More Instant Messaging Interoperability R. Mahy
Internet-Draft 2 March 2026
Intended status: Informational
Expires: 3 September 2026

A Message Status format for the More Instant Messaging Interoperability (MIMI) content format
draft-mahy-mimi-message-status-01

Abstract

The More Instant Messaging Interoperability (MIMI) content format describes a message format for instant messaging. This specification defines a concise, efficient format for communicating status of messages sent using MIMI content.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://rohanmahy.github.io/mimi-message-status/draft-mahy-mimi-message-status.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-mahy-mimi-message-status/.

Discussion of this document takes place on the More Instant Messaging Interoperability Working Group mailing list (mailto:mimi@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/mimi/. Subscribe at https://www.ietf.org/mailman/listinfo/mimi/.

Source for this draft and an issue tracker can be found at https://github.com/rohanmahy/mimi-message-status.

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 https://datatracker.ietf.org/drafts/current/.

Copyright Notice

Table of Contents

1. Introduction

This document describes the semantics of a status report of MIMI content format [I-D.ietf-mimi-content] messages. Because some messaging systems deliver messages in batches and allow a user to mark several messages read at a time, the report format allows a single report to convey the read/delivered status of multiple messages (by message ID) within the same MIMI room at a time. This specification defines a concise, efficient format for communicating status of messages sent using MIMI content. It could also represent messages sent using other messaging formats that have similar per-

message unique message IDs and security characteristics.

2. Conventions and Definitions

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 document uses the term "room" as defined in [I-D.ietf-mimi-arch].

3. Delivery Reporting and Read Receipts

In instant messaging systems, read receipts typically generate a distinct indicator for each message. In some systems, the number of users in a group who have read the message is subtly displayed and the list of users who read the message is available on further inspection.

Of course, Internet mail has support for read receipts as well, but the existing message disposition notification mechanism defined for email in [RFC8098] is completely inappropriate in this context:

* notifications can be sent by intermediaries
* only one notification can be sent about a single message per recipient
* a human-readable version of the notification is expected
* each notification can refer to only one message
* it is extremely verbose

Instead, we would like to be able to include status changes about multiple messages in each report, the ability to mark a message delivered, then read, then unread, then expired for example.

The format, like the MIMI content format, uses Common Binary Object Representation (CBOR) [RFC8949] encoding. It has the media type application/mimi-message-status. It is sent by individual members of a chat room and can refer to multiple messages sent in the same room in a single message. The format contains a list of message ID / status pairs. As the status at the recipient changes, the status can be updated in subsequent notification.

The status of each message can be one of the following values:

* 0 (unread) indicates that the message was not yet read by the sender of the report.

* 1 (delivered) indicates that a messaging client of the sender of the report received the message.

* 2 (read) indicates that the sender of the report read the message.

* 3 (expired) indicates that the message expired and is not available for reading. In the case of absolute expiration, it does not indicate if the message was read before its expiry.

* 4 (deleted) indicates that the message was deleted, either by the local client, or by another member of the room with the power to retract messages.

* 5 (hidden) indicates that the message was hidden by the local client (for example archived).

* 6 (error) indicates that the sender client is aware of the message ID, but that there was an unspecified error with the reception of the message.

Not every state is relevant for every type of message, and it does not make sense to transition from any one state to any other state. For example, a transition from deleted to delivered does not make sense. The implementer of this format needs to decide which state transitions are meaningful given their implementation and its available policy options.

Depending on the policy of the room and a potential sender of delivery reports, sending delivery receipts and/or read receipt messages might be required, optional, or forbidden. Clients might also have policies about specific status values that are shared and others that are not. Some status values might only be shared among the reporting user’s own clients, for example.

4. Formal Data Definition

Below is the Concise Data Definition Language (CDDL) [RFC8610] definition for the message status format.

```
MessageStatusReport = [ * PerMessageStatus ]

PerMessageStatus = [
    messageId: MessageId,
    status: baseStatus / $extStatus / unknownStatus
]

baseStatus = &(
    unread: 0,
    delivered: 1,
    read: 2,
    expired: 3,
    deleted: 4,
    hidden: 5,
    error: 6
)
unknownStatus = &( unknown: 7..255 )

MessageId = bstr .size 32
```

Figure 1: CDDL for MIMI message status format

5. Message Status Format Example

The following example message assumes the sender user handle URL is mimi://example.com/u/bob-jones.

It uses the example message names and message IDs from Section 5 of the MIMI content [I-D.ietf-mimi-content].

```
[
  [
    /  Original message     /
    h'017ce54837404c3696e0c747b985cb17
      2716d0ed0a3d249ca63ace7d82a096f4',
    2    / status = read    /
  ],
  [
    /  Reply message        /
    h'015354973c2b65ca937bf1e035ae53a5
      ab80e947afa43d46920d4202e5cc0b27',
    2    / status = read    /
  ],
  [
    /  Mention message      /
    h'018d825adf9f6be00dcafc5704c4102f
      5022e74219d0b603e4ba7622654042af',
    0    / status = unread  /
  ],
  [
    /  Expiring message     /
    h'01e59db8173939facc2c8a4a0f0ae8d0
      c7a11a81239626630c9464a8d6717a03',
    3    / status = expired /
  ]
]
```

Figure 2: Example message report

A CBOR pretty printed hexadecimal version is shown below:

```
84                                      # array(4)
   82                                   # array(2)
      58 20                             # bytes(32)
         017ce54837404c3696e0c747b985cb17
         2716d0ed0a3d249ca63ace7d82a096f4
      02                                # unsigned(2)
   82                                   # array(2)
      58 20                             # bytes(32)
         015354973c2b65ca937bf1e035ae53a5
         ab80e947afa43d46920d4202e5cc0b27
      02                                # unsigned(2)
   82                                   # array(2)
      58 20                             # bytes(32)
         018d825adf9f6be00dcafc5704c4102f
         5022e74219d0b603e4ba7622654042af
      00                                # unsigned(0)
   82                                   # array(2)
      58 20                             # bytes(32)
         01e59db8173939facc2c8a4a0f0ae8d0
         c7a11a81239626630c9464a8d6717a03
      03                                # unsigned(3)
```

6. Security Considerations

Delivery and Read Receipts can provide useful information inside a group, or they can reveal sensitive private information. In many IM systems there are per-group policies for read receipts (and/or delivery notifications):

* they are required

* they are permitted, but optional

* they are forbidden

In the first case, everyone in the group would have to claim to support read receipts to be in the group and agree to the policy of sending them whenever a message was read. A user who did not wish to send read receipts could review the policy (automatically or manually) and choose not to join the group. Of course, requiring read receipts is a cooperative effort just like using self-deleting messages. A malicious client could obviously read a message and not send a read receipt, or send a read receipt for a message that was never rendered. However, cooperating clients have a way to agree that they will send read receipts when a message is read in a specific group.

In the second case, sending a read receipt would be at the discretion of each receiver of the message (via local preferences).

7. IANA Considerations

RFC EDITOR: Please replace XXXX throughout with the RFC number assigned to this document.

7.1. MIME subtype registration of application/mimi-message-status

This document proposes registration of a media subtype with IANA.

Type name: application

Subtype name: mimi-message-status

Required parameters: none

Optional parameters: none

Encoding considerations:
This message type should be encoded as binary data

Security considerations:
See Section 6 of RFC XXXX

Interoperability considerations:
See Section 3 of RFC XXXX

Published specification: RFC XXXX

Applications that use this media type:
Instant Messaging Applications

Fragment identifier considerations: N/A

Additional information:

Deprecated alias names for this type: N/A
Magic number(s): N/A
File extension(s): N/A
Macintosh file type code(s): N/A

Person & email address to contact for further information:
IETF MIMI Working Group mimi@ietf.org

8. References

8.1. Normative References

[I-D.ietf-mimi-arch]
Barnes, R., "An Architecture for More Instant Messaging Interoperability (MIMI)", Work in Progress, Internet-Draft, draft-ietf-mimi-arch-02, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-mimi-arch-02>.

[I-D.ietf-mimi-content]
Mahy, R., "More Instant Messaging Interoperability (MIMI) message content", Work in Progress, Internet-Draft, draft-ietf-mimi-content-07, 7 July 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-mimi-content-07>.

[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/rfc/rfc2119>.

[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/rfc/rfc8174>.

[RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <https://www.rfc-editor.org/rfc/rfc8610>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/rfc/rfc8949>.

8.2. Informative References

[RFC8098] Hansen, T., Ed. and A. Melnikov, Ed., "Message Disposition Notification", STD 85, RFC 8098, DOI 10.17487/RFC8098, February 2017, <https://www.rfc-editor.org/rfc/rfc8098>.

Author’s Address

Rohan Mahy
Email: rohan.mahy@gmail.com