XEP-0308: Last Message Correction

Kevin Smith
mailto:kevin@kismith.co.uk
xmpp:kevin@doomsong.co.uk

2022-05-17
Version 1.2.1

Status	Type	Short Name
Draft	Standards Track	message-correct

This specification defines a method for indicating that a message is a correction of the last sent message.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2024 by the XMPP Standards Foundation (XSF).

Permissions

Permission is hereby granted, free of charge, to any person obtaining a copy of this specification (the "Specification"), to make use of the Specification without restriction, including without limitation the rights to implement the Specification in a software program, deploy the Specification in a network service, and copy, modify, merge, publish, translate, distribute, sublicense, or sell copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the condition that the foregoing copyright notice and this permission notice shall be included in all copies or substantial portions of the Specification. Unless separate permission is granted, modified works that are redistributed shall not contain misleading information regarding the authors, title, number, or publisher of the Specification, and shall not claim endorsement of the modified works by the authors, any organization or project to which the authors belong, or the XMPP Standards Foundation.

Warranty

NOTE WELL: This Specification is provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall the XMPP Standards Foundation or any author of this Specification be liable for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising from, out of, or in connection with the Specification or the implementation, deployment, or other use of the Specification (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if the XMPP Standards Foundation or such author has been advised of the possibility of such damages.

Conformance

This XMPP Extension Protocol has been contributed in full conformance with the XSF's Intellectual Property Rights Policy (a copy of which can be found at <https://xmpp.org/about/xsf/ipr-policy> or obtained by writing to XMPP Standards Foundation, P.O. Box 787, Parker, CO 80134 USA).

Contents

1   Introduction

When sending a message, people often introduce typing errors and send a follow-up message to correct them. This specification allows the sending client to flag the second message as correcting the first.

2   Discovering support

If a client implements message correction, it MUST specify the 'urn:xmpp:message-correct:0' feature in its service discovery information features as specified in Service Discovery (XEP-0030) [1] and the Entity Capabilities profile specified in Entity Capabilities (XEP-0115) [2].

Listing 1: Client requests information about a chat partner's client

```
<iq type='get'
    from='romeo@montague.net/orchard'
    to='juliet@capulet.net/balcony'
    id='info1'>
  <query xmlns='http://jabber.org/protocol/disco#info'/>
</iq>
```

Listing 2: Partner's client advertises support for correction

```
<iq type='get'
    to='romeo@montague.net/orchard'
    from='juliet@capulet.net/balcony'
    id='info1'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
...
    <feature var='urn:xmpp:message-correct:0'/>
...
  </query>
</iq>
```

Message corrections sent to clients that do not support them will be rendered as duplicate (corrected) messages. In most Instant Messaging environments (particularly within a Multi-User Chat (XEP-0045) [3] room, but also with a recipient having multiple clients using Message Carbons (XEP-0280) [4] and Message Archive Management (XEP-0313) [5]) it is unknown whether some or all receiving devices support this extension. It is suggested that a client should always allow sending corrections, but may make the user aware of the potential for duplicate

---

[1] XEP-0030: Service Discovery <https://xmpp.org/extensions/xep-0030.html>.
[2] XEP-0115: Entity Capabilities <https://xmpp.org/extensions/xep-0115.html>.
[3] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.
[4] XEP-0280: Message Carbons <https://xmpp.org/extensions/xep-0280.html>.
[5] XEP-0313: Message Archive Management <https://xmpp.org/extensions/xep-0313.html>.

messages to be interpreted by the recipients. In restricted environments, implementors could choose to allow or disallow sending in such cases, as is appropriate for the intended deployments.

3 Use Case

When a user indicates to the client that he wants to correct the most recently sent message to a contact, the client will resend the corrected message with a new id, and with the replace payload refering to the previous message by id. The receiving client then treats the newly received payloads as completely replacing all payloads of the original message.

Listing 3: User sends a message with a mistake in

```
<message to='juliet@capulet.net/balcony' id='bad1'>
  <body>But soft, what light through yonder airlock breaks?</body>
</message>
```

Listing 4: User corrects the message and sends

```
<message to='juliet@capulet.net/balcony' id='good1'>
  <body>But soft, what light through yonder window breaks?</body>
  <replace id='bad1' xmlns='urn:xmpp:message-correct:0'/>
</message>
```

4 Business Rules

The 'id' attribute is included on the replace to prevent situations where messages being routed to a different resource than the intended cause incorrect replacements.

A receiving client can choose to replace the previous message in whatever display is used for messages, or in any stored history, or can choose to display the correction in another way.

A client SHOULD alert the user that the displayed message has been edited since it was originally sent.

Clients MUST send ids on messages if they allow the user to correct messages.

To deal with multiple payloads, the sender MUST re-send the entire stanza, only altering id and the payloads being corrected and adding the 'replace' payload. It is expected that the receiver SHOULD then treat the new stanza as complete replacement for all the payloads received in the original stanza.

The Sender MUST NOT include a correction for a message with non-messaging payloads. For example, a sender MUST NOT include a correction for a roster item exchange request or a file transfer part.

A single message may be corrected multiple times by subsequent edits.

A correction MUST only be allowed when both the original message and correction originate

from the same sender[6].

While it's not possible to prevent this protocol from being used in such a way, it is not intended that it provides a way to continue expanding a previous message indefinitely and clients, in as much as it is sensible, should present use of this extension only for correction rather than for providing a continuous stream, for which In-Band Real Time Text (XEP-0301) [7] can be used instead.

If the same message is to be corrected several times, the id of the original message is used in each case (e.g. If a message of id 1 is corrected by a message of id 2, a subsequent correction should correct 1, not 2).

Correction MUST only be used to change the logical content details of a stanza (e.g. the message body) and not to change the nature of the stanza or its metadata (e.g. correction MUST NOT be used to turn a chat message into a pubsub notification). Where a correction stanza includes metadata, for example stanza or MAM ids, these are properties of the correction stanza, and do not correct aspects of the original stanza.

While it is possible to use this protocol to correct messages older than the most recent received from a full JID, such use is out of scope for this document and support for this SHOULD NOT be assumed without further negotiation.

5 Security Considerations

The replacement message could have an entirely different meaning from the original message, so clients will need to make users aware that the displayed message has been edited. It is also suggested that clients make the original message available in some way, although this UI consideration is out of the scope of this document.

There exist some payload types where correction is problematic, for some deployments. In particular, care needs to be taken that security labels from Security Labels in XMPP (XEP-0258) [8] are handled in an appropriate manner for the security domains of a given deployment.

When used in a Multi-User Chat (XEP-0045) [9] context, corrections must not be allowed (by the receiver) for messages received before the sender joined the room - particularly a full JID leaving the room then rejoining and correcting a message SHOULD be disallowed, as the entity behind the full JID in the MUC may have changed.

---

[6]In direct conversations, this means the bare-JID must match the original bare-JID, in MUCs and MUC-PMs the correction's full-JID must match the original full-JID, and the recipient needs to ensure that the real bare JID of the sending occupant didn't change in between, e.g. by keeping track of leave/join presences.

[7]XEP-0301: In-Band Real Time Text <https://xmpp.org/extensions/xep-0301.html>.

[8]XEP-0258: Security Labels in XMPP <https://xmpp.org/extensions/xep-0258.html>.

[9]XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.

6 IANA Considerations

None.

7 XMPP Registrar Considerations

7.1 Protocol Namespaces

The XMPP Registrar [10] includes 'urn:xmpp:message-correct:0' in its registry of protocol namespaces (see <https://xmpp.org/registrar/namespaces.html>).

- urn:xmpp:message-correct:0

7.2 Protocol Versioning

If the protocol defined in this specification undergoes a revision that is not fully backwards-compatible with an older version, the XMPP Registrar shall increment the protocol version number found at the end of the XML namespaces defined herein, as described in Section 4 of XEP-0053.

8 XML Schema

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

<xs:schema
    xmlns:xs='http://www.w3.org/2001/XMLSchema'
    targetNamespace='urn:xmpp:message-correct:0'
    xmlns='urn:xmpp:message-correct:0'
    elementFormDefault='qualified'>

  <xs:element name='replace'>
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base='xs:string'>
          <xs:attribute name='id' type='xs:string' use='required'/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
```

---

[10]The XMPP Registrar maintains a list of reserved protocol namespaces as well as registries of parameters used in the context of XMPP extension protocols approved by the XMPP Standards Foundation. For further information, see <https://xmpp.org/registrar/>.

```
</xs:schema>
```

9 Acknowledgements