



# XMPP

## XEP-0193: Proposed Resource Binding Improvements

Peter Saint-Andre  
<mailto:peter@andyet.net>  
<xmpp:stpeter@stpeter.im>  
<https://stpeter.im/>

2012-02-08  
Version 1.2

Status	Type	Short Name
Obsolete	Standards Track	N/A

This document proposes improvements to the definition of resource binding for inclusion in the specification that supersedes RFC 3920.

# Legal

## Copyright

This XMPP Extension Protocol is copyright © 1999 – 2017 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

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Unbinding a Resource</b>	<b>1</b>
<b>3</b>	<b>Business Rules</b>	<b>2</b>
3.1	From Addresses . . . . .	2
3.2	Server Sessions . . . . .	2
<b>4</b>	<b>Examples</b>	<b>2</b>
<b>5</b>	<b>Security Considerations</b>	<b>5</b>
<b>6</b>	<b>IANA Considerations</b>	<b>6</b>
<b>7</b>	<b>XMPP Registrar Considerations</b>	<b>6</b>
<b>8</b>	<b>XML Schema</b>	<b>6</b>

## 1 Introduction

RFC 3920 introduced the concept of binding a resource to an XML stream (this concept superseded part of the obsolete jabber:iq:auth protocol described in [Non-SASL Authentication \(XEP-0078\)](#)<sup>1</sup>). As defined in RFC 3920, resource binding enables a client to bind one resource to a stream but does not enable a client to unbind a resource and leaves underspecified what a server and client should do if a client binds more than one resource to a stream. Because the ability to bind multiple resources to a stream is desirable in certain environments (e.g., for devices that are unable to open more than one TCP connection or when a machine runs a daemon process that is used by multiple applications), this document proposes improvements to resource binding in order to address these shortcomings.

Note: The recommendations from this document were NOT incorporated into [RFC 6120](#)<sup>2</sup> and this document is Obsolete.

## 2 Unbinding a Resource

In order to properly manage the resources associated with an XML stream, a client must be able to unbind resources. This shall be completed by sending an IQ-set with a child element of <unbind/> qualified by the 'urn:ietf:params:xml:ns:xmpp-bind' namespace, which in turn has a child element of <resource/> whose XML character data specifies the resource to be unbound:

```
<iq from='someid@domain.tld/someresource' type='set' id='bind_2'>
  <unbind xmlns='urn:ietf:params:xml:ns:xmpp-bind'>
    <resource>someresource</resource>
  </unbind>
</iq>
```

If the server does not understand the <unbind/> element, it MUST return an error of <bad-request/>. Otherwise, if there is no such resource, the server MUST return an error of <item-not-found/>. When the client unbinds the only resource associated with the stream, the server SHOULD close the stream and terminate the TCP connection. The 'from' address included in bind and unbind stanzas SHOULD be the full JID <localpart@domain.tld/resource> associated with the resource in question; however, the server MUST NOT accept further stanzas from that full JID after successfully processing the unbind request.

A server SHOULD advertise its support for the 'urn:ietf:params:xml:ns:xmpp-bind' namespace and the unbind functionality by returning an appropriate stream feature as shown below:

```
<stream:features>
  <bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'>
    <required/>
  </bind>
</stream:features>
```

<sup>1</sup>XEP-0078: Non-SASL Authentication <<https://xmpp.org/extensions/xep-0078.html>>.

<sup>2</sup>RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <<http://tools.ietf.org/html/rfc6120>>.

```
</bind>
<unbind xmlns='urn:ietf:params:xml:ns:xmpp-bind' />
</stream:features>
```

## 3 Business Rules

### 3.1 From Addresses

When a client binds multiple resources to the same stream, proper management of 'from' addresses is imperative. In particular, a client **MUST** specify a 'from' address on every stanza it sends over a stream to which it has bound multiple resources, where the 'from' address is the full JID <localpart@domain.tld/resource> associated with the relevant resource. If a client does not specify a 'from' address on a stanza it sends over a stream to which it has bound multiple resources (or if it specifies as the 'from' address a full JID other than one of the bound resources), the server **MUST** return the stanza to the client with an <unknown-sender/> stanza error.<sup>3</sup>

Naturally, the existing rules from RFC 3920 regarding validation of asserted 'from' addresses still apply.

### 3.2 Server Sessions

In instant messaging and presence applications, an XMPP server manages a session on behalf of a connected client. A server **MUST** create and manage a separate session for each distinct resource, even if there are multiple resources associated with a given XML stream. In particular:

1. A server **MUST** consider each resource to be a distinct source of presence information, both with regard to presence notifications and with regard to presence probes.
2. A server **MUST** manage rosters (see RFC 3920) and [Privacy Lists \(XEP-0016\)](#)<sup>4</sup> separately for each resource.

## 4 Examples

The following examples show a possible flow of resource binding and unbinding (stanzas prefixed by "C:" are sent by the client, stanzas prefixed by "S:" are sent by the server). First, the client binds an initial resource to the stream.

---

<sup>3</sup>Currently there is no <unknown-sender/> stanza defined in RFC 3920. The author will work to add such an error condition (with a type of "modify") to the document that revises RFC 3920.

<sup>4</sup>XEP-0016: Privacy Lists <<https://xmpp.org/extensions/xep-0016.html>>.

Listing 1: Binding an Initial Resource

```

C: <iq from='juliet@capulet.com/core' type='set' id='bind-1'>
  <bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'>
    <resource>core</resource>
  </bind>
</iq>

S: <iq to='juliet@capulet.com/core' type='result' id='bind-1'>
  <bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'>
    <jid>juliet@capulet.com/core</jid>
  </bind>
</iq>

```

Now the client sends some stanzas, making sure to set its 'from' address:

Listing 2: Sending Some Stanzas

```

C: <iq from='juliet@capulet.com/core' type='get' id='roster-1'>
  <query xmlns='jabber:iq:roster' />
</iq>

S: <iq to='juliet@capulet.com/core' type='result' id='roster-1'>
  <query xmlns='jabber:iq:roster'>
    <item jid='romeo@montague.net' />
  </query>
</iq>

C: <presence from='juliet@capulet.com/core' />

```

Now the client binds a second resource to the stream.

Listing 3: Binding a Second Resource

```

C: <iq from='juliet@capulet.com/balcony' type='set' id='bind-2'>
  <bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'>
    <resource>balcony</resource>
  </bind>
</iq>

S: <iq to='juliet@capulet.com/balcony' type='result' id='bind-2'>
  <bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'>
    <jid>juliet@capulet.com/balcony</jid>
  </bind>
</iq>

```

If the server does not allow entities to bind multiple resources to the stream, it MUST return a <not-allowed/> error as described in RFC 3920.

Now the client sends more stanzas.

Listing 4: Sending More Stanzas

```

C: <iq from='juliet@capulet.com/balcony' type='get' id='roster-2'>
  <query xmlns='jabber:iq:roster' />
</iq>

C: <iq to='juliet@capulet.com/balcony' type='result' id='roster-2'>
  <query xmlns='jabber:iq:roster'>
    <item jid='romeo@montague.net' />
  </query>
</iq>

C: <presence from='juliet@capulet.com/balcony' />

C: <message to='romeo@montague.net'>
  <body>Wherefore art thou?</body>
</message>

S: <message type='error'>
  <body>Wherefore art thou?</body>
  <error type='modify'>
    <unknown-sender xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</message>

```

In handling the last stanza shown above, the server returns an `<unknown-sender/>` error to the sender because the sender did not include a 'from' address. Now the client binds a third resource to the stream.

Listing 5: Binding a Third Resource

```

C: <iq from='juliet@capulet.com/softphone' type='set' id='bind-3'>
  <bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'>
    <resource>softphone</resource>
  </bind>
</iq>

S: <iq to='juliet@capulet.com/softphone' type='result' id='bind-3'>
  <bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'>
    <jid>juliet@capulet.com/softphone</jid>
  </bind>
</iq>

```

Now the client unbinds its initial resource.

Listing 6: Unbinding a Resource

```

C: <iq from='juliet@capulet.com/core' type='set' id='unbind-1'>
  <unbind xmlns='urn:ietf:params:xml:ns:xmpp-bind'>
    <resource>core</resource>
  </unbind>
</iq>

```

```
    </unbind>
  </iq>
S: <iq from='juliet@capulet.com/core' type='result' id='unbind-1'/>
```

Now the client unbinds another resource.

Listing 7: Unbinding a Resource

```
C: <iq from='juliet@capulet.com/softphone' type='set' id='unbind-2'>
  <unbind xmlns='urn:ietf:params:xml:ns:xmpp-bind'>
    <resource>softphone</resource>
  </unbind>
</iq>
S: <iq to='juliet@capulet.com/softphone' type='result' id='unbind-2'/>
```

Now the client unbinds its last remaining resource.

Listing 8: Unbinding the Final Resource

```
C: <iq from='juliet@capulet.com/balcony' type='set' id='unbind-3'>
  <unbind xmlns='urn:ietf:params:xml:ns:xmpp-bind'>
    <resource>balcony</resource>
  </unbind>
</iq>
S: <iq to='juliet@capulet.com/balcony' type='result' id='unbind-3'/>
```

The server now SHOULD close the stream and terminate the underlying TCP connection.

Listing 9: Server Closes the Stream

```
S: </stream:stream>
```

## 5 Security Considerations

If properly implemented, the modifications described herein do not introduce any new security concerns above and beyond those defined in RFC 3920. However, care must be taken to properly manage 'from' addresses in order to avoid the delivery of stanzas from an unintended resource (which may, for example, leak presence information).



## 6 IANA Considerations

No interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)<sup>5</sup> is required as a result of this document.

## 7 XMPP Registrar Considerations

No interaction with the [XMPP Registrar](#)<sup>6</sup> is required as a result of this document, since the 'urn:ietf:params:xml:ns:xmpp-bind' namespace is already registered (see <https://xmpp.org/registrar/namespaces.html>).

## 8 XML Schema

Note: The following provisional schema is intended to replace the existing schema for the Resource Binding stream feature.

```
<?xml version='1.0' encoding='UTF-8'?>
<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='urn:ietf:params:xml:ns:xmpp-bind'
  xmlns='urn:ietf:params:xml:ns:xmpp-bind'
  elementFormDefault='qualified'>
  <xs:element name='bind'>
    <xs:complexType>
      <xs:sequence>
        <xs:choice minOccurs='0' maxOccurs='1'>
          <xs:element name='resource' type='resourceType' />
          <xs:element name='jid' type='fullJIDType' />
        </xs:choice>
        <xs:element name='required'
          minOccurs='0'
          maxOccurs='1'
          type='empty' />
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

<sup>5</sup>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/>.

<sup>6</sup>The XMPP Registrar maintains a list of reserved protocol namespaces as well as registries of parameters used in the context of XMPP extension protocols approved by the XMPP Standards Foundation. For further information, see <https://xmpp.org/registrar/>.

```
</xs:complexType>
</xs:element>

<xs:element name='unbind'>
  <xs:complexType>
    <xs:sequence minOccurs='0'>
      <xs:element name='resource' type='resourceType' />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:simpleType name='resourceType'>
  <xs:restriction base='xs:string'>
    <xs:minLength value='1' />
    <xs:maxLength value='1023' />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name='fullJIDType'>
  <xs:restriction base='xs:string'>
    <xs:minLength value='8' />
    <xs:maxLength value='3071' />
  </xs:restriction>
</xs:simpleType>

</xs:schema>
```