XEP-0333: Chat Markers

Spencer MacDonald
mailto:im@spencermacdonald.com
xmpp:im@spencermacdonald.com

2017-09-11
Version 0.3

Status	Type	Short Name
Deferred	Standards Track	NOT_YET_ASSIGNED

This specification describes a solution of marking the last received, displayed and acknowledged message in a chat.

Legal

Copyright

Permissions

Warranty

Liability

Conformance

Contents

1 Introduction

The concept of delivery and read receipts has been popularised by other messaging services such as iMessage, Google Hangouts and Blackberry Messenger. These services provide a visual indication of when a message has been delivered to any of the recipients resources and (optionally) when it has been read. These visual indications (referred to herein as "Chat Markers") are synced between all of the sender's and recipient's resources automatically, so the state of a chat is always consistent. If one or more of the resources is not connected, it can fetch Chat Markers from the Message Archive upon reconnecting.

Message Delivery Receipts (XEP-0184) [1] currently provides delivery receipts on a per message basis, but it does not provide any mechanism for the user to indicate that they have read or acknowledged the message. As delivery receipts are sent on a per message basis it would require multiple messages to "sync up" up delivery receipts between resources.

Moreover by using Chat State Notifications (XEP-0085) [2] you could infer that a message has been read by the recipient, if they become active at any point after the message has been delivered, but again it would require multiple messages to "sync up" chat states between resources.

This XEP outlines an efficient message based protocol to provide this functionally using Chat Markers.

Note: Chat Markers do not mark each individual message, nor do they assume a reliable transport. This means that Chat Markers can only provide a heuristic solution, but this is often satisfactory for the majority of use cases.

2 Terminology

The term "active chat" refers to a chat that a user is currently active in. See Chat State Notifications (XEP-0085) [3].

The term "system notification" refers to a notification (typically a preview) that is displayed separately to a Chat.

The term "read" in the context of iMessage, Google Hangouts and Blackberry Messenger, directly maps to the displayed element in the Chat Marker namespace.

The term "markable message" refers to the stanza for which the original sender would like to receive a Chat Marker.

The term "Chat Marker" refers to the stanza by which the recipient replies to a "markable message" with a marker.

[1] XEP-0184: Message Delivery Receipts <https://xmpp.org/extensions/xep-0184.html>.
[2] XEP-0085: Chat State Notifications <https://xmpp.org/extensions/xep-0085.html>.
[3] XEP-0085: Chat State Notifications <https://xmpp.org/extensions/xep-0085.html>.

3 Requirements

This document addresses the following requirements:

1. Enable a client to mark a message as markable.
2. Enable a client to mark the last received message in a chat.
3. Enable a client to mark the last displayed message in a chat.
4. Enable a client to mark the last acknowledged message in a chat.
5. Enable a client to fetch and set Chat Markers regardless of wether the other users in a chat are online.

4 Determining support

If an entity supports the Chat Markers protocol, it MUST report that by including a Service Discovery (XEP-0030) [4] feature of "urn:xmpp:chat-markers:0" in response to disco#info requests:

Listing 1: Client queries for server features

```
<iq type='get' id='disco1' to='capulet.lit' from='juliet@capulet.lit/
    balcony'>
  <query xmlns='http://jabber.org/protocol/disco#info'/>
</iq>
```

Listing 2: Entity responds with features

```
<iq type='result' id='disco1' from='capulet.lit' to='juliet@capulet.
    lit/balcony'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    ...
    <feature var='urn:xmpp:chat-markers:0'/>
    ...
  </query>
</iq>
```

Support can also be determined via Entity Capabilities (XEP-0115) [5], a.k.a. "caps".

[4] XEP-0030: Service Discovery <https://xmpp.org/extensions/xep-0030.html>.
[5] XEP-0115: Entity Capabilities <https://xmpp.org/extensions/xep-0115.html>.

5 When to send Chat Markers and markable messages

5.1 Bare JID

If the sender knows only the recipient's bare JID, it cannot determine (via Service Discovery (XEP-0030) [6] or Entity Capabilities (XEP-0115) [7]) whether the intended recipient supports the Chat Markers protocol. In this case, the sender MAY send a Chat Marker or markable message.

5.2 Full JID

If the sender knows a full JID for the recipient (e.g., via presence), it SHOULD attempt to determine (via service disco or entity capabilities) whether the client at that full JID supports the Chat Markers protocol before attempting to send a Chat Marker or markable message.
If the sender determines that the recipient's client does not support the Chat Markers protocol then it SHOULD NOT send Chat Markers or markable messages.
If the sender determines that the recipient's client supports the Chat Markers protocol then it MAY send a Chat Marker or markable message to that full JID.

5.3 Chat Marker

To prevent looping, an entity MUST NOT send a Chat Maker to mark up to a Chat Marker.

6 Protocol Format

Chat Markers use a dedicated protocol extension qualified by the 'urn:xmpp:chat-markers:0' namespace.
There are 4 allowable elements in the namespace, the first 'markable' indicates that a message can be marked with a Chat Marker and is therefore a "markable message".
The 3 other allowable elements in this namespace are used to mark a message (in order of significance):

- received -- the message has been received by a client.
- displayed -- the message has been displayed to a user in a active chat and not in a system notification.
- acknowledged -- the message has been acknowledged by some user interaction e.g. pressing an acknowledgement button.

[6] XEP-0030: Service Discovery <https://xmpp.org/extensions/xep-0030.html>.
[7] XEP-0115: Entity Capabilities <https://xmpp.org/extensions/xep-0115.html>.

The Chat Marker MUST have an 'id' which is the 'id' of the message being marked.
The Chat Marker message stanza MUST have a 'thread' child element if the message has been received, displayed or acknowledged in the context of a thread.
A Chat Marker Indicates that all messages up to and including that message 'id' have been marked. If a thread is supplied, a Chat Marker is only valid in the context of that thread.

Listing 3: Example Content Message with a markable Chat Marker

```
<message
    from='northumberland@shakespeare.lit/westminster'
    id='message-1'
    to='kingrichard@royalty.england.lit/throne'>
  <thread>sleeping</thread>
  <body>My lord, dispatch; read o'er␣these␣articles.</body>
␣␣<markable␣xmlns='urn:xmpp:chat-markers:0'/>
</message>
```

Note: A sender MUST include an 'id' attribute on every markable message.
If recipient does not support the Chat Markers protocol it SHOULD NOT return an error.

Listing 4: Example Message marked with a recieved Chat Marker

```
<message
    from='kingrichard@royalty.england.lit/throne'
    id='message-2'
    to='northumberland@shakespeare.lit/westminster'>
  <thread>sleeping</thread>
  <received xmlns='urn:xmpp:chat-markers:0'
            id='message-1'/>
</message>
```

When the recipient sends a Chat Marker, it SHOULD ensure that the message stanza contains only the Chat Marker child element and optionally (when appropriate) a thread child element. Naturally, intermediate entities might add other extension elements to the message when routing or delivering the receipt message, e.g., a <delay/> element as specified in Delayed Delivery (XEP-0203) [8].

7 Requirements

Clients SHOULD use Message Carbons (XEP-0280) [9] to support multiple online resources.
Clients SHOULD use Message Archiving (XEP-0136) [10] or Message Archive Management

[8] XEP-0203: Delayed Delivery <https://xmpp.org/extensions/xep-0203.html>.
[9] XEP-0280: Message Carbons <https://xmpp.org/extensions/xep-0280.html>.
[10] XEP-0136: Message Archiving <https://xmpp.org/extensions/xep-0136.html>.

(XEP-0313) [11] to support offline updating of Chat Markers. Chat Markers SHOULD be archived, so they can be fetched and set regardless of whether the other users in a chat are online.
Messages MUST have an 'id' to use Chat Markers.
Messages MUST include the 'markable' element to use Chat Markers.
Chat Markers MUST only move forward. If a Chat Maker is received for an earlier message than the current Chat Marker, it MUST be ignored by the client.
Chat Markers for unknown messages MUST be ignored by the client. A client MAY store the Chat Marker incase the associated message is retrieved later.

8 Business Rules

8.1 Optimizations

Less Significant Chat Markers SHOULD only be sent if they are later than the more significant Chat Marker i.e. if a Message has been marked as displayed, a received Chat Marker should only be sent if it has a later timestamp than the displayed Chat Marker.
To avoid sending redundant Chat Markers while retrieving archived messages, Chat Markers SHOULD only be sent after retrieving the most recent message for a chat.
Only Messages that can be displayed in a chat SHOULD be markable.

8.2 Never Auto Acknowledge

Clients MUST NOT mark a message as acknowledged without any user interaction.

8.3 Marking Sent Messages

Clients MAY mark a sent or received message, as Chat Markers are inclusive of of both previously sent and received messages.

8.4 Interaction with Delivery Receipts

Chat Markers MAY be used alongside Delivery Receipts.

8.5 Interaction with Chat States

Chat Markers MAY be used alongside Chat States.

[11] XEP-0313: Message Archive Management <https://xmpp.org/extensions/xep-0313.html>.

9 Security Considerations

A user may not wish to disclose that they have received, displayed or acknowledge a message. It is possible for a sender to leak its presence when updating Chat Markers; therefore, a sender SHOULD NOT send Chat Markers to recipients who are not otherwise authorized to view its presence.

10 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA).

11 XMPP Registrar Considerations

11.1 Protocol Namespaces

This specification defines the following XML namespace:

- 'urn:xmpp:chat-markers:0'

12 XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>

<xs:schema attributeFormDefault="unqualified"
  elementFormDefault="qualified"
  targetNamespace="urn:xmpp:chat-markers:0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-0333: http://xmpp.org/extensions/xep-0333.html
    </xs:documentation>
  </xs:annotation>

  <xs:element name="markable">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
```

```
  </xs:element>

  <xs:element name="received">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute type="xs:string" name="id"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

  <xs:element name="displayed">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute type="xs:string" name="id"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>


  <xs:element name="acknowledged">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute type="xs:string" name="id"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

</xs:schema>
```