XEP-0272: Multiparty Jingle (Muji)

Sjoerd Simons
mailto:sjoerd.simons@collabora.co.uk
xmpp:sjoerd.simons@collabora.co.uk

Dafydd Harries
mailto:dafydd.harries@collabora.co.uk
xmpp:dafydd.harries@collabora.co.uk

Marvin Wißfeld
mailto:xmpp@larma.de
xmpp:jabber@larma.de

2024-08-21
Version 0.2.0

Status	Type	Short Name
Experimental	Standards Track	muji

This specification defines an XMPP protocol extension for initiating and managing multiparty voice and video conferences within an XMPP MUC

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 How it Works 1

3 Joining a Conference 1

4 Leaving a Conference 3

5 Adding a Content Type 3

6 Removing a Content Type 4

7 Relays and Mixers 4

8 Call Invites 5

1 Introduction

Jingle (XEP-0166) [1] is used to negotiate peer to peer media sessions. Muji (short for Multiparty Jingle) is a way to coordinate Jingle sessions between a group of people. Muji conferences are held in Multi-User Chat (XEP-0045) [2] rooms.

2 How it Works

A Muji conference has a number of contents, each of which has unique name, content type, and an encoding. Each participant may provide a stream for each content, and communicates which contents they are willing to provide streams for, along with encoding information, in their MUC presence. This serves two purposes. Firstly, so that each participant knows which contents every other participant provides. Secondly, so that there is a global payload type (PT) mapping for the various contents, so that clients only need to encode and payload each content that they provide once.
Participants are not required to participate all the contents that are available. For example, a Muji client might choose to only request audio streams.

3 Joining a Conference

Joining a conference is done in two stages. The first step is to declare that preparations are being done to either join or start a Muji session inside the MUC. This is indicated by the client sending a presence stanza to the MUC with a preparing element in the Muji section.

```
<presence from='wiccarocks@shakespeare.lit/laptop'
  to='darkcave@chat.shakespeare.lit/oldhag'>
  <c xmlns="http://jabber.org/protocol/caps"
    node="http://telepathy.freedesktop.org/wiki/Muji"
    ver="48QdBuXRCJFb8qIzgy1FOHSGO0U="
    hash="sha-1" />
  <muji xmlns='urn:xmpp:jingle:muji:0'>
    <preparing />
  </muji>
</presence>
```

The client MUST then wait until the MUC rebroadcasts its presence message, after which it MUST wait for all other participants that had a preparing element in their presence to finish preparation. Afterwards it should finish its own preparation by updating its presence with the contents it wants to take part in.

[1] XEP-0166: Jingle <https://xmpp.org/extensions/xep-0166.html>.
[2] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.

```
<presence from='wiccarocks@shakespeare.lit/laptop'
  to='darkcave@chat.shakespeare.lit/oldhag'>
  <c xmlns="http://jabber.org/protocol/caps"
    node="http://telepathy.freedesktop.org/wiki/Muji"
    ver="48QdBuXRCJFb8qIzgy1FOHSGO0U="
    hash="sha-1" />
  <muji xmlns='urn:xmpp:jingle:muji:0'>
    <content creator='initiator' name='video'>
      <description xmlns='urn:xmpp:jingle:apps:rtp:0' media='video'>
        <payload-type id='97' name='theora' clockrate='90000'/>
      </description>
    </content>
    <content creator='initiator' name='voice'>
      <description xmlns='urn:xmpp:jingle:apps:rtp:0' media='audio'>
        <payload-type id='97' name='speex' clockrate='8000'/>
        <payload-type id='18' name='G729'/>
      </description>
    </content>
  </muji>
</presence>
```

When a client adds a payload ID to a content description, it MUST have the same codec name and receiving parameters as the corresponding entries in other participants' payload maps for that content. For instance, if Alice defines a payload type with ID 98, codec Speex and a clock rate of 8000 for a content called "voice0", then Bob must define payload type 98 identically or not at all for that content.

Furthermore, each content description MUST include at least one payload type that every other participant supports. In other words, the intersection of payload type mappings in descriptions for a content must not be the empty set. This avoids clients having to encode the same stream multiple times, which can be very costly, and also allows sending the encoded data only once where the transport makes this possible (e.g. IP multicast).

Once a client has constructed content descriptions and advertised them in its MUC presence, it MUST initiate a Jingle session with every other participant. The requirement that it is the joining participant that initiates sessions avoids race conditions.

Jingle sessions are initiated between the real JIDs of participants, since IQ routing is not guaranteed to work in Multi-User Chat (XEP-0045) [3] and in Mediated Information eXchange (MIX) (XEP-0369) [4]. To indicate that a Jingle session-initiate belongs to a MUJI conference, a <muji> element is placed in the <jingle> element. The receiver SHOULD only assume that a Jingle session-initiate belongs to a Muji conference if the sender is actually part of the referenced MUC. Content names inside Muji-related Jingle sessions always refer to the content with the same name inside the Muji conference.

[3] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.
[4] XEP-0369: Mediated Information eXchange (MIX) <https://xmpp.org/extensions/xep-0369.html>.

Listing 1: Jingle session-initiate referencing a Muji conference

```
<iq to='wiccarocks@shakespeare.lit/laptop' type='set' id='id1'>
  <jingle xmlns='urn:xmpp:jingle:1'
          action='session-initiate'
          initiator='hag66@shakespeare.lit/mobile'
          sid='sid1'>
    <muji room='darkcave@chat.shakespeare.lit' xmlns='
        urn:xmpp:jingle:muji:0' />
    <content creator='initiator' name='this-is-a-stub'>
      <description xmlns='urn:xmpp:jingle:apps:stub:0'/>
      <transport xmlns='urn:xmpp:jingle:transports:stub:0'/>
    </content>
  </jingle>
</iq>
```

4 Leaving a Conference

To leave a conference the Muji information MUST first be removed from the participant's presence; subsequently it SHOULD terminate all Jingle sessions related to that conference. Updating the presence first reduces the likelihood of situations where new participants initiate sessions with participants who are leaving the conference.

5 Adding a Content Type

Adding a stream follows a process similar to the joining of a conference. As a first step, an updated presence stanza MUST be sent which contains a preparing element as part of the Muji section.

```
<presence from='wiccarocks@shakespeare.lit/laptop'
      to='darkcave@chat.shakespeare.lit/oldhag'>
      <c xmlns="http://jabber.org/protocol/caps"
        node="http://telepathy.freedesktop.org/wiki/Muji"
        ver="48QdBuXRCJFb8qIzgy1FOHSGO0U="
        hash="sha-1" />
      <muji xmlns='urn:xmpp:jingle:muji:0'>
        <content creator='initiator' name='voice'>
          <description xmlns='urn:xmpp:jingle:apps:rtp:0' media='audio
              '>
            <payload-type id='97' name='speex' clockrate='8000'/>
            <payload-type id='18' name='G729'/>
          </description>
        </content>
        <preparing/>
      </muji>
```

```
    </presence>
```

The client MUST then wait until the MUC rebroadcasts its presence message, after which it MUST wait for all other participants that had a preparing element in their presence to finish their changes.
Afterwards the client should add the new content to the Muji section of its presence and add the content to all the Jingle sessions it had with participants it shared the content with.

```
<presence from='wiccarocks@shakespeare.lit/laptop'
      to='darkcave@chat.shakespeare.lit/oldhag'>
      <c xmlns="http://jabber.org/protocol/caps"
        node="http://telepathy.freedesktop.org/wiki/Muji"
        ver="48QdBuXRCJFb8qIzgy1FOHSGO0U="
        hash="sha-1" />
      <muji xmlns='urn:xmpp:jingle:muji:0'>
        <content creator='initiator' name='video'>
          <description xmlns='urn:xmpp:jingle:apps:rtp:0' media='video
              '>
            <payload-type id='97' name='theora' clockrate='90000'/>
          </description>
        </content>
        <content creator='initiator' name='voice'>
          <description xmlns='urn:xmpp:jingle:apps:rtp:0' media='audio
              '>
            <payload-type id='97' name='speex' clockrate='8000'/>
            <payload-type id='18' name='G729'/>
          </description>
        </content>
      </muji>
    </presence>
```

6 Removing a Content Type

To remove a content type, the participant SHOULD first send an updated presence without the content in its Muji section. Afterwards, it MUST remove the content from all the Jingle sessions it has open.

7 Relays and Mixers

When scaling to conferences with a big number of participants it's no longer viable for all participants to have direct connections. On connections where upstream bandwidth is the limiting factor, an RTP relay which is able to relay the stream to multiple participants on the behalf of the clients and which forwards the streams of other participants back to the client can be used. If the limiting factor is either CPU or downstream bandwidth then a mixer can be

used, which receives the media streams from other participants and mixes them on behalf of the client, so that the client only has to deal with receiving and decoding a single stream for each media type. On the sending side a mixer acts like a relay and relays the clients stream to all other participants. Both these services can either be provided by dedicated services or by other clients.

8 Call Invites

To invite someone to a Muji conference, a Call Invites (XEP-0482)[5] <invite> is sent containing a <muji> element in the urn:xmpp:jingle:muji:0 namespace with a 'room' attribute specifying the JID of the Muji MUC.

Listing 2: Invite to a MUJI conference

```
<message to='user@example.com' type='chat'>
  <invite xmlns='urn:xmpp:call-invites:0'>
    <muji xmlns='urn:xmpp:jingle:muji:0' room='room@conference.example
        .com' />
  </invite>
</message>>
```

[5] XEP-0482: Call Invites <https://xmpp.org/extensions/xep-0482.html>.