

Table of Contents

1. Introduction	2
1.1. Conventions	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 Behavior	4
4. APPEND Response	4
5. Formal Syntax	5
6. Security Considerations	5
7. IANA Considerations	5
8. References	6
8.1. Normative References	6
8.2. Informative References	6
Acknowledgements	7
Authors' Addresses	7

1. Introduction

Some IMAP servers have limits for message 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 a message with huge attachments consumes device resources like device battery power and mobile data.

The IMAP APPENDLIMIT extension provides the ability to advertise a maximum upload size allowed by the IMAP server, so that the 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

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 the APPENDLIMIT extension advertises this by including the name APPENDLIMIT in its capability list in the 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 size for mail uploads. An 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 mailboxes. The following example demonstrates the APPENDLIMIT capability with the 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, as they 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 can get an APPENDLIMIT value by either issuing a STATUS or a LIST command.

An IMAP client implementing this extension should be able to parse both mailbox-specific and global APPENDLIMIT responses. By looking at the upload size advertised by the IMAP server, a client can avoid trying to APPEND mail more than the advertised limit.

3. Mailbox-Specific APPENDLIMIT

An IMAP server can have mailbox-specific APPENDLIMIT values that will not be advertised as part of the CAPABILITY response. The IMAP server can publish specific values for each mailbox, and it 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 a STATUS command. An IMAP client should issue a STATUS command with an 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 a LIST command 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 Behavior

Computing the APPENDLIMIT should be fast and should not take Access Control Lists (ACLs), quotas, or 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, even if the message being appended is smaller than the APPENDLIMIT.

4. APPEND Response

If a client uploads a message that exceeds the maximum upload size set for that mailbox, then the server SHALL reject the APPEND command with a tagged TOOBIG response code. Refer to Section 4 of [RFC4469] for various APPEND response codes and their handling.

A client SHOULD avoid use of non-synchronizing literals [RFC7888] when the maximum upload size supported by the IMAP server is unknown. Refer to 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 of that document. [RFC3501] defines the non-terminals "capability" and "status-att", and [RFC4466] defines "status-att-val".

All alphabetic characters are case insensitive. The use of uppercase or lowercase 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

This extension provides additional information that cooperative clients can use as an optimization and does not introduce new security concerns. This extension does not address abusive clients that intend to consume server resources, and servers will still have to take action to disconnect and/or restrict access to clients that exhibit abusive behavior.

7. IANA Considerations

IANA has added "APPENDLIMIT" to the "IMAP Capabilities" registry, using this document as its reference.

8. References

8.1. Normative References

- [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>>.
- [RFC7888] Melnikov, A., Ed., "IMAP4 Non-synchronizing Literals", RFC 7888, DOI 10.17487/RFC7888, May 2016, <<http://www.rfc-editor.org/info/rfc7888>>.

8.2. 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>>.

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.

Authors' Addresses

Jayantheesh SrimushnamBoovaraghamoorthy
Samsung Electronics America
685 US Highway 202/206
Bridgewater, NJ 08807
United States

Email: jayantheesh.sb@gmail.com

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

Email: narendrasingh.bisht@gmail.com

