



XMPP

XEP-0386: Bind 2.0

Kevin Smith

<mailto:kevin.smith@isode.com>

<xmpp:kevin.smith@isode.com>

2019-01-31

Version 0.3.0

Status	Type	Short Name
Deferred	Standards Track	bind2

This specification provides a single-request replacement for several activities an XMPP client needs to do at startup.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2020 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	Use Cases	1
3.1	Discovering support	1
3.2	Performing the bind	2
3.3	Pipelining	3
4	Security Considerations	3
5	IANA Considerations	3
6	XMPP Registrar Considerations	4

1 Introduction

As XMPP has grown more feature-rich over time, more steps have been introduced that clients are likely to perform at startup, e.g. resource binding, archive synchronisation, enabling Carbons. Some of these introduce race conditions - e.g. if a client synchronises the archive before enabling Carbons, it can miss stanzas sent between these events, or if it enables Carbons before synchronising the archive it can receive duplicate messages. It may also cause duplicate messages by combining archive synchronisation and receipt of offline messages, or by receipt of messages addressed to the full JID between resource binding and archive synchronisation. This document provides a mechanism for atomically performing these operations to avoid these race conditions. It also provides information to a client that is generally useful about the state of the archive.

There have been other suggestions of further enhancements to the stream startup process in XEP-0388 (SASL2), and it is expected that in the future some protocol here will be reframed in terms of these facilities, but the core premise of the XEP (that these features are needed, and need enabling together) remains. It may be desirable to make the enabling of features extensible, such that the client can request which features are needed, together, but the current approach should serve as a suitable basis for discussion.

The returning of unread message state to the client relies on the archive having these data, which is a topic for another specification.

2 Requirements

- Allow MAM synchronisation and Carbons enabling without a race condition leading to missed or duplicated stanzas
- Avoid offline messages being received by clients using the MAM archive for message retrieval
- Allow clients to know which messages are unread in the archive, providing a cross-client alternative to offline messages
- Avoid clients receiving 'live' messages until they have synchronised their state from the server (to avoid double-receipt of messages, one from the archive one live)

3 Use Cases

3.1 Discovering support

A client does not advertise support for bind 2.0. If a server supports bind 2.0, it **MUST** advertise this in the stream features with a feature named 'bind' in the namespace 'urn:xmpp:bind2:0'.

Listing 1: Server advertises support in stream features

```
<stream:features>
  <bind xmlns='urn:xmpp:bind2:0' />
  <bind xmlns='urn:ietf:params:xml:ns:xmpp-bind' />
  <sm xmlns='urn:xmpp:sm:3' />
</stream:features>
```

3.2 Performing the bind

After authentication, a client performs a bind 2.0 by sending an element 'bind' in the same-space 'urn:xmpp:bind2:0'. (Note: this gets rid of manual resource binding altogether. For discussion on standards@)

Listing 2: Client initiates a bind 2.0

```
<bind xmlns='urn:xmpp:bind2:0' />
```

When it receives a bind 2.0 on an authenticated not-yet-bound session, the server MUST:

- Clear the offline messages for this user, if any, without sending them (as they will be provided by MAM).
- Perform resource binding to a random resource (see 6120)
- Work out which contacts have unread messages in the user's MAM archive, how many, and what the id of the last read message is
- Get the id of the newest stanza in the user's MAM archive
- Silently enable carbons for this session

After processing the bind stanza, as above, the server MUST respond with an element of type 'bound' in the namespace 'urn:xmpp:bind2:0', as in the below example

Listing 3: Server responds after processing the bind

```
<bound xmlns='urn:xmpp:bind2:0'>
  <jid>nurse@shakespeare.lit/abtet29a9fu</jid>
  <latest-id>balisoseb</latest-id>
  <unreads>
    <unread jid='romeo@shakespeare.lit' read-id='onuheuhoeu', unread-
      count='3' />
    <unread jid='juliet@shakespeare.lit' read-id='koeunhth29', unread-
      count='43' />
  </unreads>
</bound>
```

Servers SHOULD support the <unreads> feature, but if they do not then the <unreads> element MUST be elided, to distinguish between an empty element (no unread messages) and a missing feature.

3.3 Pipelining

A server supporting this specification MUST allow the following initial commands to be pipelined:

- Initial client stream header
- SASL Authentication*
- Post-authentication stream header
- Bind 2.0

[* pipelining will clearly not work for multi-stage SASL mechanisms, but the initial stage MUST be capable of being pipelined after the stream header, and the post-authentication stream header MUST be able to be pipelined after the final authentication element]

In this way, a client is able to (if using 'xmpps' for avoiding starttls, to be defined elsewhere), on second and subsequent login cache the presence of bind 2.0 in stream features from a previous session, and pipeline the entire stream initialisation process by sending the stream header, followed by auth, followed by a stream header, followed by a bind 2.0 in a single chunk.

Note: also enable acks? discuss on standards@

4 Security Considerations

This specification is mostly a reworking of existing protocol/features. The additional facility provided here is to provide information on the user's archive; as this facility is provided post-authentication and is only providing the user's data to the user, it doesn't introduce new security considerations.

5 IANA Considerations

None.

6 XMPP Registrar Considerations

The urn:xmpp:bind2:0 namespace must be registered..