# XEP-0463: MUC Affiliations Versioning


Maxime Buquet
mailto:pep@bouah.net
xmpp:pep@bouah.net

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



2022-03-08
Version 0.1.0

| Status | Type | Short Name |
| --- | --- | --- |
| Experimental | Standards Track | NOT_YET_ASSIGNED |



This specification provides a way to reduce the amount of queries necessary to stay up-to-date with affiliations in a MUC room.

# Legal

## Copyright



## Permissions



## Warranty



## Liability



## Conformance

# Contents

# 1 Introduction

Affiliations are a way in Multi-User Chat (XEP-0045) [1] to handle permissions (ownership, membership, etc.). Currently, an observer gets the data in presence of online participants at the time of join, of new participants when they join, or when affiliation is changed. This observer has to send in one query per category (owner, admin, member, outcast) if they want to get a full view.
Caching this data can be tricky, as any member removed when the observer is offline is not likely to be caught until the 4 queries are run again. Having multiple queries can also produce race conditions where an occupant is moved out of a group (e.g., owners) to another group (e.g., members) while an observer queries, and they end up not seeing this occupant at all, or twice, or not in the correct group.
Affiliations have become more and more used lately, for example in so-called private groupchats, that is, a non-public, non-anonymous (visible JIDs), members-only room. Having to run all queries is particularly annoying in these private rooms where some implementations prefer to display all participants instead of online participants only, and where the affiliations list is used to know whom to encrypt to in an e2ee context.
This specification sets a versioning mechanism in place, allowing an observer in a room to get the latest changes from a known version, and to reduce the amount of round-trips to handle affiliations.

# 2 Requirements

- Reduce the number of round-trips necessary to get the list of affiliations.
- Allow an implementation to request a diff of affiliations since a known version.

# 3 Glossary

**Full response** A response including the complete affiliation list.

**Boostrap request** A request asking for a full reponse.

# 4 Discovering Support

A server implementing this specification MUST advertise the urn:xmpp:muc:affiliations:0 as a Service Discovery (XEP-0030) [2] feature on the Multi-User Chat (XEP-0045) [3] room JID itself.

[1] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.
[2] XEP-0030: Service Discovery <https://xmpp.org/extensions/xep-0030.html>.
[3] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.

# 5 Usage Flow

## 5.1 Client request

When sending a join presence to a Multi-User Chat (XEP-0045) [4] room, a client may include a since attribute in the urn:xmpp:muc:affiliations:0 namespace on the <x xmlns='http://jabber.org/protocol/muc'> element. This attribute, a unique and opaque string, indicates the last affiliation version sent by the server that the client has seen, and cached. Sending an empty since attribute is a called bootstrap request, which asks the server for a full response.
This since attribute MUST NOT be broadcasted by the server to other participants. A room not stripping the attribute may disclose information about an occupant to other participants, even though this information is not intended for them.

Listing 1: Bootstrap request to get the full list.

```
<presence to='news@chat.commons.example' from='louise@commons.example/
    desktop'>
  <x xmlns='http://jabber.org/protocol/muc'
     xmlns:mav='urn:xmpp:muc:affiliations:0'
     mav:since='{}' />
</presence>
```

Listing 2: Request since specific version.

```
<presence to='news@chat.commons.example' from='rosa@commons.example/
    toaster'>
  <x xmlns='http://jabber.org/protocol/muc'
     xmlns:mav='urn:xmpp:muc:affiliations:0'
     mav:since='9pacabr2q1' />
</presence>
```

## 5.2 Server Response

### 5.2.1 On Join

Returning the list of affiliation changes is done as a <message/> stanza inside the <x xmlns='http://jabber.org/protocol/muc#user'> element to which since and/or until attributes of namespace urn:xmpp:muc:affiliations:0 MAY be added. Each affiliation is to be sent as an <item> element in the same namespace as its x parent, and MUST at least contain a jid attribute and an affiliation attribute, similarly to what is specified in MUC Affiliations.
The since attribute is used to reflect the version sent by the client and is the starting point of the diff that will be computed. The until attribute is the latest version the server has. When these attributes are present they MUST contain a valid version string (unique and opaque).
The reponse MUST be sent during the join process before any <presence/> is sent to the

[4] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.

joining user.
If a client sends a version that the server doesn't know, (e.g., because it only stores the last N versions, or the client made a mistake), the response MUST be a full response, with the since attribute not present, and the until attribute filled with the latest version.
If both attributes have the same value, meaning a client already has the latest version, the x element MUST be empty, only containing the two filled attributes.
There can be no empty diff but there can be empty full reponses (no affiliations).

Listing 3: Room full response for a version that it didn't know.

```
<message from='news@chat.commons.example'
         to='louise@commons.example/desktop'>
  <x xmlns='http://jabber.org/protocol/muc#user'
     xmlns:mav='urn:xmpp:muc:affiliations:0'
     mav:until='ruz41312vw'>
    <item jid='louise@commons.example' affiliation='owner' />
    <item jid='peter@commons.example' affiliation='owner' />
    <item jid='rosa@commons.example' affiliation='owner' />
  </x>
</message>
```

Listing 4: Room versioned response.

```
<message from='news@chat.commons.example'
         to='louise@commons.example/desktop'>
  <x xmlns='http://jabber.org/protocol/muc#user'
     xmlns:mav='urn:xmpp:muc:affiliations:0'
     mav:since='9pacabr2q1'
     mav:until='ruz41312vw'>
    <item jid='vladimir@commons.example' affiliation='none' />
    <item jid='rosa@commons.example' affiliation='owner' />
  </x>
</message>
```

Listing 5: Room full response with no affiliation

```
<message from='news@chat.commons.example'
         to='louise@commons.example/desktop'>
  <x xmlns='http://jabber.org/protocol/muc#user'
     xmlns:mav='urn:xmpp:muc:affiliations:0'
     mav:until='ruz41312vw'/>
</message>
```

### 5.2.2 Affiliation Changes

A new unique (to the room) version string MUST be generated for every affiliation change and included in the broadcast of this change.

When broadcasting an affiliation change (as a <presence/> or <message/>), on the <x xmlns='http://jabber.org/protocol/muc#user'>, a MUC room MUST add the since attribute in the urn:xmpp:muc:affiliations:0 namespace, containing the original version string (before the affiliation change), and the new version string (after the affiliation change) in the until attribute (of the same namespace).
Affiliation changes broadcasted to room occupants as <message/> defined as a MAY in Multi-User Chat (XEP-0045) [5], for example in Granting Owner Status, Granting Admin Status, or Revoking Admin Status, MUST be supported by the MUC room when implementing this specification.

Listing 6: Room broadcasts an affiliation change for a user not in the room.

```
<message from='news@chat.commons.example'
         to='louise@commons.example/desktop'>
  <x xmlns='http://jabber.org/protocol/muc#user'
     xmlns:mav='urn:xmpp:muc:affiliations:0'
     mav:since='9pacabr2q1'
     mav:until='ruz41312vw'>
    <item jid='peter@commons.example/mobile' affiliation='none' role='
       none'/>
  </x>
</message>
```

### 5.2.3 Error Cases

Permissions are handled as with affiliation iq queries in Multi-User Chat (XEP-0045) [6], the same rules should be applied by the server when deciding who can use this protocol.
Some notes regarding permissions can be find in Multi-User Chat (XEP-0045) [7], for example in Affiliations, or Modifying the member list.
An auth forbidden error MUST be returned in a <message/> stanza if an observer doesn't have the necessary permissions to request affiliations.

Listing 7: This observer isn't allowed to query affiliations in the room

```
<message from='news@chat.commons.example'
         to='vladimir@commons.example/web'
         id='foo' type='error'>
  <error by='news@chat.commons.example' type='auth'>
    <forbidden xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
  </error>
</message>
```

[5] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.
[6] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.
[7] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.

## 6 Security Considerations

An observer has to join the MUC room to use the protocol as the since element must be included in the join presence. This makes this specification less prone to vulnerabilities that may have happened with protocols such as MAM in the past (i.e., scraping information without being joined).
A server should be careful not to disclose past affiliation states. If an observer requests a version of which they weren't a part of, a server MUST return an error as specified in Error Cases.
When caching server responses, a client should make sure to associate the received version string to the room JID and not have a global cache for affiliation versions to prevent any cache poisoning issues.

## 7 IANA Considerations

This document requires no interaction with the the Internet Assigned Numbers Authority (IANA) [8].

## 8 XMPP Registrar Considerations

The XMPP Registrar includes the urn:xmpp:muc:affiliations:0 namespace in its registry of protocol namespaces at `https://xmpp.org/registrar/namespaces.html`.

## 9 Design Considerations

This specification is based on the design of roster versioning, with the difference of since and until attributes, to ensure clients stay in sync because unlike roster versioning this may be running over c2s, **and** s2s which may break.
The <x xmlns='http://jabber.org/protocol/muc[#user]/> elements are not defined extensible in the Multi-User Chat (XEP-0045) [9] specification, but as this is a negociated change there shouldn't be any issue.
**TODO? RSM.** Pagination may have been useful for large rooms, when receiving a full response. Versioning will reduce the amount of bandwidth used for further queries though.

[8] 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/>.
[9] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.

## 10 XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>

<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='urn:xmpp:muc:affiliations:0'
  xmlns='urn:xmpp:muc:affiliations:0'
  elementFormDefault='qualified'>

  <xs:annotations>
    <xs:documentation>
      The protocol documented by this schema is defined
      in XEP-xxxx: https://xmpp.org/extensions/xep-xxxx.html.
    </xs:documentation>
  </xs:annotations>

  <!-- TODO: How do I schematize standalone attributes? -->
  </xs:element>
</xs:schema>
```

## 11 Acknowledgements

Thanks to Emmanuel Gil Peyrot and Matthew Wild for the valuable feedback.