XEP-0061: Shared Notes

Jeremie Miller
mailto:jer@jabber.org
xmpp:jer@jabber.org

2003-09-30
Version 0.2

Status	Type	Short Name
Deferred	Informational	Not yet assigned

A simplistic mechanism for shared notes, modeled after common stickie note applications.

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 1

2 Message Extension 1

3 Implementation Notes 1

1 Introduction

A very simple namespace contaning display hints for the content in a message. Can be used for person-person collaboration, or by a service managing notes.

2 Message Extension

Normal messages are sent, with a sharednote namespace extending them hinting to any supporting client on how to display the message as a note instead. Any changes to the note within that client should then be sent back to the sender, either automatically or when the user saves the note (depending on the update element, by default on a save action by the user).

Listing 1: An Example Shared Note Message

```
<message from="jer@jabber.org/foo" to="stpeter@jabber.org/bar">
  <thread>1X5440</thread>
  <subject>Council Votes</subject>
  <body>Need votes from bob, tom, and jane yet for XEP-0000</body>
  <note xmlns="http://www.jabber.org/protocol/sharednote">
    <color>#001122</color>
    <bgcolor>#221100</bgcolor>
    <font>font-name</font>
    <x>%left</x>
    <y>%top</y>
    <z>#</z>
    <width>%</width>
    <height>%</height>
    <update>auto|user</update>
  </note>
</message>
```

Any element not specified in the note should use the last known setting or client defaults, so that when a change is sent, only the changed elements are returned.

3 Implementation Notes

Each thread is a different shared note. Auto updates should use an internal client timer and batch large changes into chunks, when the user is typing every 5-10 seconds or so. When the user has made changes that haven't been sent and an update comes in on the same thread the client should prompt the user with the changes offering to replace or save their changes.