

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: June 20, 2016

Jayanthesh  
Narendra  
Samsung Electronics America  
December 18, 2015

**The IMAP APPENDLIMIT Extension**  
**draft-ietf-imapapnd-appendlimit-extension-07.txt**

Abstract

This document defines an extension to the IMAP service whereby a server can inform the client about a maximum mail upload size, allowing the client to avoid sending APPEND commands that will fail because of the messages are too large.

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 <http://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 June 20, 2016.

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. STATUS response to the STATUS command . . . . .](#) [4](#)
- [3.2. STATUS response to the LIST command . . . . .](#) [4](#)
- [3.3. APPENDLIMIT behaviour . . . . .](#) [4](#)
- [4. APPEND response . . . . .](#) [5](#)
- [5. Formal syntax . . . . .](#) [5](#)
- [6. Security Considerations . . . . .](#) [5](#)
- [7. IANA Considerations . . . . .](#) [6](#)
- [8. Acknowledgements . . . . .](#) [6](#)
- [9. References . . . . .](#) [6](#)
- [9.1. Normative References . . . . .](#) [6](#)
- [9.2. Informative References . . . . .](#) [7](#)
- Authors' Addresses . . . . . [7](#)

**1. Introduction**

Some IMAP servers have limits for mail upload size, and those limits are not published to the email client. When the email client APPENDS a message with huge attachments, using non-synchronizing literals, the APPEND fails because of the upload limit, but the client has already sent the message data anyway. 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. By implementing this extension, IMAP server side processing of huge attachments above the maximum upload size can be avoided.

**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 [[RFC2119](#)].

In examples, "C:" and "S:" indicate lines sent by the client and server respectively. If a single "C:" or "S:" label applies to multiple lines, then the line breaks between those lines are for editorial clarity only and are not part of the actual protocol exchange.



## 2. APPENDLIMIT Extension

An IMAP server that supports APPENDLIMIT extension advertises this by including the name APPENDLIMIT in its capability list in authenticated state. The server may also advertise this extension before the user has logged in. If this capability is omitted, no information is conveyed about the server's fixed maximum mail upload size. IMAP server can publish the APPENDLIMIT capability in two formats.

(a) APPENDLIMIT=<number>

This indicates that the IMAP server has the same upload limit for all the mailboxes. The following example, demonstrates the APPENDLIMIT capability with same upload limit for all mailboxes.

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

(b) APPENDLIMIT

The APPENDLIMIT capability without any value indicates that the IMAP server supports this extension, and that the client will need to discover upload limits for each mailbox, which might differ from mailbox to mailbox. The following example, demonstrates the APPENDLIMIT capability without any value.

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

In this case the client SHOULD get APPENDLIMIT value by issuing STATUS or LIST command. New response code APPENDLIMIT is added to get the mailbox specific limit. Refer [section 5](#) for response code syntax.

IMAP client SHOULD be able to parse both kind of formats. By looking at the upload size advertised by the IMAP server, client MUST NOT try to APPEND mail more than advertised limit.

## 3. Mailbox specific APPENDLIMIT

An IMAP server can have mailbox-specific APPENDLIMIT values, which will not be advertised as part of CAPABILITY response. The IMAP server can publish specific values for each mailbox, and can publish "NIL" for a mailbox to convey that there is no APPENDLIMIT for that



mailbox. The following subsections describe the changes to the STATUS and LIST commands in support of this situation.

### **[3.1.](#) STATUS response to the STATUS command**

A new attribute APPENDLIMIT is added to get the limit set by the server for a mailbox as part of STATUS command. IMAP client should issue STATUS command with APPENDLIMIT item to get the mailbox specific upload value. The following example, demonstrates its usage.

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

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

### **[3.2.](#) STATUS response to the LIST command**

If the server advertises the LIST-STATUS capability [[RFC5819](#)], the client can issue LIST in combination with the STATUS return option to get the mailbox-specific upload value. 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.
```

The IMAP server MUST recognize the APPENDLIMIT attribute and include an appropriate STATUS response for each matching mailbox. Refer to [Section 5](#) for the syntax.

If the server does not support the STATUS return option on the LIST command, then the client should use the STATUS command instead.

### **[3.3.](#) APPENDLIMIT behaviour**

Computing the APPENDLIMIT should be fast, and need not take ACLs, quotas, and other such information into account. The APPENDLIMIT specifies one part of the policy, but an APPEND command can still fail due to issues related to ACLs and quotas issues, even if the message being appended is smaller than the APPENDLIMIT.



#### 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 [[RFC4469](#)] Section (4) for various APPEND response codes and its handling.

Client SHOULD avoid use of non-synchronizing literals [[LITERAL-](#)], when maximum upload size supported by the IMAP server is unknown. Refer [section 4.2.2.3 of \[RFC4549\]](#) for usage of non-synchronizing literals and its risk for disconnected IMAP clients.

#### 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" and "status-att", and [[RFC4466](#)] defines "status-att-val".

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 /= "APPENDLIMIT" ["=" number]
           ;; capability is defined in RFC 3501
```

```
status-att /= "APPENDLIMIT"
           ;; status-att is defined in RFC 3501
```

```
status-att-val /= "APPENDLIMIT" SP (number / nil)
                ;; status-att-val is defined in RFC 4466
```

The number indicates the fixed maximum message size in octets that the server will accept. An APPENDLIMIT number of 0 indicates the server will not accept any APPEND commands at all for the affected mailboxes.

#### 6. Security Considerations

The IMAP APPENDLIMIT extension described in this document can conceivably be used to facilitate Denial-of-Service attacks by allowing an attacker to home in on a critical value right away. The attacker might want to send a large data block to the server repeatedly, forcing the server to process the block, but would not want to limit the scope of its attack by filling an actual mailbox





with successful appends. Without this extension, the attacker needs to guess: a too-small guess results in an appended message that takes up the user's quota, while a far-too-large guess might simply cause the server to terminate the connection because of suspected abuse. But with this extension, the attacker can immediately choose a value that's a little too large, but not so much as to trigger an "abuse" response, making it easier to mount such an attack. To mitigate this extension's input to such an attack, a server might take a harder line on message sizes that are above the APPENDLIMIT value -- because the client knows the limit and should not even be trying to send such commands, a server might consider even a single attempt to be abusive, and terminate the IMAP connection straight away.

## 7. IANA Considerations

IANA is asked to add "APPENDLIMIT" to the IMAP Capabilities registry, using this document as its reference.

## 8. Acknowledgements

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.

## 9. References

### 9.1. Normative References

[LITERAL-]

Melnikov, A., "IMAP4 non-synchronizing literals, <https://tools.ietf.org/html/draft-ietf-imapapnd-rfc2088bis-01> (work in progress)", October 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

[RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", [RFC 3501](#), DOI 10.17487/RFC3501, March 2003, <<http://www.rfc-editor.org/info/rfc3501>>.

[RFC4466] Melnikov, A. and C. Daboo, "Collected Extensions to IMAP4 ABNF", [RFC 4466](#), DOI 10.17487/RFC4466, April 2006, <<http://www.rfc-editor.org/info/rfc4466>>.



- [RFC4469] Resnick, P., "Internet Message Access Protocol (IMAP) CATENATE Extension", [RFC 4469](#), DOI 10.17487/RFC4469, April 2006, <<http://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, <<http://www.rfc-editor.org/info/rfc5234>>.
- [RFC5819] Melnikov, A. and T. Sirainen, "IMAP4 Extension for Returning STATUS Information in Extended LIST", [RFC 5819](#), DOI 10.17487/RFC5819, March 2010, <<http://www.rfc-editor.org/info/rfc5819>>.

## **9.2. Informative References**

- [RFC4549] Melnikov, A., Ed., "Synchronization Operations for Disconnected IMAP4 Clients", [RFC 4549](#), DOI 10.17487/RFC4549, June 2006, <<http://www.rfc-editor.org/info/rfc4549>>.

### Authors' Addresses

Jayantheesh S B  
Samsung Electronics America  
685 US Highway 202/206  
Bridgewater, NJ 08807  
USA

Email: jayantheesh.sb@gmail.com

Narendra Singh Bisht  
Samsung Electronics America  
685 US Highway 202/206  
Bridgewater, NJ 08807  
USA

Email: narendrasingh.bisht@gmail.com

