



XMPP

XEP-0351: Recipient Server Side Notifications Filtering

Sergey Dobrov

<mailto:Binary@JRUdevels.org>

<xmpp:Binary@JRUdevels.org>

2017-09-11

Version 0.2

Status	Type	Short Name
Deferred	Standards Track	NOT_YET_ASSIGNED

This specification defines a modern efficient way to deliver PubSub notifications.

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	2
3.1	Switching protocols	2
3.2	Sending and receiving a notification	2
3.3	Advertising client's filtering preferences	3
4	Security Considerations	4
5	IANA Considerations	4
6	XMPP Registrar Considerations	4
7	XML Schema	4

1 Introduction

[Publish-Subscribe \(XEP-0060\)](#)¹ in conjunction with [Entity Capabilities \(XEP-0115\)](#)² defines a way for a client to get rid of unwanted pubsub notifications and hence save significant amount of traffic. The protocol is quite flexible and reliable and is able to work even with servers that don't support filtering. Unfortunately, this reliability has its costs and the protocol has the following disadvantages:

- it's impossible to be subscribed to a person's feed if the person is not subscribed to you because in this case his server doesn't know your presence information and your capabilities and hence it can't filter the messages appropriately, so you won't receive any notifications at all;
- servers need to know not only their users' capabilities but also the capabilities of all the users that it wants to communicate;
- clients can't decide what kind of notifications they want to receive; node owner decides instead, but clients might not be interested in some of notifications or vice versa.

The most actual example of the problems that are beget with these disadvantages is the [Microblogging Over XMPP \(XEP-0277\)](#)³: you won't be able to subscribe to a user's blog if he doesn't want to read your blog as well. This will be a big problem for popular bloggers that can't read all the feeds of their readers.

Another one is a bot which want to gather information from a lot of users. For instance, we can want to write a bot which gather geo-location from a lot of devices, then it will subscribe to these devices but they may not want to do so. The new protocol will allow such bots to operate this way.

Also, some clients may want to receive retract items events to appropriately update their feed data storage but some other clients may not want to store feeds at all and just show the recent events and corollary don't need in these events. This standard allows the first one to receive those events and the last one to get rid of them saving traffic, regardless of node settings which can be configured solely by the node's owner.

2 Requirements

The new protocol is needed to solve the stated problems that will allow to move the notifications filtering process from the sender side to the recipient side, that will allow us to filter them in a more flexible way. Also, we must care of:

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

²XEP-0115: Entity Capabilities <<https://xmpp.org/extensions/xep-0115.html>>.

³XEP-0277: Microblogging over XMPP <<https://xmpp.org/extensions/xep-0277.html>>.

- backwards-compatibility: users of servers that don't support this XEP should be able to receive the notifications through the old protocol;
- users should be able to control what kind of events they want to receive besides the nodes types.

3 Use Cases

This document should cover the following use cases:

- switching to the new protocol if the recipient's server support it;
- sending a notification;
- receiving a notification by server and broadcasting it to its users;
- advertising client's filtering preferences.

3.1 Switching protocols

Since we require the implementations to be backwards compatible, we need a mechanism for a sender to know if the recipient's server supports the modern protocol or not.

To know this, sender's server should check recipient's capabilities as described at [Entity Capabilities \(XEP-0115\)](#)⁴ in the "Stream Feature" section: if the server supports the "urn:xmpp:rsf:0" feature then the server SHOULD use the protocol that's described here, if the server lacks support of this feature or doesn't provide its capabilities at all then the sender MUST NOT use this protocol.

3.2 Sending and receiving a notification

When using this protocol to deliver notifications, the server MUST follow these rules:

- send all the notifications that can be generated by the pubsub node without any care of node settings to make it possible for users to choose which notifications they want to receive instead.

Listing 1: Sending the notification

```
<message from='user@example.net'  
        to='user1@example.com'>  
  <event xmlns='http://jabber.org/protocol/pubsub#event'>  
    <items node='n48ad4fj78zn38st734'>
```

⁴XEP-0115: Entity Capabilities <<https://xmpp.org/extensions/xep-0115.html>>.

```

    <item id='i1s2d3f4g5h6bjeh936'>
      <geoloc xmlns='http://jabber.org/protocol/geoloc'>
        <description>Venice</description>
        <lat>45.44</lat>
        <lon>12.33</lon>
      </geoloc>
    </item>
  </items>
</event>
</message>
<message from='user@example.net'
  to='user2@example.com'>
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
    <items node='n48ad4fj78zn38st734'>
      <item id='i1s2d3f4g5h6bjeh936'>
        <geoloc xmlns='http://jabber.org/protocol/geoloc'>
          <description>Venice</description>
          <lat>45.44</lat>
          <lon>12.33</lon>
        </geoloc>
      </item>
    </items>
  </event>
</message>
... and so on

```

Recipient server is responsible now to filter out and broadcast that messages with events that were request by each user's resource.

3.3 Advertising client's filtering preferences

As described at [Publish-Subscribe \(XEP-0060\)](#)⁵'s "Filtered notifications" section, client should add payload's NodeID and "+notify" postfix to tell which types of content it can and want to process.

This document extends this filtering feature to allow clients also select which event types it wants to receive.

To advertise user's preferences it should add a feature in its capabilities that MUST consist of payload's NodeID with appended "+events-" postfix and then append a list of necessary payload types. Here are the rules of generating this list:

- list should contain payload types those can be one of the following: config, delete, retract, sub as described in [Publish-Subscribe \(XEP-0060\)](#)⁶;
- order the list ascending;

⁵XEP-0060: Publish-Subscribe <<https://xmpp.org/extensions/xep-0060.html>>.

⁶XEP-0060: Publish-Subscribe <<https://xmpp.org/extensions/xep-0060.html>>.

- concatenate the items through the "-" sign.

For example, the following string will mean that user wants to receive new items, node configuration changes and new subscriptions geolocation events, it should add the following features in its capabilities:

- <http://jabber.org/protocol/geoloc+notify>
- <http://jabber.org/protocol/geoloc+events-config-sub>

The recipient's server MUST drop the notifications that are not in the list. New items notifications are implied by the "+notify" filter. If user doesn't specify the "+events" filter then the server SHOULD treat it as absence of any filtering at all.

4 Security Considerations

PENDING

5 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA).

6 XMPP Registrar Considerations

REQUIRED.

7 XML Schema

This protocol doesn't define any XML stanzas.