XEP-0201: Best Practices for Message Threads

Peter Saint-Andre
mailto:stpeter@stpeter.im
xmpp:stpeter@jabber.org
https://stpeter.im/

Ian Paterson
mailto:ian.paterson@clientside.co.uk
xmpp:ian@zoofy.com

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

2010-11-29
Version 1.0

Status	Type	Short Name
Active	Informational	N/A

This specification defines recommended handling of XMPP message threads.

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

Although message threads are re-used in XMPP extension protocols such as Chat State Notifications (XEP-0085) [1], best practices for generating and handling message threads have never been well specified (e.g., in RFC 3921 [2] or RFC 6121 [3]). This document attempts to clearly specify those matters for implementation by XMPP clients.

2 Motivation

Threads matter because they enable XMPP clients to:

- Track conversation topics within the context of a one-to-one chat session, a Multi-User Chat (XEP-0045) [4] room, or exchange of normal messages.

- Restart a conversation from message history.

- Enable clients to distinguish between different conversation threads when presenting a user's message, chat, and groupchat histories, thus providing a more coherent user interface (e.g. by collapsing threads to a single history entry).

- Separate logical sessions from physical interface objects such as windows.

- Route XMPP stanzas within a client (e.g., dispatching different content types to different windows), thus facilitating the creation of more robust plugin architectures.

3 Generation

3.1 Inclusion

Depending on the type of the message (i.e., the value of the 'type' attribute), the <thread/> should be included as follows:

Message Type	Inclusion
chat	RECOMMENDED
groupchat	RECOMMENDED
headline	OPTIONAL
normal	OPTIONAL

---

[1]XEP-0085: Chat State Notifications <https://xmpp.org/extensions/xep-0085.html>.

[2]RFC 3921: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <http://tools.ietf.org/html/rfc3921>.

[3]RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <http://tools.ietf.org/html/rfc6121>.

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

3.2 New Threads

Unless a <message/> stanza is written in direct reply to another <message/> stanza, if a ThreadID is included then its value SHOULD be newly generated when a human user initiates a chat conversation with another user (i.e., a <message/> stanza of type 'chat'), starts a new conversation in the context of a multi-user chat environment (i.e., a <message/> stanza of type 'groupchat'), or sends a normal message.

If the <message/> stanza is written in direct reply to another <message/> stanza, then the ThreadID SHOULD be the value from the the original <message/> stanza. (Determining what constitutes a <message/> stanza written in reply to another is a matter left to individual implementation, but it is envisaged that in most cases it would be the result of, e.g., the user clicking a 'reply' button when reading the contents of the previous stanza.)

3.3 Child Threads

In some situations, the conversation veers from the original topic. In this situation, it can be sensible to generate a new thread that is an offshoot or child of the original thread. The connection of the child thread to the parent thread is indicated by including the original ThreadID as the value of the 'parent' attribute.

Listing 1: Thread with parent

```xml
<message
    to='romeo@example.net/orchard'
    from='juliet@example.com/balcony'
    id='asiwe8289ljfdalk'
    type='chat'
    xml:lang='en'>
  <body>Art thou not Romeo, and a Montague?</body>
  <thread parent='7edac73ab41e45c4aafa7b2d7b749080'>
    e0ffe42b28561960c6b12b944a092794b9683a38
  </thread>
</message>
```

4 Handling

In general, the XMPP <thread/> element is handled in a manner similar to the "References:" header field from email (see RFC 5322 [5]) and netnews (see RFC 5536 [6]), as well as the THREAD

---

[5]RFC 5322: Internet Message Format <http://tools.ietf.org/html/rfc5322>.
[6]RFC 5536: Netnews Article Format <http://tools.ietf.org/html/rfc5536>.

extension to IMAP (see RFC 5256 [7]). Detailed guidelines for particular XMPP message types are provided in the following sections.

4.1 Chat Messages

For <message/> stanzas of type "chat" exchanged between two entities, the value of the <thread/> element shall be considered equivalent to a unique identifier for the chat session or conversation thread. If an entity receives such a message with a new or unknown ThreadID, it SHOULD treat the message as part of a new chat session. A client MAY destroy the thread when it goes offline, but SHOULD NOT destroy the thread if a human user merely disengages from the chat session (e.g., by closing a window in a client interface).

If an entity receives an XMPP presence stanza of type "unavailable" from the other entity during a chat session, it SHOULD NOT destroy the thread; instead, it SHOULD assume that the other entity will still be able to continue the session (perhaps the other entity was temporarily disconnected by a network error or is persisting the state of the session until it reconnects and receives "offline" messages).

If an entity receives a message of type "chat" without a thread ID, then it SHOULD create a new session with a new thread ID (and include that thread ID in all the messages it sends within the new session).

4.2 Groupchat Messages

For <message/> stanzas of type "groupchat" exchanged between multiple entities in a Multi-User Chat (XEP-0045) [8] room or similar environment, the value of the <thread/> element shall be considered equivalent to a unique identifier for a conversation thread in the multi-user environment.

When displaying a threaded groupchat conversation within a user interface, a client SHOULD provide a visual indication of the thread to which a message belongs. Methods for such indications include (non-exhaustively) the grouping together of all messages from the same thread, providing an index of threads, or formatting all messages within a thread in a cohesive manner, e.g. with uniform coloring.

4.3 Normal Messages

For <message/> stanzas of type "normal", the value of the <thread/> element shall be considered equivalent to a unique identifier for a conversation thread that proceeds outside the context of a "real-time" chat session or groupchat session.

When displaying threaded messages of type "normal" within a user interface, a client SHOULD

---

[7]RFC 5256: Internet Message Access Protocol - SORT and THREAD Extensions <http://tools.ietf.org/html/rfc5256>.

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

provide a visual indication of the thread to which a message belongs. Methods for such indications include (non-exhaustively) the grouping together of all messages from the same thread, providing an index of threads, or formatting all messages within a thread in a cohesive manner, e.g. with uniform coloring.

4.4 Headline Messages

There are no special handling requirements related to threads for <message/> stanzas of type "headline", because it is not expected that a client will allow the recipient to reply to such messages.

4.5 Messages That Have Been Archived

When displaying historical conversations within a user interface, a client SHOULD provide a visual indication of the thread to which a message belongs. Methods for such indications include (non-exhaustively) the grouping together of all messages from the same thread, providing an index of threads, or formatting all messages within a thread in a cohesive manner, e.g. with uniform coloring.

5 Security Considerations

Several security considerations related to XMPP threads are described in RFC 6121 [9].

6 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [10].

---

[9]RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <http://tools.ietf.org/html/rfc6121>.

[10]The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols, such as port numbers and URI schemes. For further information, see <http://www.iana.org/>.

7 XMPP Registrar Considerations

This document requires no interaction with the XMPP Registrar [11].

---

[11]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/>.