



# XMPP

## XEP-0253: PubSub Chaining

Ralph Meijer  
<mailto:ralphm@ik.nu>  
<xmpp:ralphm@ik.nu>

Peter Saint-Andre  
<mailto:stpeter@jabber.org>  
<xmpp:stpeter@jabber.org>  
<https://stpeter.im/>

2009-11-18  
Version 0.2

Status	Type	Short Name
Deferred	Standards Track	NOT_YET_ASSIGNED

This specification defines a method for chaining pubsub nodes together, resulting in lightweight repeaters for pubsub notifications.

# 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	Introduction	1
2	How It Works	1
3	Notifications	3
4	Determining Support	4
5	Security Considerations	4
6	IANA Considerations	4
7	XMPP Registrar Considerations	5
	7.1 Protocol Namespaces . . . . .	5
	7.2 Field Standardization . . . . .	5
8	Acknowledgements	5

## 1 Introduction

To enable lightweight repeaters for [Publish-Subscribe](#)<sup>1</sup> notifications, we need the ability to subscribe one pubsub node to another pubsub node. This specification defines a method for doing so, using [Ad-Hoc Commands](#)<sup>2</sup>.

## 2 How It Works

The owner of a pubsub node can subscribe that "local" node to a "remote" node using the flow described below. In these examples, the node owner is admin@consumer.tld, the local node is weatherbot@consumer.tld/Chicagoland, and the remote node has a NodeID of "OHR" at the pubsub service notify.weather.tld.

Listing 1: Owner requests chaining

```
<iq from='admin@consumer.tld/client'
  id='chaining-1'
  to='weatherbot@consumer.tld'
  type='set'
  xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
    action='execute'
    node='http://jabber.org/protocol/pubsub#chaining' />
</iq>
```

Unless an error occurs, the service SHOULD return the appropriate form.

Listing 2: Service returns chaining form to node owner

```
<iq from='weatherbot@consumer.tld'
  id='chaining-1'
  to='admin@consumer.tld/client'
  type='result'
  xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
    node='http://jabber.org/protocol/pubsub#chaining'
    sessionId='a73sjjvkl37jfea'
    status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Chaining Request</title>
      <instructions>Fill out this form to complete your request.</
        instructions>
      <field type='hidden' var='FORM_TYPE'>
```

---

<sup>1</sup>XEP-0060: Publish-Subscribe <<http://xmpp.org/extensions/xep-0060.html>>.

<sup>2</sup>XEP-0050: Ad-Hoc Commands <<http://xmpp.org/extensions/xep-0050.html>>.

```

    <value>http://jabber.org/protocol/pubsub#chaining</value>
  </field>
  <field label='The Node ID of the local pubsub node'
        type='text-single'
        var='local-node'>
    <required/>
  </field>
  <field label='The Jabber ID of the remote pubsub service'
        type='jid-single'
        var='remote-service'>
    <required/>
  </field>
  <field label='The NodeID of the remote pubsub node'
        type='text-single'
        var='remote-node'>
    <required/>
  </field>
</x>
</command>
</iq>

```

Listing 3: Admin submits chaining form to service

```

<iq from='admin@consumer.tld/client'
  id='chaining-2'
  to='weatherbot@consumer.tld'
  type='submit'
  xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
    node='http://jabber.org/protocol/pubsub#chaining'
    sessionid='a73sjjvkl37jfea'
    status='executing'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/pubsub#chaining</value>
      </field>
      <field var='local-node'>
        <value>Chicagoland</value>
      </field>
      <field var='remote-service'>
        <value>notify.weather.tld</value>
      </field>
      <field var='remote-node'>
        <value>OHR</value>
      </field>
    </x>
  </command>
</iq>

```

Listing 4: Service informs admin of completion

```
<iq from='weatherbot@consumer.tld'
  id='chaining-2'
  to='admin@consumer.tld/client'
  type='result'
  xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
    node='http://jabber.org/protocol/pubsub#chaining'
    sessionId='a73sjjvkl37jfea'
    status='completed' />
</iq>
```

At this point, the service itself will subscribe to the remote node.

Listing 5: Service subscribes to remote node for chaining

```
<iq type='set'
  from='weatherbot@consumer.tld/Chicagoland'
  to='notify.weather.tld'
  id='sub1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <subscribe
      node='OHR'
      jid='weatherbot@consumer.tld/Chicagoland' />
  </pubsub>
</iq>
```

Now subscribers who are local to the consumer.tld XMPP service can subscribe directly to weatherbot@consumer.tld/Chicagoland instead of the remote pubsub node at notify.weather.tld. We therefore refer to weatherbot@consumer.tld/Chicagoland as a "chaining node" and the remote node as a "chained node".

### 3 Notifications

When a chaining node delivers a notification to its subscribers, it SHOULD include an [Extended Stanza Addressing](#)<sup>3</sup> header of "ofrom" to specify the chained node or service that generated the notification (note: this header is not yet defined in XEP-0033).

Listing 6: Chaining node notifies subscribers

```
<message from='weatherbot@consumer.tld/bot'
  to='subscriber@consumer.tld'
  id='foo'>
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
```

<sup>3</sup>XEP-0033: Extended Stanza Addressing <<http://xmpp.org/extensions/xep-0033.html>>.

```
<items node='Chicagoland'>
  <item id='ae890ac52d0df67ed7cfd51b644e901'>
    <example xmlns='urn:xmpp:example'>message</example>
  </item>
</items>
</event>
<addresses xmlns='http://jabber.org/protocol/address'>
  <address type='ofrom' jid='notify.weather.tld' />
</addresses>
</message>
```

## 4 Determining Support

If a pubsub service supports Ad-Hoc Commands, it MUST advertise the commands it supports via [Service Discovery](#)<sup>4</sup> (as described in XEP-0050: Ad-Hoc Commands); such commands exist as well-defined discovery nodes associated with the service. In particular, if a pubsub service supports chaining it MUST advertise a command of "http://jabber.org/protocol/pubsub#chaining".

## 5 Security Considerations

The ability to subscribe one node to another node introduces the possibility of exposing non-public information in a public way, since the access controls for the node that originates a notification might not be known or enforced by the downstream node. Therefore, the upstream node (or its owner) is advised to make a careful access decision before allowing a downstream node (or any other entity) to subscribe.

Note: The upstream node can discover the identity of the downstream node by sending a service discovery information ("disco#info") request to the downstream node, and MAY cancel or decline the downstream node's subscription if it determines that the node has an identity of "pubsub/leaf" or "pubsub/collection".

## 6 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)<sup>5</sup>.

<sup>4</sup>XEP-0030: Service Discovery <<http://xmpp.org/extensions/xep-0030.html>>.

<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/>>.

## 7 XMPP Registrar Considerations

The [XMPP Registrar](#)<sup>6</sup> shall include the following information in its registries.

### 7.1 Protocol Namespaces

The XMPP Registrar shall include 'http://jabber.org/protocol/pubsub#chaining' in its registry of protocol namespaces (see <<http://xmpp.org/registrar/namespaces.html>>).

### 7.2 Field Standardization

[Field Standardization for Data Forms](#)<sup>7</sup> defines a process for standardizing the fields used within [Data Forms](#)<sup>8</sup> scoped by a particular namespace (see <<http://xmpp.org/registrar/ormtypes.html>>). The reserved fields for the 'http://jabber.org/protocol/pubsub#chaining' namespace are specified below.

```
<form_type>
  <name>http://jabber.org/protocol/pubsub#chaining</name>
  <doc>XEP-0253</doc>
  <desc>Forms used for chaining of pubsub nodes.</desc>
  <field var='local-node'
        type='text-single'
        label='The_Node_ID_of_the_local_node' />
  <field var='remote-node'
        type='text-single'
        label='The_NodeID_of_the_remote_node' />
  <field var='remote-service'
        type='jid-single'
        label='The_Jabber_ID_of_the_remote_pubsub_service' />
</form_type>
```

## 8 Acknowledgements

Thanks to Joe Hildebrand for his feedback.

---

<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 <<http://xmpp.org/registrar/>>.

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

<sup>8</sup>XEP-0004: Data Forms <<http://xmpp.org/extensions/xep-0004.html>>.