

Internet Engineering Task Force (IETF)
Request for Comments: 9422
Category: Standards Track
ISSN: 2070-1721

N. Freed
J. Klensin
February 2024

The LIMITS SMTP Service Extension

Abstract

This document defines a LIMITS extension for the Simple Mail Transfer Protocol (SMTP), including submission, as well as the Local Mail Transfer Protocol (LMTP). It also defines an associated limit registry. The extension provides the means for an SMTP, submission, or LMTP server to inform the client of limits the server intends to apply to the protocol during the current session. The client is then able to adapt its behavior in order to conform to those limits.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc9422>.

Copyright Notice

Copyright (c) 2024 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction
2. Terminology
3. The LIMITS SMTP Extension
 - 3.1. Limits
 - 3.2. Limit Naming Conventions
 - 3.3. Interaction with Pipelining
 - 3.4. Varying Limits
 - 3.5. Interaction with SMTP Minimums
 - 3.6. Multiple EHLO Commands
 - 3.7. Syntax Errors in the LIMITS Parameter Value
 - 3.8. Caching of Limit Settings between Sessions Involving the Same Client and Server SMTP
4. Initial Limits

4.1.	MAILMAX Limit
4.2.	RCPTMAX Limit
4.3.	RCPTDOMAINMAX Limit
5.	Example
6.	Security Considerations
7.	IANA Considerations
7.1.	SMTP Service Extension Registry
7.2.	SMTP Server Limits Registry
8.	References
8.1.	Normative References
8.2.	Informative References
	Acknowledgments
	Authors' Addresses

1. Introduction

The Simple Mail Transfer Protocol provides the ability to transfer [SMTP] or submit [SUBMIT] multiple email messages from one host to another, each with one or more recipients, using a single or multiple connections.

The Local Mail Transfer Protocol [LMTP] provides the ability to deliver messages to a system without its own mail queues. Like SMTP, it allows multiple messages with multiple recipients.

In order to conserve resources as well as protect themselves from malicious clients, it is necessary for servers to enforce limits on various aspects of the protocol, e.g., a limit on the number of recipients that can be specified in a single transaction.

Additionally, servers may also wish to alter the limits they apply depending on their assessment of the reputation of a particular client.

The variability of the limits that may be in effect creates a situation where clients may inadvertently exceed a particular server's limits, causing servers to respond with temporary (or in some cases, permanent) errors. This in turn can lead to delays or even failures in message transfer.

The LIMITS extension provides the means for a server to inform a client about specific limits in effect for a particular SMTP or LMTP session in the EHLO or LHLO command response. This information, combined with the inherent flexibility of these protocols, makes it possible for clients to avoid server errors and the problems they cause.

SMTP and LMTP servers have always been able to announce a limit using distinguished syntax in a reply, but this approach requires that the client first needs to issue a command. The mechanism specified here avoids the overhead of that approach by announcing limits prior to any substantive interaction.

Limits are registered with the IANA. Each registration includes the limit name, value syntax, and a description of its semantics.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This specification uses the Augmented Backus-Naur Form [ABNF] notation and its core rules to define the formal syntax of the LIMITS

extension.

This specification makes extensive use of the terminology specified and used in [SMTP].

3. The LIMITS SMTP Extension

The extension mechanism for SMTP is defined in Section 2.2 of the current SMTP specification [RFC5321a]. LMTP [LMTP] inherits SMTP's extension mechanism.

The name of the extension is LIMITS. Servers implementing this extension advertise a LIMITS as a keyword in the response to EHLO (LHLO for LMTP). The associated parameter is used by the server to communicate one or more limits, each with an optional value, to the client. The syntax of the parameter is:

```
limits-param = limit-name-value 0*[SP limit-name-value]
limit-name-value = limit-name ["=" limit-value]
limit-name = 1*(ALPHA / DIGIT / "-" / "_")
limit-value = 1*(%x21-3A / %x3C-7E) ; Any VCHAR except ";"
```

This extension introduces no new SMTP commands and does not alter any existing command. However, it is possible for a LIMITS parameter to be associated with another SMTP extension that does these things.

3.1. Limits

In order to achieve consistent behavior, all limits MUST be registered with the IANA, as described below.

3.2. Limit Naming Conventions

Limit names MUST be comprehensible, but also should be kept as short as possible. The use of commonly understood abbreviations, e.g., "MAX" for "maximum", is encouraged.

When a limit is associated with a particular command, its name SHOULD begin with the name of that command.

Limit names SHOULD end with one or more terms that describe the type of limit.

3.3. Interaction with Pipelining

The "Pipelining" extension [PIPELINING] is commonly used to improve performance, especially over high latency connections. Pipelining allows an entire transaction to be sent without checking responses, and in some cases it may be possible to send multiple transactions.

The use of pipelining affects the LIMITS extension in an important way: Since a pipelining client cannot check intermediate command responses without stalling the pipeline, it cannot count the number of successful versus failed responses and adjust its behavior accordingly. Limit designers need to take this into account.

For example, it may seem like it would be better to impose a limit on the number of successful RCPT TO commands as opposed to the way the RCPTMAX limit is specified in Section 4.2 below. But counting the total number of RCPT TOs is simple, whereas counting the number of successful RCPT TO stalls the pipeline.

3.4. Varying Limits

This extension provides an announcement as part of the reply to an EHLO command.

Some servers vary their limits, as a session progresses, based on their obtaining more information. This extension does not attempt to handle in-session limit changes.

3.5. Interaction with SMTP Minimums

SMTP specifies minimum values for various server sizes, limits, and timeouts [RFC5321b], e.g., servers must accept a minimum of 100 RCPT TO commands [RFC5321c]). Unfortunately, the reality is that servers routinely impose smaller limits than what SMTP requires, and when this is done it is especially important for clients to be aware that this is happening.

For this reason there is no requirement that the limits advertised by this extension comply with the minimums imposed by SMTP.

3.6. Multiple EHLO Commands

These protocols require that the EHLO command (LHLO in LMTP) be reissued under certain circumstances, e.g., after successful authentication [AUTH] or negotiation of a security layer [STARTTLS].

Servers MAY return updated limits any time the protocol requires clients to reissue the EHLO command.

Clients MUST discard any previous limits in favor of those provided by the most recent EHLO. This includes the case where the original EHLO provided a set of limits but the subsequent EHLO did not; in this case, the client MUST act as if no limits were communicated.

3.7. Syntax Errors in the LIMITS Parameter Value

Syntax errors in the basic parameter syntax are best handled by ignoring the value in its entirety; in this case, clients SHOULD proceed as if the LIMITS extension had not been used.

Syntax or other errors in the value syntax of a specific limit, including unrecognized value keywords, are best handled by ignoring that limit; in this case, the client MUST proceed as if that limit had not been specified.

It is possible that a future specification may create multiple limits that are interrelated in some way; in this case, that specification MUST specify how an error in the value syntax of one limit affects the other limits.

3.8. Caching of Limit Settings between Sessions Involving the Same Client and Server SMTP

Clients MAY cache limits determined during one session and use them to optimize their behavior for subsequent sessions. However, since servers are free to adjust their limits at any time, clients MUST be able to accommodate any limit changes that occur between sessions.

4. Initial Limits

An initial set of limits is specified in the following sections.

4.1. MAILMAX Limit

Name: MAILMAX

Value syntax: %x31-39 0*5DIGIT ; "0" not allowed, six-digit maximum

Description: MAILMAX specifies the maximum number of transactions

(MAIL FROM commands) the server will accept in a single session. The count includes all MAIL FROM commands, regardless of whether they succeed or fail.

Restrictions: None.

Security Considerations: See Section 6

4.2. RCPTMAX Limit

Name: RCPTMAX

Value syntax: %x31-39 0*5DIGIT ; "0" not allowed, six-digit maximum

Description: RCPTMAX specifies the maximum number of RCPT TO commands the server will accept in a single transaction. It is not a limit on the actual number of recipients the message ends up being sent to; a single RCPT TO command may produce multiple recipients or, in the event of an error, none.

Restrictions: None.

Security Considerations: See Section 6

4.3. RCPTDOMAINMAX Limit

Name: RCPTDOMAINMAX

Value syntax: %x31-39 0*5DIGIT ; "0" not allowed, six-digit maximum

Description: RCPTDOMAINMAX specifies the maximum number of different domains that can appear in a recipient (RCPT TO) address within a single session. This limit is imposed by some servers that bind to a specific internal delivery mechanism on receipt of the first RCPT TO command.

Restrictions: None.

Security Considerations: See Section 6

5. Example

A server announces two limits it implements to the client, along with various other supported extensions, as follows:

```
S: [wait for open connection]
C: [open connection to server]
S: 220 mail.example.com ESMTP service ready
C: EHLO example.org
S: 250-mail.example.com
S: 250-SMTPUTF8
S: 250-LIMITS RCPTMAX=20 MAILMAX=5
S: 250-SIZE 100000000
S: 250-8BITMIME
S: 250-ENHANCEDSTATUSCODES
S: 250-PIPELINING
S: 250-CHUNKING
S: 250 STARTTLS
```

The client now knows to limit the number of recipients in a transaction to twenty and the number of transactions in a session to five.

6. Security Considerations

A malicious server can use limits to overly constrain client

behavior, causing excessive use of client resources.

A malicious client may use the limits a server advertises to optimize the delivery of unwanted messages.

A man-in-the-middle attack on unprotected SMTP connections can be used to cause clients to misbehave, which in turn could result in delivery delays or failures. Loss of reputation for the client could also occur.

All that said, decades of operational experience with the SMTP "SIZE" extension [SIZE], which provides servers with the ability to indicate message size, indicates that such abuse is rare and unlikely to be a significant problem.

Use of the LIMITS extension to provide client-specific information - as opposed to general server limits - unavoidably provides senders with feedback about their reputation. Malicious senders can exploit this in various ways, e.g., start by sending good email and then, once their reputation is established, sending bad email.

7. IANA Considerations

7.1. SMTP Service Extension Registry

The IANA has added "LIMITS" to the "SMTP Service Extensions" registry:

EHLO Keyword: LIMITS

Description: Server limits

Reference: RFC 9422

Note: See "SMTP Server Limits" registry.

7.2. SMTP Server Limits Registry

The IANA has created a new registry in the "MAIL Parameters" group for SMTP server limits. The policy for this registry is "Specification Required". Registry entries consist of these required values:

1. The name of the limit.
2. The syntax of the limit value, if the limit has one. This syntax MUST conform to the provisions of Section 3 above. In most cases, the syntax will consist of a keyword designating the limit type followed by a limit value, using a "name=value" form as illustrated by the registrations created by this specification in Section 4 above. Use of ABNF [ABNF] is preferred but not required. If there is no limit value, that should be explicit in the registration request and the registry.
3. A description of the limit's semantics.
4. Restrictions, if any, on the use of the limit. If the limit is specific to any of SMTP, message submission, or LMTP, it should be documented here.
5. Security considerations for the limit.

The Designated Expert(s) appointed under the "Specification Required" procedure should check that registration requests are consistent with the requirements and intent of the extension mechanisms associated with SMTP [SMTP], Section 3 above, and the provision of this

specification more generally and offer advice accordingly.

The IANA has registered the limits specified in Section 4 of this document.

8. References

8.1. Normative References

- [ABNF] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/info/rfc5234>>.
- [PIPELINING] Freed, N., "SMTP Service Extension for Command Pipelining", STD 60, RFC 2920, DOI 10.17487/RFC2920, September 2000, <<https://www.rfc-editor.org/info/rfc2920>>.
- [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>>.
- [RFC5321a] Klensin, J., "Simple Mail Transfer Protocol", Section 2.2, RFC 5321, October 2008, <<https://www.rfc-editor.org/rfc/rfc5321>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [SMTP] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <<https://www.rfc-editor.org/info/rfc5321>>.

8.2. Informative References

- [AUTH] Siemborski, R., Ed. and A. Melnikov, Ed., "SMTP Service Extension for Authentication", RFC 4954, DOI 10.17487/RFC4954, July 2007, <<https://www.rfc-editor.org/info/rfc4954>>.
- [LMTP] Myers, J., "Local Mail Transfer Protocol", RFC 2033, DOI 10.17487/RFC2033, October 1996, <<https://www.rfc-editor.org/info/rfc2033>>.
- [RFC5321b] Klensin, J., "Simple Mail Transfer Protocol", Section 4.5.3.1, RFC 5321, October 2008, <<https://www.rfc-editor.org/rfc/rfc5321>>.
- [RFC5321c] Klensin, J., "Simple Mail Transfer Protocol", Section 4.5.3.1.8, RFC 5321, October 2008, <<https://www.rfc-editor.org/rfc/rfc5321>>.
- [SIZE] Klensin, J., Freed, N., and K. Moore, "SMTP Service Extension for Message Size Declaration", STD 10, RFC 1870, DOI 10.17487/RFC1870, November 1995, <<https://www.rfc-editor.org/info/rfc1870>>.
- [STARTTLS] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, DOI 10.17487/RFC3207, February 2002, <<https://www.rfc-editor.org/info/rfc3207>>.
- [SUBMIT] Gellens, R. and J. Klensin, "Message Submission for Mail", STD 72, RFC 6409, DOI 10.17487/RFC6409, November 2011,

<<https://www.rfc-editor.org/info/rfc6409>>.

Acknowledgments

The concept for this extension came from, and was developed by, Ned Freed and most of this specification, including substantially all of the technical details, was written by him. Unfortunately, he became ill and eventually passed away in May 2022 without being able to complete the document and manage it through IETF Last Call. His contributions to the Internet, work in the IETF, and the personal style that made both possible are irreplaceable and greatly missed. With the support of the community, John Klensin picked the document up, responded to some additional feedback, and got the document into what is believed to be a finished state. In the interest of preserving this as Ned's document, a few comments that proposed additional features and options were set aside for future work rather than our having to guess at whether Ned would have approved of them.

The acknowledgments below are divided into two parts, those written by Ned and those associated with input to the document after his passing.

* Acknowledgments from the Last Version Prepared by Ned Freed

A lot of people have helped make this specification possible. The author wishes to thank Claus Assmann, Laura Atkins, Alex Brotman, Richard Clayton, Dave Crocker, Viktor Dukhovni, Arnt Gulbrandsen, Jeremy Harris, Todd Herr, Mike Hillyer, Matthias Leisi, John Klensin, Valdis Kltneiks, John Levine, Alexey Melnikov, Keith Moore, Michael Peddemors, Hector Santos, George Schlossnagle, Rolf E. Sonneveld, and Alessandro Vesely for their contributions and reviews.

* Acknowledgments from Versions Subsequent to Ned Freed's Passing

Additional helpful suggestions were received when the draft was requeued in the last part of 2022 and thereafter. Reviews and suggestions from Alex Brotman, Dave Crocker, Gene Hightower, Murray S. Kucherawy, Barry Leiba, John Levine, and Phil Pennock were especially helpful in identifying errors and suggesting clarifying some issues with the document without changing Ned's fundamental issues or very much of his text.

IETF Last Call comments (and comments immediately before it started) from people not listed above that resulted in document changes were received from Amanda Baber (for IANA), Linda Dunbar, and Paul Kyzivat.

Authors' Addresses

Ned Freed

John C. Klensin
1770 Massachusetts Ave, Suite 322
Cambridge, MA 02140
United States of America
Email: john-ietf@jck.com