

INTERNET-DRAFT
Intended status: Standards Track
Expires: December 2015

Jayantheesh S B
Samsung
Narendra Singh Bisht
Samsung
July 20, 2015

The IMAP APPENDLIMIT Extension
draft-ietf-imapapnd-appendlimit-extension-00

Abstract

This memo defines an extension to the IMAP service whereby a server can advertise its capability, to support maximum mail upload size using CAPABILITY, SELECT/EXAMINE and LIST command.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#). This document may not be modified, and derivative works of it may not be created, except to publish it as an RFC and to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on December, 2015.

Copyright Notice

Copyright (c) 2015 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.

Table of Contents

- [1.](#) Introduction [2](#)
- [1.1.](#) Conventions and Terminology [2](#)
- [2.](#) APPENDLIMIT Extension [3](#)
- [3.](#) Mailbox specific APPENDLIMIT [3](#)
- [3.1.](#) SELECT response [4](#)
- [3.2.](#) LIST response [4](#)
- [4.](#) APPEND response [5](#)
- [5.](#) Formal syntax [5](#)
- [6.](#) Security Considerations [6](#)
- [7.](#) IANA Considerations [6](#)
- [8.](#) References [6](#)
- [8.1](#) Normative References [6](#)
- [8.2](#) Informative References [7](#)
- [9.](#) Acknowledgement [7](#)
- [10.](#) Author's Address [7](#)

[1. Introduction](#)

Several IMAP server have limitation for mail upload size which is not published to the email client. When email client APPEND a mail with huge attachments, it fails due to size restriction set by the IMAP server. This results in unnecessary resource usage. Especially in the mobile device environment, appending mail with huge attachment consumes device resources like device battery power and mobile data.

The IMAP APPENDLIMIT extension provides an ability to advertise maximum upload size allowed by the IMAP server, so that email client knows the size limitation beforehand.

[1.1. Conventions and Terminology](#)

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 [RFC 2119](#).

Example lines prefaced by "C:" are sent by the client and ones prefaced by "S:" by the server. The five characters [...] means that something has been elided.

2. APPENDLIMIT Extension

An IMAP server that supports APPENDLIMIT advertises this by including the word APPENDLIMIT in its capability list. IMAP server shall publish the supported mail upload size as part of CAPABILITY response. The advertised upload limit is common across the mailboxes, but client can still issue SELECT/EXAMINE or LIST command to get the mailbox specific upload limit set by the IMAP server. In this case, APPENDLIMIT value obtained as part of SELECT/EXAMINE or LIST command takes precedence over the value returned as part of CAPABILITY response.

The following example, demonstrates the APPENDLIMIT capability with mailbox limit.

```
C: t1 CAPABILITY
S: * CAPABILITY IMAP4rev1 ID APPENDLIMIT=257890
S: t1 OK foo
```

If APPENDLIMIT value is omitted in CAPABILITY response, then client SHOULD issue SELECT/EXAMINE or LIST command to get the mailbox specific limit set by the server. New response code APPENDLIMIT is added to get the mailbox specific limit. Refer [section 5](#) for response code syntax.

The following example demonstrates, its usage.

```
C: t1 CAPABILITY
S: * CAPABILITY IMAP4rev1 ID APPENDLIMIT
S: t1 OK foo

C: t2 SELECT INBOX
S: * 172 EXISTS
S: * OK [APPENDLIMIT 257890] Maximum upload limit
S: * OK [UIDVALIDITY 3857529045] UIDs valid
S: * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
S: * OK [PERMANENTFLAGS (\Deleted \Seen \*)] Limited
S: t2 OK [READ-WRITE] SELECT completed
```

By looking at the upload size restriction set by the IMAP server, client MUST not try to upload mail more than advertised limit in the APPEND command.

3. Mailbox specific APPENDLIMIT

IMAP server can still have mailbox specific APPENDLIMIT restriction, which may not be advertised as part of CAPABILITY response. In this case, client can issue SELECT/EXAMINE command to get the per mailbox

specific limit set by the server. Similarly, if client wish to know the mailbox specific limit in authenticated state, can be done by issuing the LIST command in combination with STATUS command.

[3.1](#) SELECT response

Client can get the per mailbox append limit by issuing the SELECT/EXAMINE command. APPENDLIMIT size to this mailbox is obtained as part of untagged OK response. In this case, this APPENDLIMIT value will supersede the value received as part of CAPABILITY response. If no per-mailbox APPENDLIMIT is specified for a folder, but the server did specify a common APPENDLIMIT in the CAPABILITY response, then the common APPENDLIMIT applies to that folder.

```
C: t2 SELECT INBOX
S: * 172 EXISTS
S: * OK [APPENDLIMIT 257890] Maximum upload limit
S: [...]
S: t2 OK [READ-WRITE] SELECT completed
```

In the above example, APPENDLIMIT represents the maximum upload size for this mailbox.

OK [APPENDLIMIT <n>] Maximum upload limit for this mailbox, in bytes. Refer to [section 5](#) for more information. If this is missing, the client can always honour the value received as part of CAPABILITY response.

[3.2](#) LIST response

IMAP client can get the mailbox specific APPENDLIMIT in authenticated state, where it do not need to issue SELECT/EXAMINE command. LIST command in combination with STATUS command can be issued to get the per mailbox specific APPENDLIMIT set by the server. Refer [RFC 5819](#) for the usage of LIST command in combination with STATUS command. Note that a server implementing this extension, is syntactically compatible with [RFC 5819](#), however support for [RFC 5258](#) or [RFC 5819](#) is not required, when implementing this extension.

The following example demonstrates, its usage.

```
C: t1 LIST "" % RETURN (STATUS (APPENDLIMIT))
S: * LIST () "." "INBOX"
S: * STATUS "INBOX" (APPENDLIMIT 257890)
S: t1 OK List completed.
```

New attribute APPENDLIMIT is added to get the limit set by the server for this mailbox as part of STATUS command. The STATUS response occurs as a result of an STATUS command. It returns the mailbox name that matches the STATUS specification and the requested mailbox status information. IMAP server should recognize an extra "RETURN (STATUS (APPENDLIMIT))" at the end of a LIST command and emit an extra STATUS response for each matching mailbox. Refer to [section 5](#) for the syntax.

Invoking STATUS command with APPENDLIMIT is also acceptable. Below example demonstrates, its usage.

```
C: t1 STATUS INBOX (APPENDLIMIT)
S: * STATUS INBOX (APPENDLIMIT 257890)
S: t1 OK STATUS completed
```

4. APPEND response

If client uploads a mail which exceeds the maximum upload size set to that mailbox, then server SHALL reject the APPEND command with a tagged TOOBIG response code. Refer [RFC 4469](#) Section (4) for various APPEND response codes and its handling.

Client can avoid use of LITERAL+, when maximum upload size supported by the IMAP server is unknown.

5. Formal syntax

The following syntax specification uses the Augmented Backus-Naur Form (ABNF) notation as specified in [[RFC5234](#)] including the core rules in [Appendix B.1](#). [[RFC3501](#)] defines the non-terminals "capability", "resp-text-code" and "status-att".

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.

```
appendlimit-cap = "APPENDLIMIT" ["=" append-limit-value]
capability /= appendlimit-cap
```

```
appendlimit-respcode = "APPENDLIMIT" SP append-limit-value
resp-text-code /= appendlimit-respcode
```

```
append-limit-value = 1*DIGIT
                    ; Unsigned 64-bit integer
                    ; (0 <= n < 18,446,744,073,709,551,615)
```

```
appendlimit-status-att = "APPENDLIMIT"
status-att /=appendlimit-status-att
```

A number indicating the fixed maximum message size in bytes that the server will accept. APPENDLIMIT=0 indicates the server SHALL not accept APPEND command due to size restriction. The syntax of the parameter follows the augmented BNF notation of [RFC5234]. If this capability is omitted, no information is conveyed about the server's fixed maximum mail upload size.

6. Security Consideration

It is believed that this extension doesn't add any new security considerations that are not already present in the base IMAP protocol [RFC3501].

7. IANA Considerations

The IANA is requested to add APPENDLIMIT to the IMAP4 Capabilities Registry. [[Note to RFC-editor: please remove the following before publication: This registration should take place at the following location: <http://www.iana.org/assignments/imap4-capabilities>]]

8. References

8.1 Normative References

The following documents contain definitions or specifications that are necessary to understand this document properly

- [RFC2119] Bradner, "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), Harvard University, March 1997.
- [RFC3501] Crispin, "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", University of Washington, March 2003
- [RFC5234] Crocker, Overell, "Augmented BNF for Syntax Specifications: ABNF", [RFC 5234](#), Brandenburg Internetworking, Demon Internet Ltd, January 2008
- [RFC5322] P. Resnick, Ed, "Internet Message Format", [RFC 5322](#), Qualcomm Incorporated, October 2008
- [RFC2088] J. Myers, Carnegie Mellon, "IMAP4 non-synchronizing literals", January 1997
- [RFC4469] P. Resnick, "Internet Message Access Protocol (IMAP) CATENATE Extension", April 2006

- [[RFC5819](#)] A. Melnikov, T. Sirainen, "IMAP4 Extension for Returning STATUS Information in Extended LIST", March 2010
- [RFC5258] A. Melnikov, B. Leiba, "Internet Message Access Protocol version 4 - LIST Command Extensions", March 2010

8. 2 Informative References

The following documents describe related protocols:

- [RFC2087] Myers, J., "IMAP4 QUOTA extension", [RFC 2087](#), January 1997
- [RFC7377] B. Leiba, A. Melnikov, "IMAP4 Multimapbox SEARCH Extension", [RFC 7377](#), October 2014

9. Acknowledgement

Thanks to Alexey Melnikov, Dave Cridland, Adrien de Croy, Michael M Slusarz, Timo Sirainen, Chris Newman, Pete Maclean, Jamie Nicolson, Stu Brandt, Bron Gondwana, Arnt Gulbrandsen, Cyrus Daboo, Jan Kundrat, Brandon Long and Barry Leiba for providing valuable comments.

10. Author's Address

Jayantheesh S B
Samsung Telecommunications America
[685 US Highway 202/206](#)
Bridgewater, NJ 08807
USA

Email: jayantheesh.sb@gmail.com

Narendra Singh Bisht
Samsung Telecommunications America
[685 US Highway 202/206](#)
Bridgewater, NJ 08807
USA

Email: narendrasingh.bisht@gmail.com