Network Working Group
Request for Comments: 2359
Category: Standards Track

J. Myers
Netscape Communications

June 1998

IMAP4 UIDPLUS extension

Status of this Memo

Copyright Notice

IESG NOTE

The IMAP extension described here assumes a particular means of using IMAP to support disconnected operation. However, this means of supporting disconnected operation is not yet documented. Also, there are multiple theories about how best to do disconnected operation in IMAP, and as yet, there is no consensus on which one should be adopted as a standard.

This document is being approved as a Proposed Standard because it does not appear to have technical flaws in itelf. However, approval of this document as a Proposed Standard should not be considered an IETF endorsement of any particular means of doing disconnected operation in IMAP.

Table of Contents

1. Abstract

The UIDPLUS extension of the Internet Message Access Protocol [IMAP4] provides a set of features intended to reduce the amount of time and resources used by some client operations. The features in UIDPLUS are primarily intended for disconnected-use clients.

2. Conventions Used in this Document

In examples, "C:" and "S:" indicate lines sent by the client and server respectively.

The key words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", and "MAY" in this document are to be interpreted as defined in "Key words for use in RFCs to Indicate Requirement Levels" [KEYWORDS].

3. Introduction and Overview

The UIDPLUS extension is present in any IMAP4 server implementation which returns "UIDPLUS" as one of the supported capabilities to the CAPABILITY command. The UIDPLUS extension contains one additional command and additional data returned with successful APPEND and COPY commands.

Clients that wish to use the new command in UIDPLUS must of course first test for the presence of the extension by issuing a CAPABILITY command. Each of the features in UIDPLUS are optimizations; clients can provide the same functionality, albeit more slowly, by using commands in the base protocol. With each feature, this document recommends a fallback approach to take when the UIDPLUS extension is not supported by the server.

4. Features

4.1. UID EXPUNGE Command

Arguments: message set

Data: untagged responses: EXPUNGE

Result: OK - expunge completed
NO - expunge failure (e.g. permission denied)
BAD - command unknown or arguments invalid

The UID EXPUNGE command permanently removes from the currently selected mailbox all messages that both have the \Deleted flag set and have a UID that is included in the specified message set. If a message either does not have the \Deleted flag set or is has a UID that is not included in the specified message set, it is not affected.

This command may be used to ensure that a replayed EXPUNGE command does not remove any messages that have been marked as \Deleted between the time that the user requested the expunge operation and the time the server processes the command.

If the server does not support the UIDPLUS capability, the client should fall back to using the STORE command to temporarily remove the \Deleted flag from messages it does not want to remove. The client could alternatively fall back to using the EXPUNGE command, risking the unintended removal of some messages.

```
Example:    C: A003 UID EXPUNGE 3000:3002
            S: * 3 EXPUNGE
            S: * 3 EXPUNGE
            S: * 3 EXPUNGE
            S: A003 OK UID EXPUNGE completed
```

4.2. APPENDUID response code

Successful APPEND commands return an APPENDUID response code in the tagged OK response. The APPENDUID response code contains as arguments the UIDVALIDITY of the destination mailbox and the UID assigned to the appended message.

If the server does not support the UIDPLUS capability, the client can only discover this information by selecting the destination mailbox and issuing FETCH commands.

```
Example:    C: A003 APPEND saved-messages (\Seen) {310}
            C: Date: Mon, 7 Feb 1994 21:52:25 -0800 (PST)
            C: From: Fred Foobar <foobar@Blurdybloop.COM>
            C: Subject: afternoon meeting
            C: To: mooch@owatagu.siam.edu
            C: Message-Id: <B27397-0100000@Blurdybloop.COM>
            C: MIME-Version: 1.0
            C: Content-Type: TEXT/PLAIN; CHARSET=US-ASCII
            C:
            C: Hello Joe, do you think we can meet at 3:30 tomorrow?
            C:
            S: A003 OK [APPENDUID 38505 3955] APPEND completed
```

4.3. COPYUID response code

Successful COPY and UID COPY commands return a COPYUID response code in the tagged OK response whenever at least one message was copied. The COPYUID response code contains as an argument the UIDVALIDITY of the appended-to mailbox, a message set containing the UIDs of the messages copied to the destination mailbox, in the order they were copied, and a message containing the UIDs assigned to the copied messages, in the order they were assigned. Neither of the message sets may contain extraneous UIDs or the symbol '*'.

If the server does not support the UIDPLUS capability, the client can only discover this information by selecting the destination mailbox and issuing FETCH commands.

```
Example:    C: A003 COPY 2:4 MEETING
            S: A003 OK [COPYUID 38505 304,319:320 3956:3958] Done
            C: A003 UID COPY 305:310 MEETING
            S: A003 OK Done
```

5. Formal Syntax

The following syntax specification uses the augmented Backus-Naur Form (BNF) notation as specified in [RFC-822] as modified by [IMAP4]. Non-terminals referenced but not defined below are as defined by [IMAP4].

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.

```
resp_code_apnd  ::= "APPENDUID" SPACE nz_number SPACE uniqueid

resp_code_copy  ::= "COPYUID" SPACE nz_number SPACE set SPACE set

uid_expunge     ::= "UID" SPACE "EXPUNGE" SPACE set
```

6. References

[IMAP4] Crispin, M., "Internet Message Access Protocol - Version 4rev1", RFC 2060, December 1996.

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC-822] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, RFC 822, August 1982.

7. Security Considerations

There are no known security issues with this extension.

8. Author's Address

John Gardiner Myers
Netscape Communications
501 East Middlefield Road
Mail Stop MV-029
Mountain View, CA 94043

EMail: jgmyers@netscape.com

9. Full Copyright Statement