XEP-0228: Requirements for Shared Editing

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

2007-08-22
Version 0.1

Status	Type	Short Name
Deferred	Standards Track	N/A

This document defines requirements for the design of XMPP-based shared editing protocols.

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

Shared editing applications such as whiteboarding have been developed using XMPP, but there is no general editing and synchronization protocol that can be used by such applications. This document attempts to define the requirements for such a protocol so that work can proceed on protocol development.

2 Requirements

2.1 General

1. The underlying protocol shall be designed for synchronization of XML document instances between entities that communicate via XMPP.

2. All parties to a shared XML editing session must have the same representation of the XML document after all synchronization messages have been processed.

3. It should be possible to use the protocol for multiple application types; examples include but are not limited to whiteboarding using Scalable Vector Graphics (SVG) [1], editing of XHTML 1.0 [2] documents, and Collaborative Data Objects (XEP-0204) [3].

4. There is no requirement to synchronize non-XML data.

5. It must be possible to synchronize XML document instances either between two entities or among more than two entities.

6. The protocol should not (unduly) limit the number of parties to a shared XML editing session.

7. The protocol should be as lightweight as possible in order to minimize bandwidth consumption.

8. Certain applications may need to handle particular features of the XML syntax for that application type (e.g., layers in SVG or pages in XHTML); the protocol must not forbid such specialized semantics.

---

[1]Scalable Vector Graphics (SVG) 1.1 Specification <http://www.w3.org/TR/SVG11/>.
[2]XHTML 1.0 <http://www.w3.org/TR/xhtml1>.
[3]XEP-0204: Collaborative Data Objects <https://xmpp.org/extensions/xep-0204.html>.

2.2 Transport

1. It must be possible to use the protocol "directly" between two entities without assistance from intermediaries, either via an XMPP server (XMPP Core [4]) or in serverless mode (Link-Local Messaging (XEP-0174) [5]).

2. It must be possible to use the protocol through a Multi-User Chat (XEP-0045) [6] room.

3. It should be possible to use the protocol with a specialized synchronization component attached to an XMPP server.

2.3 Editing

1. It must be possible to add new "nodes" (elements and attributes) to the XML document.

2. It must be possible to edit existing nodes.

3. It must be possible to remove existing nodes.

4. It should be possible to move a node from one location to another within the XML document.

5. Where possible, all edits should be commutative in order to minimize the need for locking of the XML document.

6. It should be possible to send the complete content of a node or only the difference between the last version of the node and the current version.

2.4 Data Formats

1. It should be possible to re-use the XML document outside of a shared editing session (e.g., for offline editing in a different application, or for archiving).

2. It should be possible to specify the version of the protocol.

---

[4]RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.
[5]XEP-0174: Link-Local Messaging <https://xmpp.org/extensions/xep-0174.html>.
[6]XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.

3. It should be possible to refer to outside resources (e.g., images hosted at websites) from within the protocol.

4. It should be possible to validate synchronization messages against a defined schema for the application type in real time.

5. It should be possible to log all synchronization messages and associated metadata for archiving and logging purposes.

6. It must be possible to uniquely identify each node in an XML document, both within the scope of a standalone shared XML editing session and if the edited object is shared with other appliclations or embedded in other objects.

7. It must be possible to maintain the ordering of XML elements in the document.

8. It must be possible to specify the parent of any given node.

9. It should be possible to associate an XML document with other XML documents.

10. It must be possible to represent human-readable text in a language and character set appropriate for a human user.

11. It should be possible to associate relevant metadata with each node in the XML document, which might include time of creation, identity of the creator, time of last modification, and identity of last modifier.

12. It should be possible to mix text nodes with other child nodes (namely elements).

2.5 Sessions

1. It must be possible to initiate, update, join, and terminate a shared XML editing session.

2. It must be possible to specify the roles of parties to a shared XML editing session.

3. It should be possible to re-use the permission from a Multi-User Chat room, Publish-Subscribe node, or other service that has a permissions model.

4. It should be possible to declare shared editing of the XML document to be complete (e.g., in preparation for archiving of the XML document or editing by another application).

5. It should be possible to declare shared editing of part of the XML document to be complete.

6. It must be possible to reject out-of-order synchronization messages.

7. It should be possible to specify that all parties shall move their viewing context to a particular location in the XML data object.

2.6 States

1. It should be possible to retrieve the current state of the XML document from a party to the shared editing session.

2. It should be possible to retrieve the history of edits to the XML document from a party to the shared editing session.

3. It should be possible to retrieve all the changes to the XML document since a specific sequence number.

2.7 Discovery

1. It must be possible to discover whether another entity can engage in a shared XML editing session.

2. It must be possible to discover which application types another entity can handle.

3. It should be possible to query an entity for the shared XML editing sessions to which it is a party.

4. It should be possible to query an entity for the XML documents it owns or knows about (which may be the result of a past editing session).

3 IANA Considerations

This document requires no action by the Internet Assigned Numbers Authority (IANA) [7].

4 XMPP Registrar Considerations

This document requires no action by the XMPP Registrar [8].

5 Acknowledgements

For their previous work on XMPP-based shared editing technologies, the author wishes to thank Mats Bengtsson, Michael Bishop, Dave Bryson, Boyd Fletcher, Joonas Govenius, Marshall Huss, Huib-Jan Imbens, Jasen Jacobsen, Michael Krutsch, Keith Lirette, Ian Paterson, Chad Smith, and Dan Winkowski.

---

[7]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/>.

[8]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/>.