
TigaseDoc

Release 0.1

Tigase, Inc.

Aug 24, 2022

TIGASE MIX COMPONENT

| | | |
|----------|--|----------|
| 1 | Overview | 3 |
| 1.1 | What is MIX? | 3 |
| 1.2 | How does it work? | 4 |
| 1.3 | Benefits for mobile-first applications relying on push | 4 |
| 1.4 | Is MUC obsolete? | 5 |
| 1.5 | Tigase MIX Release Notes | 5 |
| 2 | Configuration | 7 |
| 2.1 | Setting ACL | 7 |
| 2.2 | Disabling support for MUC | 8 |
| 2.3 | Setting limit of cached channels | 8 |

Welcome to Tigase Mediated Information eXchange (MIX) component guide. The MIX component allows you to have multi user group chats (channels), which are better suited for multi device usage.

OVERVIEW

Tigase MIX component is a component extending Tigase PubSub Component and providing support for [XEP-0369: MIX](#) protocol extensions being part of [MIX specification family](#).

Additionally, it provides basic support for [MUC protocol](#) to provide support and interoperability with older software not supporting MIX,

It is configured by default to run under the name `mix`. Installations of Tigase XMPP Server ($\geq 8.2.0$) run this component enabled by default under the same name even if not explicitly enabled/configured.

1.1 What is MIX?

MIX stands for Mediated Information eXchange (MIX) and it's basics are defined in [XEP-0369: Mediated Information eXchange \(MIX\)](#):

“an XMPP protocol extension for the exchange of information among multiple users through a mediating service. The protocol can be used to provide human group communication and communication between non-human entities using channels, although with greater flexibility and extensibility than existing groupchat technologies such as Multi-User Chat (MUC). MIX uses Publish-Subscribe to provide flexible access and publication, and uses Message Archive Management (MAM) to provide storage and archiving.”

Specification outlines several [requirements](#) of which those seems to be the most interesting:

- “A user’s participation in a channel persists and is not modified by the user’s client going online and offline.”
- “Multiple devices associated with the same account can share the same nick in the channel, with well-defined rules making each client individually addressable.”
- “A reconnecting client can quickly resync with respect to messages and presence.”

MIX itself serves as an umbrella for set of MIX-related XMPP extensions that specify the exact protocol. Two of them are required for the implementation to be considered as MIX compliant:

- MIX-CORE defined in [XEP-0369: Mediated Information eXchange \(MIX\)](#) - “sets out requirements addressed by MIX and general MIX concepts and framework. It defines joining channels and associated participant management. It defines discovery and sharing of MIX channels and information about them. It defines use of MIX to share messages with channel participants.”
- MIX-PAM defined in [XEP-0405: Mediated Information eXchange \(MIX\): Participant Server Requirements](#) - “defines how a server supporting MIX clients behaves, to support servers implementing MIX-CORE and MIX-PRESENCE.”

In addition to the above extensions, there are several other that are optional:

- MIX-PRESENCE defined in [XEP-0403: Mediated Information eXchange \(MIX\): Presence Support](#) - adds the ability for MIX online clients to share presence, so that this can be seen by other MIX clients. It also specifies relay of IQ stanzas through a channel. **(Not supported fully)**
- MIX-ADMIN defined in [XEP-0406: Mediated Information eXchange \(MIX\): MIX Administration](#) - specifies MIX configuration and administration of MIX.
- MIX-ANON defined in [XEP-0404: Mediated Information eXchange \(MIX\): JID Hidden Channels](#) - specifies a mechanism to hide real JIDs from MIX clients and related privacy controls. It also specifies private messages. **(Not supported fully)**
- MIX-MISC defined in [XEP-0407: Mediated Information eXchange \(MIX\): Miscellaneous Capabilities](#) - specifies a number of small MIX capabilities which are useful but do not need to be a part of MIX-CORE: handling avatars, registration of nickname, retracting of a message, sharing information about channel and inviting people, converting simple chat to a channel. **(Not supported fully)**
- MIX-MUC defined in [XEP-0408: Mediated Information eXchange \(MIX\): Co-existence with MUC](#) - defines how MIX and MUC can be used tog

1.2 How does it work?

The most stark difference to MUC is that MIX requires support from both server that hosts the channel and user's server. This is done to facilitate the notion that the user (and not particular connection or client application) joined the group and allows for greater flexibility in terms of message delivery (which can be send to one or many connections, or even generates notification over PUSH). Another important difference is the flexibility to choose which notifications from the channel user wants to receive (that can be messages, presence, participators or node information). In the most basic approach, when user decides to join a channel, it sends an IQ stanza to it's own local server indicating address of the desired channel and list of MIX nodes to which it wants to subscribe. User's server then forward's subscription request to the destination, MIX server. As a result user receives subscription confirmation and from this point onwards will receive notifications from the channel, independently of it's current network connection. Another essential bit of MIX is the reliance on [XEP-0313: Message Archive Management](#) to control message history and the complementary interaction between MIX server and user's server. Main channel history is handled by the MIX server, but user's that joined the channel will retrieve and synchronise message history querying their local server, which will maintain complete history of the channels that user has joined (based on the received notifications). This also means that even if the channel is removed, user is still able to access it's history through local MAM archive (limited to time when user was member of the channel). As a result, chatter between client, client's server and mix server is also reduced and unnecessary traffic is eliminated.

1.3 Benefits for mobile-first applications relying on push

All of this helps especially with clients that relay on constrained environment - be that unreliable network connection or operating system that limits time that application can be running. Because there is no dependency on the dynamic state of user presence/connection the issue with occupant leaving and (re)joining the room is eliminated - user gets the notification always. What's more, thanks to shared responsibilities between MIX and user's server, and the latter getting all notifications from MIX channel, it's possible to generate notifications without relying on workarounds (that most of the time are not reliable or impact resource usage).

In case of Tigase XMPP server it gets better thanks to our experimental [filtering groupchat notifications](#) feature, which allows user control when to receive PUSH notifications from group chats (always, only when mentioned or never)

1.4 Is MUC obsolete?

We think that MIX is the way forward, but we also know that this won't happen overnight. Because of that MUC is still supported in all our applications and Tigase XMPP Server implements XEP-0408: [Mediated Information eXchange \(MIX\): Co-existence with MUC](#) to allow all non-MIX client to participate in MIX channel discussions using MUC protocol.

1.5 Tigase MIX Release Notes

Welcome to Tigase MIX 1.0.0! This is a feature release for with a number of fixes and updates.

1.5.1 Tigase MIX 1.0.0 Release Notes

Major Changes

This is the introductory version of MIX specification family

All Changes

- #mix-2:Implement XEP-0369: Mediated Information eXchange (MIX)
- #mix-3:Implement XEP-0406: Mediated Information eXchange (MIX): MIX Administration
- #mix-6:Create tests for MIX CORE & Participants Server Requirements
- #mix-8:Improve caching
- #mix-9:Add support for MIX-MUC integration
- #mix-10:Invalid response for disco#items
- #mix-14:Add configuration to limit who can create channels in component
- #mix-15:NPE in MAMItemHandler
- #mix-16:Add MIX to installer as option.
- #mix-17:Could not parse new configuration of channel: PubSubException: Only participants and information nodes are supported!
- #mix-18:NPE when sending requests to removed channel nodes
- #mix-19:MAM:2 is not advertised
- #mix-20:MIX component is broadcasting messages with bare JID
- #mix-21:Possibility of duplicated subscription of a node
- #mix-22:Nickname not returned in response after being set
- #mix-23:Remove banned participants from participants list and subscriptions
- #mix-24:NPE in MIXProcessor
- #mix-25:Create MIX component documentation and publish it
- #mix-26:Allow installation admins to manager MIX channels if domain admins are allowed
- #mix-27:MIX-MUC message duplication

- [#mix-28](#):NPE in `Affiliations.getSubscriberAffiliation`
- [#mix-29](#):Weird “open channel” behaviour

CONFIGURATION

Configuration of MIX component is extended version of PubSub component configuration. We will not describe here configuration of PubSub component as it already available in PubSub component documentation.

2.1 Setting ACL

With ACL you can control who can create publicly visible channels and also ad-hoc channels. ACL properties accept following values:

ALL

Anyone can create channel

LOCAL

Only local users can create channels (from all local domains on all local domains)

ADMIN

Only installation administrator can create channels

DOMAIN_OWNER

Only domain owner of the domain as the domain under which MIX component is running can create channels

DOMAIN_ADMIN

Only domain administrator of the domain as the domain under which MIX component is running can create channels

DOMAIN

Only users from the same domain as the domain under which MIX component is running can create channels

2.1.1 Setting ACL for creation of public channels

Property name: ``publicChannelCreationAcl``

Default value: ``DOMAIN_ADMIN``

By default we allow only local domain owners or admins to create publicly browsable channels.

Allowing domain users to create public channels.

```
mix () {
  logic () {
    publicChannelCreationAcl = 'DOMAIN'
  }
}
```

2.1.2 Setting ACL for creation of ad-hoc (private) channels

Property name: ``adHocChannelCreationAcl``

Default value: ``DOMAIN``

Allowing all local users to create public channels.

```
mix () {
  logic () {
    adHocChannelCreationAcl = 'LOCAL'
  }
}
```

2.2 Disabling support for MUC

MIX component by default exposes MUC compatibility layer for clients that doesn't support MIX yet, so they would still be able to participate in the MIX channel conversation. It's possible to disable it with the following option.

Disabling support for MUC.

```
mix () {
  roomPresenceModule (active: false) {}
}
```

2.3 Setting limit of cached channels

Property name: ``maxCacheSize``

Default value: ``2000``

MIX component is caching channels configuration and affiliation in memory while it is processing request for the particular channel. To make that more efficient it is using cache to keep the most often used channels configuration in memory instead of loading it every time.

You can increase this value by setting `maxCacheSize` property in the `config` scope of the MIX component:

Setting limit of cached channels.

```
mix () {
  config () {
    maxCacheSize = 3000
  }
}
```