



XMPP

XEP-0497: Pubsub Extended Subscriptions

Jérôme Poisson

<mailto:goffi@goffi.org>

<xmpp:goffi@jabber.fr>

2024-11-20

Version 0.1.0

Status	Type	Short Name
Experimental	Standards Track	pubsub-ext-sub

This specification extends the XMPP PubSub protocol by introducing mechanisms for users to subscribe to an entire node hierarchy or to receive notifications on node metadata updates.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2024 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	Subscription Options	1
3.1	Subscription Type	1
3.2	Subscription Depth	2
4	Subscription Request and Notification	2
4.1	Request	2
4.1.1	Success Case	3
4.2	Notification Types	3
4.2.1	Items Notification	4
4.2.2	Metadata Notification	4
4.2.3	Linked Items Notification	5
4.2.4	Depth Subscription	6
5	Business Rules	6
6	Discovering Support	6
7	Security Considerations	7
8	IANA Considerations	8
9	XMPP Registrar Considerations	8
10	Acknowledgements	8

1 Introduction

One of the key mechanisms in [Publish-Subscribe \(XEP-0060\)](#)¹ is its subscription functionality, which allows users to be notified of changes in node content. This specification extends this mechanism by allowing subscriptions to an entire hierarchy using "parent" relationships as defined in XEP-XXXX: Pubsub Node Relationships. Subscribers will receive notifications when items are published or retracted, nodes are added or removed from a branch, or nodes and their descendants are deleted. Additionally, the specification introduces an option for subscribing to metadata updates, enabling users to be notified of changes in node relationships, access models, or any other kind of metadata.

2 Requirements

The design goals of this XEP are:

- be backward compatible: nothing must be broken for existing implementations;
- use existing mechanisms whenever possible;
- allow clients to get node metadata updates;
- allow clients to subscribe to node hierarchy when XEP-XXXX Pubsub Node Relationships' parent relationship is used;
- allow clients to subscribe to linked nodes when XEP-XXXX Pubsub Node Relationships' link relationship is used.

3 Subscription Options

This specification uses subscription options as defined in [Publish-Subscribe \(XEP-0060\)](#)². Those options can be configured by following [§6.3 Configure Subscription Options](#).

3.1 Subscription Type

The type of subscription is defined by the 'var' attribute with a value of "{urn:xmpp:pubsub-ext-sub:0}type". This option MUST have the type "list-multi", and it MUST default to only the "items" value. The possible values for this field are:

- **items:** subscribe to items notification, which is equivalent to a usual node subscription as defined at [XEP-0060 §6.1 Subscribe to a Node](#).

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

²XEP-0060: Publish-Subscribe <<https://xmpp.org/extensions/xep-0060.html>>.

- **metadata:** subscribe to metadata updates, as explained in [Metadata Notification](#).
- **linked items:** subscribe to items with "link" relationship to this node, as explained in [Linked Items Notification](#).

3.2 Subscription Depth

The subscription depth is used when a client wants to subscribe to a parent node and all its descendants (up to requested depth). The "parent" relationship is explained in XEP-XXXX Pubsub Node Relationships.

The 'var' attribute for the subscription depth field MUST be "{urn:xmpp:pubsub-ext-sub:0}depth", and it MUST have a type of "text-single".

The value MUST be castable to a signed integer, and it MUST default to "0". This value indicates how many levels of descendants this subscription must apply to:

- "0" for subscribing only to items on this node (the default);
- "1" for subscribing to its items and direct children's items;
- "2" for subscribing to items including up to two descendant levels, and so forth.
- A negative value means unlimited depth is requested (i.e., subscription to the whole arborescence).

However, an implementation MAY limit the maximum depth for performance reasons; in such cases, it MUST be advertised through disco as explained in [Discovering Support](#). When the subscription depth has a value different from "0", subscriptions must dynamically adapt to nodes added or removed within the arborescence: if a node gains a new "parent" relationship to any subscribed node, it will automatically become part of the subscription; conversely, if such a relationship is modified or the node is deleted, it will no longer be included in the subscription.

4 Subscription Request and Notification

4.1 Request

In this example, Romeo is subscribing to the 'some-node-name' node and requesting to receive notifications for items, metadata updates, and linked items within the entire node hierarchy (unlimited depth).

Listing 1: Romeo Subscribes to a Node and its Descendants with Linked Items and Metadata Updates

```
<iq type='set'  
  from='romeo@example.org/orchard'  
  to='pubsub.example.org'
```

```

    id='sub-1'>
<pubsub xmlns='http://jabber.org/protocol/pubsub'>
  <subscribe jid='romeo@example.org'
    node='some-node-name' />
  <options>
    <x xmlns='jabber:x:data' type='submit'>
      <field var='FORM_TYPE' type='hidden'>
        <value>http://jabber.org/protocol/pubsub#subscribe_options</
          value>
      </field>
      <field var='{urn:xmpp:pubsub-ext-sub:0}type'>
        <value>items</value>
        <value>metadata</value>
        <value>linked items</value>
      </field>
      <field var='{urn:xmpp:pubsub-ext-sub:0}depth'>
        <value>-1</value>
      </field>
    </x>
  </options>
</pubsub>
</iq>

```

4.1.1 Success Case

If the service allows the subscription, it **MUST** inform the requesting entity that it is now subscribed.

Listing 2: Service responds with success

```

<iq type='result'
  from='pubsub.example.org'
  to='romeo@example.org/orchard'
  id='sub-1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <subscription node='some-node-name'
      jid='romeo@example.org'
      subscription='subscribed' />
  </pubsub>
</iq>

```

4.2 Notification Types

This section describes the notifications that are sent for each subscription type: **items**, **metadata**, and **linked items**.

4.2.1 Items Notification

When a user subscribes to items, notifications are sent for items published or retracted within the subscribed node and its descendants (if applicable, and if the entity is allowed to see the item). Each notification contains an <items> element within an <event> element, which includes one or more <item> elements. This is the same notification as explained in [XEP-0060 §7.1.2.1 Notification With Payload](#).

Listing 3: Service Notifies Subscribers of Items

```
<message from='pubsub.example.org' to='juliet@example.org/balcony' id='items-1'>
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
    <items node='some-node-name'>
      <item id='item1'>
        <!--{}- ... -{}-->
      </item>
    </items>
  </event>
</message>
```

4.2.2 Metadata Notification

When a user subscribes to metadata updates, notifications are sent for changes in node metadata. Each notification contains a <configuration> element within an <event> element, which includes the updated metadata. The notification SHOULD include only the updated data.

Listing 4: Service Notifies Subscribers of Metadata Updates

```
<message from='pubsub.example.org' to='juliet@example.org/balcony' id='config-1'>
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
    <configuration node='some-node-name'>
      <x xmlns='jabber:x:data' type='result'>
        <field var='FORM_TYPE' type='hidden'>
          <value>http://jabber.org/protocol/pubsub#meta-data</value>
        </field>
        <field var='pubsub#title'>
          <value>New Title</value>
        </field>
        <field var='pubsub#description'>
          <value>New Description</value>
        </field>
        <field var='pubsub#access_model'>
          <value>open</value>
        </field>
      </x>
    </configuration>
  </event>
</message>
```

```

        </x>
    </configuration>
</event>
</message>

```

When a node is removed from a branch of a **parent** relationship tree-like structure by changing the 'parent' field, and if it is part of a subscription with suitable depth, the last metadata notification is sent. This notifies the XMPP client that the node has been removed by checking the modified 'parent' field.

In the following example, the 'parent' field is changed to remove the node from the branch it was on. Note that the field is present, but set to an empty value.

Listing 5: Service Notifies Subscribers of Node Removal from Tree-Like Structure

```

<message from='pubsub.example.org' to='juliet@example.org/balcony' id=
'config-2'>
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
    <configuration node='removed-node-name'>
      <x xmlns='jabber:x:data' type='result'>
        <field var='FORM_TYPE' type='hidden'>
          <value>http://jabber.org/protocol/pubsub#meta-data</
value>
        </field>
        <field var='{urn:xmpp:pubsub-relationships:0}parent' />
      </x>
    </configuration>
  </event>
</message>

```

4.2.3 Linked Items Notification

When a user subscribes to linked items, notifications are sent for items published or retracted within nodes that have a "link" relationship with the subscribed node. Each notification contains an <items> element within an <event> element, which includes one or more <item> elements, as explained in [Publish-Subscribe \(XEP-0060\)](#)³. The subscription is effective even if the linked relationship was established after the initial subscription.

Listing 6: Service Notifies Subscribers of Linked Items

```

<message from='pubsub.example.org' to='juliet@example.org/balcony' id=
'linked-1'>
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
    <items node='urn:xmpp:pubsub-attachments:1/xmpp:juliet@capulet.lit
?;node=urn%3Axmpp%3Amicroblog%3A0;item=balcony-restoration-
afd1'>

```

³XEP-0060: Publish-Subscribe <<https://xmpp.org/extensions/xep-0060.html>>.


```

    <item id='romeo@montague.lit'>
      <!--{}- ... -{}-->
    </item>
  </items>
</event>
</message>

```

4.2.4 Depth Subscription

When subscribing with a specified depth on a node, an implicit subscription is created for all descendants up to the indicated number of levels (or without limit if the depth is a negative number). The depth implicit subscription depends on the subscription type: if "linked items" are requested, linked items MUST be included; otherwise, they MUST NOT be included (i.e., only descendants with no 'link' field set will trigger notifications).

Implicit subscriptions MUST respect the access model of the node where the item has been published and all its parent nodes as explained in XEP-XXXX: Pubsub Node Relationships §4.4 Parent Relationship Rules.

5 Business Rules

A pubsub service MAY filter out metadata fields from notification, e.g., if they are not relevant in this use case, or if they would trigger too many notifications. This is an implementation-dependent decision.

6 Discovering Support

If a pubsub service supports the protocol specified in this XEP, it MUST advertise it by including the "urn:xmpp:pubsub-ext-sub:0" discovery feature in response to a [Service Discovery \(XEP-0030\)](#)⁴ information request.

Listing 7: Service Discovery Information Request

```

<iq type='get'
  from='juliet@example.org/balcony'
  to='pubsub.example.org'
  id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info' />
</iq>

```

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

Listing 8: Service Discovery Information Response

```

<iq type='result'
  from='pubsub.example.org'
  to='juliet@example.org/balcony'
  id='disco1'>
<query xmlns='http://jabber.org/protocol/disco#info'>
  ...
  <feature var='urn:xmpp:pubsub-ext-sub:0' />
  ...
</query>
</iq>

```

If a pubsub service has a limitation in the maximum supported subscription depth, it **MUST** advertise that fact in the node metadata (as specified in [XEP-0060 §5.4 Discover Node Metadata](#)) with a field which **MUST** have a 'var' attribute set to "{urn:xmpp:pubsub-ext-sub:0}max-depth", a 'type' attribute set to "text-single", and a value which **MUST** be castable to an integer. If no such field is defined, that means the pubsub service has no depth limit.

Listing 9: Pubsub Service Advertises a Max Depth of 20 Layers

```

<iq type='result'
  from='pubsub.example.org'
  to='juliet@example.org/balcony'
  id='meta1'>
<query xmlns='http://jabber.org/protocol/disco#info'
  node='some_node'>
<identity category='pubsub' type='leaf' />
<feature var='http://jabber.org/protocol/pubsub' />
<x xmlns='jabber:x:data' type='result'>
  <field var='FORM_TYPE' type='hidden'>
    <value>http://jabber.org/protocol/pubsub#meta-data</value>
  </field>
  <field var='{urn:xmpp:pubsub-ext-sub:0}max-depth' type='text-single'
    >
    <value>20</value>
  </field>
  <!--{}- ... -{}-->
</x>
</query>
</iq>

```

7 Security Considerations

When doing an implicit subscription, pubsub service **MUST** be sure to respect the access model of implicitly subscribed node and all its parents as explained in [XEP-XXXX Pubsub Node Relationships 4.4 Parent Relationship Rules](#). If an entity is not allowed by any of the

access model, the pubsub service MUST NOT send a notification. Similarly, a metadata update notification MUST NOT be send if the subscribed entity would not otherwise be allowed to see the metadata.

8 IANA Considerations

This document does not require interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)⁵.

9 XMPP Registrar Considerations

TODO

10 Acknowledgements

Thanks to NLNet foundation/NGI Zero Core for funding the work on this specification.

⁵The 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/>.