Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 19, 2015

A. Melnikov
Isode Ltd

June 17, 2015

# Simple Mail Transfer Protocol extension for relaying Metadata
## draft-melnikov-smtp-metadata-01

## Abstract

This memo defines an extension to the SMTP (Simple Mail Transfer Protocol) service whereby message metadata (such as Trace header fields, IMAP flags, Keying material, etc) can be transferred in separate containers similar to BDAT (RFC 3030, SMTP CHUNKING) command. This allows clean separation of transaction related state from the message itself.

## 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 December 19, 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

This memo defines an extension to the SMTP (Simple Mail Transfer Protocol) service whereby message metadata (such as Trace header fields, IMAP flags, Keying material, etc) can be transferred in separate containers similar to BDAT (RFC 3030, SMTP CHUNKING) command. This allows clean separation of transaction related state from the message itself.

## 2. Conventions Used in This Document

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] when they appear in ALL CAPS. These words also appear in this document in lower case as plain English words, absent their normative meanings.





The formal syntax use the Augmented Backus-Naur Form (ABNF) [RFC5234] notation including the core rules defined in Appendix B of RFC 5234 [RFC5234].

In examples, "C:" and "S:" indicate lines sent by the client and server respectively. Line breaks that do not start with a new "C:" or "S:" exist for editorial reasons and are not a part of the protocol.

## 3. Definition of the Metadata SMTP Extension

The Metadata SMTP service extension is defined as follows:

1. The textual name of this extension is "Metadata Transfer".

2. The EHLO keyword value associated with this extension is "METADATA". Any server that advertises support for the "METADATA" extension MUST also support SMTP CHUNKING (RFC 3030).

3. The EHLO keyword has no parameters

4. [[CREF1: Should BMTD be allowed before the DATA command? There is no reason why not.]] A new SMTP verb, BMTD, is defined. The BMTD verb takes one argument, which indicates the length, in octets, of the binary metadata container that follows immediately after the command. See Section 3.1 for the description of the BMTD command and Section 6 for its syntax.

5. This extension doesn't add any new parameters to MAIL FROM or RCPT TO commands.

6. The Metadata extension is valid for the submission service [RFC6409] and LMTP [RFC2033].

### 3.1. BMTD command

After all MAIL and RCPT responses are collected and processed, the message metadata is sent using a series of BMTD commands. The BMTD command takes one required argument, the exact length of the metadata segment ("container") in octets. The metadata is sent immediately after the trailing <CR> <LF> of the BMTD command line. Once the receiver-SMTP receives the specified number of octets, it will return a 250 reply code.

BMTD commands MUST be sent before any BDAT [RFC3030] or BURL [RFC4468] commands. If a server encounters BMTD command after BDAT/ BURL, it MUST respond with 503 "Bad sequence of commands" reply code.





The state resulting from this error is indeterminate. A RSET command MUST be sent to clear the transaction before continuing.

Each BMTD container starts with 2 octet container type, followed by container type specific data. This means that the metadata segment length can never be the value 1 (it can either be 0 or be equal or greater than 2).

A 250 response MUST be sent to each successful BMTD data block ("chunk") within a mail transaction. If a failure occurs after a BMTD command is received, the receiver-SMTP MUST accept and discard the associated metadata and message data before sending the appropriate 5XX or 4XX code. If a 5XX or 4XX code is received by the sender-SMTP in response to a BMTD chunk, the transaction should be considered failed and the sender- SMTP MUST NOT send any additional BMTD segments. If the receiver- SMTP has declared support for command pipelining [RFC2920], the receiver SMTP MUST be prepared to accept and discard additional BDAT/BURL/BMTD chunks already in the pipeline after the failed BMTD.

Note: An error on the receiver-SMTP such as disk full or imminent shutdown can only be reported after the BMTD segment has been received. It is therefore important to choose a reasonable chunk size given the expected end-to-end bandwidth.

Note: Because the receiver-SMTP does not acknowledge the BMTD command before the message data is sent, it is important to send the BMTD only to systems that have declared their capability to accept BMTD commands. Illegally sending a BMTD command and associated message

data to a non-METADATA capable system will result in the receiver-SMTP parsing the associated message data as if it were a potentially very long, ESMTP command line containing binary data.

More than one BMTD command can occur in a transaction. (However some BMTD container types only allow for a single BMTD command with that particular container type.) Any BMTD command MUST be followed by one or more of BMTD/BDAT/BURL commands.

## 3.2. Initial List of Metadata Container types

Type 0: Trace header fields: Received, Return-Path, Authentication-Results (RFC 7001), etc encoded as if they are a part of a message header. Containers of this type can appear multiple types in a transaction. MUST be supported by all compliant servers.

Type 1: IMAP Keywords [RFC3501] associated with the message (e.g. $MDNSent, $Forwarded, \Answered). This is a space separated list of IMAP keywords/flags. Container of this type MUST NOT appear more





than once in a transaction. If the final LMTP delivers the message to an IMAP capable mailstore, it MUST attempt setting the listed IMAP keywords/flags on the message. Flags/keywords not supported by the mailstore (or disallowed when a message is injected via LMTP) MUST be ignored.

Keying material, a la Dark Mail. TBD if there is interest.

## 3.3. Requirements on a Metadata Container type definition

Each container type definition MUST specify if it can appear more than once.

Unless specified by an extension mutually agreed by SMTP sender and SMTP recipient, no container type can be defined as required (i.e. appearing at least once in a SMTP transaction) or define how it can be relayed to a non compliant MTA.

Each container type definition MUST describe how it is going to be handled by the final MTA/LMTP server.

# 4. Handling of messages received via SMTP

This section describes how a conforming SMTP server should handle any messages received via SMTP.

## 4.1. Relay of messages to other conforming SMTP/LMTP servers

The following rules govern the behavior of a conforming MTA (in the role of an SMTP/LMTP client), when relaying a message which was received via the SMTP protocol, to an SMTP/LMTP server that supports the METADATA extension:

1. Instead of prepending trace fields to the message itself as specified in RFC 5321, a relaying MTA SHOULD [[CREF2: Cross check with RFC 5321 regarding insertion of Received header fields]] insert a single BMTD container of type 0 (Trace fields) containing its own trace header fields such as Received [RFC5321], Authentication-Results [RFC7001], etc.

2. All other BMTD commands are relayed to conforming SMTP/LMTP server in the order received. Intermediary servers SHOULD NOT coalesce or reorder metadata containers of type 0 or any other type that they understand. Intermediary servers MUST NOT coalesce, reorder or drop metadata containers of any types that they don't recognize.





## 4.2. Relay of messages to non-conforming SMTP/LMTP servers

The following rules govern the behavior of a conforming MTA (in the role of an SMTP/LMTP client), when relaying a message which was received via the SMTP protocol, to an SMTP/LMTP server that does not support the METADATA extension:

1. Data from each metadata container of type 0 (Trace fields) MUST be extracted and prepended to the header of the message in the order of BMTD commands.

2. All other BMTD chunks are discarded. [[CREF3: OPEN ISSUE. They can also be converted to some magic header fields for logging and debugging?]]

## 4.3. Gatewaying a message into a foreign environment

The following rules govern the behavior of a conforming MTA, when gatewaying a message that was received via the SMTP protocol, into a foreign (non-SMTP) environment:

1. If the destination environment is unable to provide an equivalent of the BMTD command, the conforming MTA SHOULD behave as if it is relaying to a non-conformant SMTP server (Section 4.2).

2. If the destination environment is capable of providing an equivalent of the BMTD command, the conforming MTA SHOULD behave as if it is relaying to a conformant SMTP server (Section 4.1), converting any BMTD command to the equivalent in the destination environment.

# 5. Use of METADATA with LMTP

An LMTP server can advertise support for the METADATA extension:

1. Data from containers of type 0 (Trace fields) is extracted (in the order of the corresponding BMTD commands) and prepended to the header of the message.

2. Handling of other container type is specific to the container type.

3. Unsupported BMTD container types are discarded. [[CREF4: OPEN ISSUE. They can also be converted to some magic header fields for logging and debugging?]]





# 6. Syntax

```
metadata-ehlo  = "METADATA"
          ; Complies with the <ehlo-line> ABNF production from RFC 5321.

bmtd-cmd       = "BMTD" SP chunk-size CR LF
chunk-size     = 1*DIGIT
```

```
bmtd-container = container-type container-specific-data

container-type = <2 octets, extensible>

container-specific-data = <remaining container data>

DIGIT = <Defined in RFC 5234>
```

## 7. Example

The original submission (from MUA to MSA) might look like shown below. Note that the example is also making use of the STARTTLS [RFC3207], and DSN [RFC3461] SMTP extensions, even though there is no requirement that these other extensions are to be supported when the METADATA SMTP extension is implemented.





```
S: 220 example.com SMTP server here
```

```
C: EHLO mua.example.com
S: 250-example.com
S: 250-STARTTLS
S: 250-AUTH SCRAM-SHA-1 DIGEST-MD5
S: 250-DSN
S: 250-CHUNKING
S: 250-ENHANCEDSTATUSCODES
S: 250 METADATA
C: AUTH SCRAM-SHA-1
[...authentication exchange...]
S: 235 2.7.0 Authentication successful
C: MAIL FROM:<eljefe@example.com> ENVID=QQ314159
S: 250 2.1.0 <eljefe@example.com> sender ok
C: RCPT TO:<topbanana@example.net>
S: 250 2.1.5 <topbanana@example.net> recipient ok
C: RCPT TO:<Dana@Ivory.example.net> NOTIFY=SUCCESS,FAILURE
    ORCPT=rfc822;Dana@Ivory.example.net
S: 250 2.1.5 <Dana@Ivory.example.net> recipient ok
C: BMTD 40
C: <2 octets == type> ...
S: 250 2.1.0 message metadata accepted
C: BMTD 12
C: <2 octets == type 1>$Forwarded
S: 250 2.1.0 message metadata accepted
C: BDAT 86 LAST
C: To: Susan@random.com
C: From: Sam@random.com
C: Subject: This is a bodyless test message
S: 250 2.1.0 message accepted
C: QUIT
S: 221 2.0.0 goodbye
```

[[Need to fix byte counts/BMTD commands in the example]]

[[Add another example with PIPELINING]]

# 8. Deployment Considerations

## 8.1. Multiple MX records

If multiple DNS MX records are used to specify multiple servers for a domain in section 5 of [RFC5321], it is strongly advised that all of them support the METADATA extension . If one or more servers behave differently in this respect, then it is strongly suggested that none of the servers support the METADATA extension. Otherwise, unexpected differences in message rejections can happen during temporary or

permanent failures, which users might perceive as serious reliability issues.

## 9. Open Issues/To Do

Document interaction with the SIZE extension. (Proposal: count each BMTD chunk size against the SIZE limit)

Decide what should be allowed behaviour for handling of container types unrecognized by intermediate server and final delivery agents.

## 10. IANA Considerations

This specification requests IANA to add the METADATA SMTP extension to the "SMTP Service Extensions" registry (in http://www.iana.org/assignments/mail-parameters). This extension is suitable for the Submit port.

## 11. Security Considerations

TBD

## 12. References

### 12.1. Normative References

[RFC1870] Klensin, J., Freed, N., and K. Moore, "SMTP Service Extension for Message Size Declaration", STD 10, RFC 1870, November 1995.

[RFC2033] Myers, J., "Local Mail Transfer Protocol", RFC 2033, October 1996.

[RFC2034] Freed, N., "SMTP Service Extension for Returning Enhanced Error Codes", RFC 2034, October 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2920] Freed, N., "SMTP Service Extension for Command Pipelining", STD 60, RFC 2920, September 2000.

[RFC3030] Vaudreuil, G., "SMTP Service Extensions for Transmission of Large and Binary MIME Messages", RFC 3030, December 2000.

[RFC3461] Moore, K., "Simple Mail Transfer Protocol (SMTP) Service Extension for Delivery Status Notifications (DSNs)", RFC 3461, January 2003.

[RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003.

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, October 2008.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5248] Hansen, T. and J. Klensin, "A Registry for SMTP Enhanced Mail System Status Codes", BCP 138, RFC 5248, June 2008.

[RFC6409] Gellens, R. and J. Klensin, "Message Submission for Mail", STD 72, RFC 6409, November 2011.

[RFC7001] Kucherawy, M., "Message Header Field for Indicating Message Authentication Status", RFC 7001, September 2013.

## 12.2. Informative References

[RFC5598] Crocker, D., "Internet Mail Architecture", RFC 5598, July 2009.

[RFC1845] Crocker, D. and N. Freed, "SMTP Service Extension for Checkpoint/Restart", RFC 1845, September 1995.

[RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, February 2002.

[RFC4468] Newman, C., "Message Submission BURL Extension", RFC 4468, May 2006.

[RFC5228] Guenther, P. and T. Showalter, "Sieve: An Email Filtering Language", RFC 5228, January 2008.





## Appendix A. Background on Design Choices

This Section provides some background on design choices made during development of the METADATA SMTP extension.

Use of a new command like BDAT makes it very easy to send chunks of binary data. Byte counted blobs are easy to parse and generate.

## Appendix B. Acknowledgements

The idea suggested in this document is not new. John Klensin and Paul Smith have suggested use of an SMTP extension for separating metadata from the rest of email messages. Thank you to Chris Newman for providing comments and suggesting how to make the extension easier to implement. This document was also inspired by the Dark Mail project.

This document cuts & pastes lots of text from RFC 3030.

## Author's Address

Alexey Melnikov
Isode Ltd
5 Castle Business Village
36 Station Road
Hampton, Middlesex TW12 2BX
UK

EMail: Alexey.Melnikov@isode.com