XEP-0403: Mediated Information eXchange (MIX): Presence Support.

Kevin Smith
mailto:kevin.smith@isode.com
xmpp:kevin.smith@isode.com

Steve Kille
mailto:steve.kille@isode.com
xmpp:steve.kille@isode.com

2020-12-01
Version 0.3.2

Status	Type	Short Name
Deferred	Standards Track	MIX-PRESENCE

This document defines an extension to Mediated Information eXchange (MIX) specified in XEP-0369 to provide presence information for MIX clients to MIX channel participants. It also specifies relay of IQ stanzas through a MIX channel.

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 Requirements **1**

3 Concepts **1**

3.1 Participant Information in Presence 1

3.2 Partipant JID Addressing for Presence and IQ relay 2

3.3 User Presence in MIX 2

3.4 MIX Channels in Roster 3

3.5 Presence Node 3

4 Use Cases **5**

4.1 Setting User Presence 5

4.2 Client Coming Online and Obtaining Presence from the Local Server 6

4.3 Going Offline 7

4.4 User Leaving a Channel 7

4.5 Relaying IQ Stanzas 8

4.6 Requesting a vCard through a Channel 8

5 Presence Initializion **10**

6 Internationalization Considerations **10**

7 Security Considerations **10**

8 IANA Considerations **10**

9 XMPP Registrar Considerations **10**

10 XML Schema **11**

11 Acknowledgements **11**

1 Introduction

The Mediated Information eXchange (MIX) protocol framework and core capabilities for message distribution are specified in Mediated Information eXchange (MIX) (XEP-0369) [1] (MIX-CORE). This specification enables presence status of online clients belonging to channel participants to be shared through the channel with participants that wish to see this presence status. This is a achieved by a MIX presence node, which channel participants may subscribe to.
MIX-CORE shares participant JIDs, which enables messages to be exchange directly. To facilitate communication between clients IQ stanzas MAY be routed through a MIX channel. This is needed because IQ stanzas from an unknown JID may be blocked. Routing IQ stanzas through the channel avoids this.

2 Requirements

This specification addressed a number of presence requirements:

1. The mechanism must work cleanly for participants with multiple clients.
2. Standard presence messages must be used to share presence.
3. Nick changes should be visible as changes (and not as a new user).
4. Where Mediated Information eXchange (MIX): JID Hidden Channels. (XEP-0404) [2] (MIX-ANON) is not used, a mechanism is needed so that participants are able to directly contact other participants.
5. IQ stanzas, including vCard, can be be sent indirectly through a MIX channel.

3 Concepts

3.1 Participant Information in Presence

A MIX channel implementing this specification MUST be able to distribute presence information about channel participants. In order to share JID and Nick information about a participant, this information is encoded in the presence message. This allows full presence information to be shared for each participant without the need for the client to perform any lookup.

[1] XEP-0369: Mediated Information eXchange (MIX) <https://xmpp.org/extensions/xep-0369.html>.
[2] XEP-0404: Mediated Information eXchange (MIX): JID Hidden Channels. <https://xmpp.org/extensions/xep-0404.html>.

3.2 Partipant JID Addressing for Presence and IQ relay

Participants in Presence messages and in IQ messages relayed through a channel are identified by an encoded JID of the form are encoded, using the format "<stable-participant-identifier>#<channel>@<mix domain>". This provides a stable JID for each participant. This JID format is used in three places:

1. The 'from' of presence stanzas generated by the channel, where the from encodes the Stable Participant ID of the participant to which the presence stanza relates.
2. The 'to' of IQ stanzas being addresses to a participant and being sent to the channel for indirect communication.
3. The 'from' of IQ stanzas coming from the channel, to reflect the sending participant where this stanza is being relayed by the channel.

These JIDs will be used to represent specific JID clients. The resource associated with the encoded JID can be either of the follipwing two options:

1. The resource value from the associated client JID; or
2. A mapped valued to an anonymized value. This approach MUST be used with MIX-ANON.

3.3 User Presence in MIX

MIX channels store presence of each online client for a user that chooses to publish presence, in the format that is distributed as presence. Presence is stored in the presence node with an item for each client that is publishing presence. Each item is named with an encoded JID and contains the presence information that is distributed. Where a user publishes presence for one or more clients, this information is available to all users subscribing to the channel presence.
A client participating in a MUC channel MAY send its presences status to the MIX channel using standard presence. The mechanisms to do this are summarized in the next section and standardized in Mediated Information eXchange (MIX): Participant Server Requirements (XEP-0405) [3].
The MIX channel will distribute received presence to participants that have subscribed to the participants node. The client to which each presence update refers is identified by the <from> of the presence sent by the MIX channel, using the encoded JID format.
Presence updates are distributed by a channel to the bare JID of each participant and then the subscriber's server will distribute to each of the participant's currently online clients following the rules set out in Mediated Information eXchange (MIX): Participant Server

[3] XEP-0405: Mediated Information eXchange (MIX): Participant Server Requirements <https://xmpp.org/extensions/xep-0405.html>.

Requirements (XEP-0405) [4].

3.4 MIX Channels in Roster

When a user joins a MIX channel, the channel MUST be added to the user's roster, as specified in Mediated Information eXchange (MIX): Participant Server Requirements (XEP-0405) [5]. There are two reasons for this.

1. It enables a client to determine which channels the user has joined and so may expect messages and/or presence updates from (dependent on what the user has subscribed to).
2. When the user has chosen to share presence with the channel, it enables this to happen using standard presence mechanisms. This avoids the need for MIX-specific mechanisms for clients to update presence on a channel. When a client comes online, presence information will be sent to each MIX channel that the user participates in. This will update other channel participants. It will also lead to a presence update for each MIX channel being sent to the client. So a user will receive channel presence information as the user comes online, in contrast to being subsequent to a MUC Join.

3.5 Presence Node

MIX defines one node to support presence, as summarized in the table below. This node is required to support this specification.

Name	Node	Description	Update	Distribution
Presence	'urn:xmpp:mix:nodes:presence'	For storing information about the availability status of online participants, which MAY include multiple clients for a single participant.	Presence	Presence

[4] XEP-0405: Mediated Information eXchange (MIX): Participant Server Requirements <https://xmpp.org/extensions/xep-0405.html>.

[5] XEP-0405: Mediated Information eXchange (MIX): Participant Server Requirements <https://xmpp.org/extensions/xep-0405.html>.

The presence node contains the presence value for clients belonging to participants that choose to publish presence to the channel. A MIX channel MAY require that all participants publish presence, so that active channel participants are visible. It is not possible to enforce this in the server, so participants in a channel with this option MUST publish presence. Each item in the presence node is identified by an encoded JID. The presence is encoded as a standard a presence stanza using a <presence/> element qualified by the 'jabber:client' namespace.
MIX extends the <presence> stanza using a <mix> element qualified by the 'urn:xmpp:mix:presence:0' namespace. This enables any receiver of presence to identify the JID of the client to which the presence presence refers and to have a nick to display. This element contains two child elements:

1. A <nick> element that contains the Nick of the message sender, taken from the Participants Node. This MUST be present if a Nick is defined for the user.
2. A <jid> element containing the full JID of the client. This MUST be present, unless following the "JID Hidden" model defined in MIX-ANON. If this element is omitted, the <nick> element MUST be present.

This node MAY be subscribed to by users using the user's bare JID. So presence of online clients is sent to the user's server for each user subscribed to this node. Presence is always sent using standard presence protocol and MUST NOT be sent using pubsub protocol. Clients MUST NOT directly access the presence node using pubsub. The Presence node is a permanent PubSub node. The following example shows a presence node value for the client with full JID 'hecate@shakespeare.example/UUID-x4r/2491'.

Listing 1: Value of Presence Node

```
<items node='urn:xmpp:mix:nodes:presence'>
  <item id='123456#coven@mix.shakespeare.example/UUID-x4r/2491'>
    <presence xmlns='jabber:client'>
      <mix xmlns='urn:xmpp:mix:presence:0'>
        <jid>hecate@shakespeare.example/UUID-x4r/2491</jid>
        <nick>thirdwitch</jid>
      </mix>
      <show>dnd</show>
      <status>Making a Brew</status>
    </presence>
  </item>
</items>
```

4 Use Cases

4.1 Setting User Presence

A user joins a channel over an extended period, and participation in a channel does not generally change when user goes online or offline. The user's participation in a channel is reflected by the user's bare JID in the participant node. All messages to the channel are sent to this JID.
A user MAY share presence information with the channel, for one or more online clients. The channel is entered by the user's server into the user's roster when the user subscribes to the channel as specified in Mediated Information eXchange (MIX): Participant Server Requirements (XEP-0405) [6]. Where a user shares presence information with a channel, the subscription is configured with one way presence, which will cause all presence changes for the client to be sent to the channel. When an XMPP client comes online or changes presence status, this will be communicated by the user to the user's server using broadcast presence. The user's XMPP server is then responsible to share this presence information to each entry in the roster and in particular to each MIX channel in the roster.
The MIX channel will update the "urn:xmpp:mix:nodes:presence" node with any change of status and the updated presence information and then share this updated presence with users subscribed to this node, as described below. When the user sets an explicit status, this is used to modify the presence node in the channel. When a client being used by the user goes offline, the associated server will send presence status "unavailable" to the MIX channel, which will cause the node item for that client to be removed from the presence node.
Presence status and availability is set in a MIX channel by standard presence stanzas sent to the MIX channel by the user's server. Users wishing to receive presence updates will subscribe to the MIX channel presence node. Presence updates are sent out to subscribing participants using standard presence stanzas.
A user setting status is now used as an example. Unlike in Multi-User Chat (XEP-0045) [7] where coming online is a special action, coming online in MIX is implicit when presence status is set. Going offline is a achieved by setting presence status to unavailable, which removes the client full JID entry from the presence node. When a user sets a presence status, the user's server sends updated presence to the MIX channel, and the MIX service then publishes the user's availability to the "urn:xmpp:mix:nodes:presence" node. If there is not an item named by the full JID of the client with updated presence status, this item is created. The sequence is shown in the following examples, starting with a client setting presences status on the connected server.

Listing 2: Client Sets Presence Status on Server

```
<presence xmlns='jabber:client'
                from='hag66@shakespeare.example/UUID-a1j/7533'>
  <show>dnd</show>
```

[6] XEP-0405: Mediated Information eXchange (MIX): Participant Server Requirements <https://xmpp.org/extensions/xep-0405.html>.
[7] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.

```
  <status>Making a Brew</status>
</presence>
```

The server then sends the presence information to roster entries. The following example then shows the presence message from the client's server to the MIX channel.

Listing 3: Server sends Presence Status to MIX Channel

```
<presence  from='hag66@shakespeare.example/UUID-a1j/7533'
           to='coven@mix.shakespeare.example'>
  <show>dnd</show>
  <status>Making a Brew</status>
</presence>
```

The user's presence information is then published by the service to the "urn:xmpp:mix:nodes:presence" node. The MIX channel then broadcasts the presence change to all users who are subscribed to the "urn:xmpp:mix:nodes:presence" node. The presence stanza 'from' uses an encoded JID. Note that presence is associated with a client and so will have a full JID. The following example shows a presence message as distributed by the server to a presences subscriber.

Listing 4: Channel Distributes Presence

```
<presence from='123435#coven@mix.shakespeare.example/UUID-a1j/7533'
          to='hag99@shakespeare.example'
          id='77E07BB0-55CF-4BD4-890E-3F7C0E686BBD'>
  <mix xmlns='urn:xmpp:mix:presence:0'>
     <jid>hecate@shakespeare.example/UUID-x4r/2491</jid>
     <nick>thirdwitch</jid>
  </mix>
  <show>dnd</show>
  <status>Making a Brew</status>
</presence>
```

The presence is distributed to those subscribing to the MIX channel presence node using a standard XMPP presence stanza. The presence change is recorded on the "urn:xmpp:mix:nodes:presence" node.
The history of the presence node MAY be archived using MAM. The MAM archive stores the node in PubSub format, following the node specification. This enables presence history to be retrieved using PubSub.

4.2 Client Coming Online and Obtaining Presence from the Local Server

MIX Clients obtain presence from their local server. This is specified in Mediated Information eXchange (MIX): Participant Server Requirements (XEP-0405) [8].

[8] XEP-0405: Mediated Information eXchange (MIX): Participant Server Requirements <https://xmpp.org/extensions/xep-0405.html>.

4.3 Going Offline

When a client goes offline, this presence update is sent by the client's server to the MIX channel. From the client perspective, this is the same as any other presence change. The MIX Channel also needs to remove the client from the participant's node.

Listing 5: Client Goes Offline in the Channel

```
<presence type='unavailable'
          from='hag66@shakespeare.example/UUID-a1j/7533'
          to='coven@mix.shakespeare.example'/>
```

The MIX channel will retract (remove) the item in the presence node of the MIX channel identified by the client's full JID. The MIX channel will notify subscribers to the presence node of the user going offline by sending a presence stanza to the full JID of each client. The presence stanza will use the encoded JID of the client that is going offline, as shown in the following example:

Listing 6: Channel Distributes Notification of Client going Offline

```
<presence from='12345#coven@mix.shakespeare.example/UUID-a1j/7533'
          to='hecate@shakespeare.example'
          id='77E07BB0-55CF-4BD4-890E-3F7C0E686BBD'
          type='unavailable'>
      <mix xmlns='urn:xmpp:mix:presence:0'>
        <jid>hecate@shakespeare.example/UUID-x4r/2491</jid>
        <nick>thirdwitch</jid>
      </mix>
</presence>
```

There is the possibility that the message associated with the user going offline will be lost. If this happens, "ghost" entries will appear in the presence node. A MIX service MAY take steps to address this, for example by probing client with a disco for presence items that remain unchanged for a long period.

4.4 User Leaving a Channel

The primary actions for a user leaving a channel are specified in Mediated Information eXchange (MIX) (XEP-0369) [9]. This section sets out additional actions for handling presence. When a user leaves the channel, all entries for the user's clients MUST be removed from the participants node. The MIX channel MUST distribute unavailable presence notifications for each client removed to all subscribers of the participants node.

[9] XEP-0369: Mediated Information eXchange (MIX) <https://xmpp.org/extensions/xep-0369.html>.

Listing 7: Channel Distributes Notification when User Leaves Channel

```
<presence from='12345#coven@mix.shakespeare.example/UUID-a1j/7533'
          to='hecate@shakespeare.example'
          id='77E07BB0-55CF-4BD4-890E-3F7C0E686BBD'
          type='unavailable'>
      <mix xmlns='urn:xmpp:mix:presence:0'>
        <jid>hecate@shakespeare.example/UUID-x4r/2491</jid>
        <nick>thirdwitch</jid>
      </mix>
</presence>
```

4.5 Relaying IQ Stanzas

MIX channels MAY relay IQ stanzas between participants. This is often useful to obtain client information where a direct request to the client would be blocked. When a client sends an IQ stanza through a MIX channel, it will set the 'from' to its own JID and set the 'to' to the encoded JID of the recipient. Participants may be addressed by full JID or bare JID. The MIX channel will modify the JIDs in the outgoing message, so that the 'to' is the full JID of the recipient and the 'from' is the encoded JID of the sender. This is illustrated in the vCard section below.

4.6 Requesting a vCard through a Channel

A client MAY request the vCard of a channel participant through a MIX channel, for example to get an avatar. The MIX channel MAY pass this request on or MAY block it. vCard requests MAY use vcard-temp (XEP-0054) [10] (vcard-temp) or vCard4 over XMPP (XEP-0292) [11] (vCard4 over XMPP). The MIX channel does not process the vCard requests, but simply relays them on to real bare JID of the target. A MIX service MAY choose to relay one or both protocols. In the following example, using vcard-temp, the requesting client sends a message to the encoded JID of the channel participant for which the vCard is desired.

Listing 8: Client directly requests vCard through channel

```
<iq from='hag66@shakespeare.example/UUID-c8y/1573'
    id='lx09df27'
    to='989898#coven@mix.shakespeare.example'
    type='get'>
  <vCard xmlns='vcard-temp'/>
</iq>
```

The MIX channel MAY pass on the vCard request or MAY reject with an error, dependent on channel policy. The MIX service will then address the vCard request to the user's server

[10] XEP-0054: vcard-temp <https://xmpp.org/extensions/xep-0054.html>.
[11] XEP-0292: vCard4 over XMPP <https://xmpp.org/extensions/xep-0292.html>.

(using bare JID) using an the encoded JID of the requester in the 'from'.

Listing 9: Channel passes on vCard request to the User's Server

```
<iq from='123456#coven@mix.shakespeare.example/6789'
    id='lx09df27'
    to='peter@shakespeare.example'
    type='get'>
  <vCard xmlns='vcard-temp'/>
</iq>
```

The user's server, on behalf of the user, MUST send a response or reject with an error. The user's server will send the vCard back to the channel.

Listing 10: User's Server sends vCard Response via MIX channel

```
<iq from='peter@shakespeare.example'
    id='lx09df27'
    to='123456#coven@mix.shakespeare.example/6789'
    type='result'>
    <vCard xmlns='vcard-temp'>
    <FN>Peter Saint-Andre</FN>
    <N>
      <FAMILY>Saint-Andre</FAMILY>
      <GIVEN>Peter</GIVEN>
      <MIDDLE/>
    </N>
    <NICKNAME>stpeter</NICKNAME>
    <URL>http://www.xmpp.org/xsf/people/stpeter.shtml</URL>
  </vCard>
  <query xmlns='http://jabber.org/protocol/disco#info'>
</iq>
```

The MIX channel will then send the vCard response to the requesting client on behalf of the client sending the response.

Listing 11: MIX Channel sends vCard responst to Client

```
<iq from='989898#coven@mix.shakespeare.example'
    id='lx09df27'
    to='hag66@shakespeare.example/UUID-c8y/1573'
    type='result'>
 <vCard xmlns='vcard-temp'>
    <FN>Peter Saint-Andre</FN>
    <N>
      <FAMILY>Saint-Andre</FAMILY>
      <GIVEN>Peter</GIVEN>
      <MIDDLE/>
    </N>
```

```
    <NICKNAME>stpeter</NICKNAME>
    <URL>http://www.xmpp.org/xsf/people/stpeter.shtml</URL>
  </vCard>
</iq>
```

5 Presence Initializion

For an active MIX Channel, the presence node is updated as channel participants change status and presence information is sent to the channel. When a MIX channel starts, typically when the associated MIX Service and Server start, there is a need to initialize the presence node. This is done by the XMPP server associated with the MIX channel sending out a presence probe for each channel participant, following the presence probe process specified in RFC 6121 [12]. A presence probe MUST NOT be sent for users who have set presence preference to not sharing.

6 Internationalization Considerations

See considerations in Mediated Information eXchange (MIX) (XEP-0369) [13].

7 Security Considerations

See considerations in Mediated Information eXchange (MIX) (XEP-0369) [14].
When converting a 1:1 conversation to a channel there is potential to expose sensitive information and to present invalid information.

8 IANA Considerations

None.

9 XMPP Registrar Considerations

The urn:xmpp:mix namespace needs to be registered.

[12] RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <http://tools.ietf.org/html/rfc6121>.
[13] XEP-0369: Mediated Information eXchange (MIX) <https://xmpp.org/extensions/xep-0369.html>.
[14] XEP-0369: Mediated Information eXchange (MIX) <https://xmpp.org/extensions/xep-0369.html>.

10 XML Schema

To be supplied when MIX progresses to proposed standard.

11 Acknowledgements

See Mediated Information eXchange (MIX) (XEP-0369) [15] for a list of contributors to the MIX Family of specifications.

[15] XEP-0369: Mediated Information eXchange (MIX) <https://xmpp.org/extensions/xep-0369.html>.