



# XMPP

## XEP-0267: Server Buddies

Artur Hefczyc

<mailto:artur.hefczyc@gmail.com>

<xmpp:artur.hefczyc@tigase.org>

Florian Jensen

<mailto:admin@flosoft.biz>

<xmpp:admin@im.flosoft.biz>

Mickaël Rémond

<mailto:mickael.remond@process-one.net>

<xmpp:mremond@process-one.net>

Peter Saint-Andre

<mailto:stpeter@jabber.org>

<xmpp:stpeter@jabber.org>

<https://stpeter.im/>

Matthew Wild

<mailto:mwild1@gmail.com>

<xmpp:mwild1@jaim.at>

2011-12-12

Version 0.4

Status	Type	Short Name
Experimental	Standards Track	NOT_YET_ASSIGNED

This specification defines a convention for presence subscriptions between XMPP servers.

# Legal

## Copyright

This XMPP Extension Protocol is copyright © 1999 - 2011 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 <http://xmpp.org/about-xmpp/xsf/xsf-ipr-policy/> or obtained by writing to XMPP Standards Foundation, 1899 Wynkoop Street, Suite 600, Denver, CO 80202 USA).

## Contents

1	Description	1
2	Ad-Hoc Command	2
3	Determining Support	4
4	Security Considerations	4
5	IANA Considerations	5
6	XMPP Registrar Considerations	5
6.1	Service Discovery Features . . . . .	5
6.2	Field Standardization . . . . .	5
7	Acknowledgements	5

## 1 Description

In XMPP, rosters and presence subscriptions have been used to date only among IM users (see [XMPP IM](#)<sup>1</sup>). However, nothing prevents the application of these concepts to other XMPP entities, such as components and servers. Given that a presence subscription typically indicates some level of trust in a peer, server deployments can use the sharing of XMPP presence information as a way to indicate that a given server has a trust relationship with a peer server (informally, we can say that the two servers consider each other "buddies"). The server might then share certain kinds of additional information only with its trusted peers, for example [Incident Reporting](#)<sup>2</sup>.

To establish such a trust relationship with a peer, a server sends a presence subscription request to the peer, just as is done between XMPP users.

Listing 1: Service sends subscription request to peer

```
<presence from='montague.lit'
  to='capulet.lit'
  type='subscribe' />
```

A server **MUST NOT** send such a presence subscription request unless explicitly requested to do so by a server administrator (see below).

Upon receiving such a presence subscription request, the XMPP server software running at the peer shall either prompt the server administrator to approve the request or (if explicitly configured to accept subscriptions requests) automatically approve it. (A future version of this specification might define an approval method based on [Data Forms](#)<sup>3</sup>.)

If the request is approved, the peer server then informs the originating server that the request has been approved.

Listing 2: Peer sends approval to server

```
<presence from='capulet.lit'
  to='montague.lit'
  type='subscribed' />
```

The peer **SHOULD** also send a subscription request to the originating server.

Listing 3: Peer sends subscription request to server

```
<presence from='capulet.lit'
  to='montague.lit'
  type='subscribe' />
```

---

<sup>1</sup>RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <<http://tools.ietf.org/html/rfc6121>>.

<sup>2</sup>XEP-0268: Incident Reporting <<http://xmpp.org/extensions/xep-0268.html>>.

<sup>3</sup>XEP-0004: Data Forms <<http://xmpp.org/extensions/xep-0004.html>>.

The originating server would then approve that subscription request.

Listing 4: Services sends approval to peer

```
<presence from='montague.lit'  
  to='capulet.lit'  
  type='subscribed' />
```

If an XMPP server implementation supports this usage of presence subscriptions, it **MUST** keep a list of approved entities, which we denote a "server roster". The implementation **MAY** use that roster for access control purposes defined in other specifications.

## 2 Ad-Hoc Command

This section defines an [Ad-Hoc Commands](#)<sup>4</sup> node scoped by the [Field Standardization for Data Forms](#)<sup>5</sup> FORM\_TYPE specified in [Service Administration](#)<sup>6</sup>. Upon advancement of this specification to Draft, this section ought to be moved to XEP-0133.

The command node for this use case **SHOULD** be "http://jabber.org/protocol/admin#server-buddy".

A sample protocol flow for this use case is shown below.

Listing 5: Admin Subscribes Server to Peer Server

```
<iq from='bard@shakespeare.lit/globe'  
  id='server-buddy-1'  
  to='shakespeare.lit'  
  type='set'  
  xml:lang='en'>  
  <command xmlns='http://jabber.org/protocol/commands'  
    action='execute'  
    node='http://jabber.org/protocol/admin#server-buddy' />  
</iq>
```

Unless an error occurs (see the "Error Handling" section of XEP-0133), the service **SHOULD** return the appropriate form.

Listing 6: Service Returns Server Buddy Form to Admin

```
<iq from='shakespeare.lit'  
  id='server-buddy-1'  
  to='bard@shakespeare.lit/globe'  
  type='result'
```

---

<sup>4</sup>XEP-0050: Ad-Hoc Commands <<http://xmpp.org/extensions/xep-0050.html>>.

<sup>5</sup>XEP-0068: Field Data Standardization for Data Forms <<http://xmpp.org/extensions/xep-0068.html>>.

<sup>6</sup>XEP-0133: Service Administration <<http://xmpp.org/extensions/xep-0133.html>>.

```

    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
          node='http://jabber.org/protocol/admin#server-buddy'
          sessionId='server-buddy:20040408T0337Z'
          status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Subscribing to a Peer Server</title>
      <instructions>Fill out this form to subscribe your server to a
        peer server.</instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The_domain_name_of_the_peer_server'
            type='jid-single'
            var='peerjid'>
        <required/>
      </field>
    </x>
  </command>
</iq>

```

Note: In virtual hosting environments, the server can determine the the domain name from which to send the presence subscription based on the 'to' address of the <iq/> stanza.

Listing 7: Admin Submits Server Buddy Form to Service

```

<iq from='bard@shakespeare.lit/globe'
    id='server-buddy-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
          node='http://jabber.org/protocol/admin#server-buddy'
          sessionId='server-buddy:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='serverjid'>
        <value>shakespeare.lit</value>
      </field>
      <field var='peerjid'>
        <value>marlowe.lit</value>
      </field>
    </x>
  </command>
</iq>

```

Listing 8: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
  id='server-buddy-2'
  to='bard@shakespeare.lit/globe'
  type='result'
  xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
    node='http://jabber.org/protocol/admin#server-buddy'
    sessionid='server-buddy:20040408T0337Z'
    status='completed' />
</iq>
```

Notification of completion MAY include the processed data in a data form of type "result".

### 3 Determining Support

To advertise its support for the server buddy feature, when replying to service discovery information ("disco#info") requests a server MUST return a URNs "urn:xmpp:server-presence".

Listing 9: Service discovery information request

```
<iq from='jabber.org'
  id='uw72g176'
  to='xmpp.net'
  type='get'>
  <query xmlns='http://jabber.org/protocol/disco#info' />
</iq>
```

Listing 10: Service discovery information response

```
<iq from='xmpp.net'
  id='uw72g176'
  to='jabber.org'
  type='result'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    <feature var='urn:xmpp:server-presence' />
  </query>
</iq>
```

## 4 Security Considerations

Because server "buddies" might be granted greater privileges than unknown peers, care should be taken in sending or approving presence subscription requests. In particular, a server MUST NOT send a presence subscription request unless explicitly requested to do so by

a server administrator.

## 5 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)<sup>7</sup>.

## 6 XMPP Registrar Considerations

### 6.1 Service Discovery Features

```
<var>
  <name>urn:xmpp:server-presence</name>
  <desc>Support for subscriptions to server presence</desc>
  <doc>XEP-0267</doc>
</var>
```

### 6.2 Field Standardization

[Field Standardization for Data Forms](#)<sup>8</sup> defines a process for standardizing the fields used within Data Forms scoped by a particular namespace. This registration adds two more reserved fields to the 'http://jabber.org/protocol/admin' namespace defined in XEP-0133.

```
<form_type>
  <name>http://jabber.org/protocol/admin</name>
  <field var='peerjid'
        type='jid-single'
        label='The_Jabber_ID_of_a_peer_server' />
</form_type>
```

## 7 Acknowledgements

Thanks to Kim Alvefur, Waqas Hussain, and Tobias Markmann for their feedback.

---

<sup>7</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>8</sup>XEP-0068: Field Data Standardization for Data Forms <http://xmpp.org/extensions/xep-0068.html>.