMAILID.

THREADID A server allocated opaque string value (1..255 bytes) which is the same for messages which the server has, with its own algorithm, decided are "related" in some way. This is generally based on some combination of References, In-Reply-To and Subject but the exact logic is left up to the server implementation. If the mailbox does not support THREADID, it will return NIL for fetch.

THREADID MUST NOT change if EMAILID is the same.

Example:

C: 5 append inbox "20-Mar-2018 03:07:37 +1100" {733}

[...]

Subject: Message A

Message-ID: <fake.1521475657.54797@hotmail.com>

[...]

S: 5 OK [APPENDUID 1521475658 1] Completed

C: 11 append inbox "20-Mar-2018 03:07:37 +1100" {793}

[...]

Subject: Re: Message A

Message-ID: <fake.1521475657.21213@gmail.com>

References: <fake.1521475657.54797@hotmail.com>

[...]

S: 11 OK [APPENDUID 1521475658 2] Completed

C: 17 append inbox "20-Mar-2018 03:07:37 +1100" {736}

[...]

Subject: Message C

Message-ID: <fake.1521475657.60280@hotmail.com>

[...]

S: 17 OK [APPENDUID 1521475658 3] Completed

C: 22 fetch 1:* (emailid threadid)

S: * 1 FETCH (EMAILID Md8976d99ac3275bb4e THREADID T4964b478a75b7ea9)

S: * 2 FETCH (EMAILID Mdd3c288836c4c7a762 THREADID T4964b478a75b7ea9)

S: * 3 FETCH (EMAILID Mf2e25fdc09b49ea703 THREADID T6311863d02dd95b5)

S: 22 OK Completed (0.000 sec)

C: 23 move 2 foo

S: * OK [COPYUID 1521475659 2 1] Completed

S: * 2 EXPUNGE

S: 23 OK Completed

C: 24 fetch 1:* (emailid threadid)

S: * 1 FETCH (EMAILID Md8976d99ac3275bb4e THREADID T4964b478a75b7ea9)

S: * 2 FETCH (EMAILID Mf2e25fdc09b49ea703 THREADID T6311863d02dd95b5)

S: 24 OK Completed (0.000 sec)

C: 25 select "foo"

C: 25 select "foo"

[...]

S: 25 OK [READ-WRITE] Completed

C: 26 fetch 1:* (emailid threadid)

S: * 1 FETCH (EMAILID Mdd3c288836c4c7a762 THREADID T4964b478a75b7ea9)

S: 26 OK Completed (0.000 sec)

6. SEARCH Command Extension

This extension defines two new search keys for the SEARCH command:

EMAILID blob Messages with the exactly matching EMAILID (bytes, does not depend on charset, case IS significant)

THREADID blob Messages with the exactly matching THREADID (bytes, does not depend on charset, case IS significant)

Example: (as if run before the MOVE above when the mailbox had 3 messages)

```
C: 27 search emailid Md8976d99ac3275bb4e918af4
S: * SEARCH 1
S: 27 OK Completed (1 msgs in 0.000 secs)
C: 28 search threadid T4964b478a75b7ea9
S: * SEARCH 1 2
S: 28 OK Completed (2 msgs in 0.000 secs)
```

7. Response Codes

This specification defines a new response code "MAILBOXID" to the "CREATE" command.

Example:

```
C: 3 create foo
S: 3 OK [MAILBOXID 87nn6m5t8azrrrywp4qfxkez] Completed
```

8. Formal syntax

resp-text-code =/ "MAILBOXID"

uniqueid = 1*255(ALPHA / DIGIT / "_" / "-")

9. Implementation considerations

The case of RENAME INBOX may need special handling for unique ids.

It is OK to change the mailboxid on a folder RENAME, but you MUST NOT ever re-use a MAILBOXID which has been shown to a client.

It is advisable (though not required) to have MAILBOXID be globally unique, but they it is only required to be unique within a single server.

If you have unique IDs larger than 255 bytes in a data store, it is safe to use a cryptographically strong hash to convert your IDs into a MAILBOXID value to display for this extension. It may be worth caching that value, as STATUS MAILBOXID is expected to be cheap for the server to calculate.

Ideas for implementing EMAILID:

- o Digest of (MailboxName/UIDVALIDITY/UID) - is not kept when moving messages, but is guarantee unique.
- o Digest of message content ([RFC822](#) bytes) - expensive unless cached
- o ID allocated at creation time - very efficient but requires storage of an additional value.

Ideas for implementing THREADID:

- o Derive from EMAILID of first seen message in the thread.
- o ID allocated at creation time.

There is a need to index and look up reference/in-reply-to data efficiently at message creation to efficiently find matching messages for threading. Threading may be either across folders, or within each folder only. The server has significant leeway here.

10. Future considerations

This extension is intentionally defined to be compatible with the data model in [[I-D.ietf-jmap-mail](#)].

A future extension could be proposed to give a way to SELECT a mailbox by MAILBOXID rather than name.

An extension to allow fetching message content directly via EMAILID and message listings by THREADID could be proposed.

11. IANA Considerations

The IANA is requested to add "UNIQUEID" to the "IMAP Capabilities" registry located at <<http://www.iana.org/assignments/imap-capabilities>>.

12. Security Considerations

If globally unique identifiers are used as MAILBOXIDs on IMAP folders, then it may be possible to tell when an account or folder has been renamed, even if all the mail has been deleted, if the folders themselves are retained.

13. TODO

- o add ABNF
- o consider whether IDs should be constrained to a smaller set of allowed characters (in conjunction with JMAP group)
- o consider whether STATUS response should be MAILBOXID ("value") to allow non-integer responses.

14. Changes

14.1. 01

- o renamed UNIQUEID (status item) to MAILBOXID
- o renamed MSGID to EMAILID
- o renamed THRID to THREADID
- o added TODO section

15. Acknowledgments

The EXTRA working group at IETF.

The gmail team's X-GM-THRID and X-GM-MSGID implementation.

16. Normative References

- [I-D.ietf-jmap-mail]
Jenkins, N., "JMAP for Mail", [draft-ietf-jmap-mail-04](#)
(work in progress), March 2018.
- [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>>.
- [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>>.
- [RFC5819] Melnikov, A. and T. Sirainen, "IMAP4 Extension for Returning STATUS Information in Extended LIST", [RFC 5819](#), DOI 10.17487/RFC5819, March 2010, <<https://www.rfc-editor.org/info/rfc5819>>.
- [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

Bron Gondwana (editor)
FastMail
Level 2, 114 William St
Melbourne VIC 3000
Australia

Email: brong@fastmailteam.com
URI: <https://www.fastmail.com>

