



XMPP

XEP-0230: Service Discovery Notifications

Joe Hildebrand
<mailto:jhildebr@cisco.com>
<xmpp:hildjj@jabber.org>

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

2016-10-04
Version 0.1.1

Status	Type	Short Name
Deferred	Standards Track	NOT YET ASSIGNED

This specification defines a method for requesting and receiving notifications regarding XMPP service discovery items.

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

1	Introduction	1
2	Protocol	1
3	Security Considerations	3
4	IANA Considerations	3
5	XMPP Registrar Considerations	4

1 Introduction

[Entity Capabilities \(XEP-0115\)](#)¹ defines a way for entities to receive dynamically-updated information about [Service Discovery \(XEP-0030\)](#)² capabilities, rather than polling for updates by sending repeated disco#info requests. However, there is no comparable method for receiving updated disco#items information (this might be helpful when the number of associated entities is large or dynamic, e.g., at a multi-user chat service). This document specifies such a method by re-using semantics from [Publish-Subscribe \(XEP-0060\)](#)³.

2 Protocol

Before a requesting entity asks for service discovery items with notifications, it SHOULD send directed presence to the responding entity so that the responding entity knows when to stop sending notifications.

Listing 1: Requesting entity sends presence

```
<presence from='bill@shakespeare.lit/globe' to='chat.shakespeare.lit' />
```

The requesting entity then sends a disco#items request to the responding entity, including a <subscribe/> element qualified by the 'http://jabber.org/protocol/pubsub' namespace, which in turn specifies a node of 'http://jabber.org/protocol/disco#items'.

Listing 2: Requesting all items

```
<iq type='get'
  from='bill@shakespeare.lit/globe'
  to='chat.shakespeare.lit'
  id='items1'>
  <query xmlns='http://jabber.org/protocol/disco#items'>
    <subscribe xmlns='http://jabber.org/protocol/pubsub' />
  </query>
</iq>
```

If the responding entity does not recognize inclusion of the <subscribe/> element as a child of the <query/> element, it MUST return the service discovery items but MUST NOT send subsequent notifications to the requesting entity.

If the requesting entity did not share presence with the responding entity, the responding entity MUST return the service discovery items but SHOULD NOT send subsequent notifications to the requesting entity.

¹XEP-0115: Entity Capabilities <<https://xmpp.org/extensions/xep-0115.html>>.

²XEP-0030: Service Discovery <<https://xmpp.org/extensions/xep-0030.html>>.

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

If the responding entity recognizes inclusion of the <subscribe/> element as a child of the <query/> element and the requesting entity is allowed to receive notifications, the responding entity **MUST** return the service discovery items (including a <subscription/> element) and **SHOULD** send subsequent notifications to the requesting entity.

Listing 3: Result-set for all items

```
<iq type='result'
  from='chat.shakespeare.lit'
  to='bill@shakespeare.lit/globe'
  id='items1'>
  <query xmlns='http://jabber.org/protocol/disco#items'>
    <item jid='alls-well-that-ends-well@chat.shakespeare.lit' />
    <item jid='as-you-like-it@chat.shakespeare.lit' />
    <item jid='cleopatra@chat.shakespeare.lit' />
    <item jid='comedy-of-errors@chat.shakespeare.lit' />
    <item jid='coriolanus@chat.shakespeare.lit' />
    <item jid='cymbeline@chat.shakespeare.lit' />
    <item jid='hamlet@chat.shakespeare.lit' />
    <item jid='henry-the-fourth-one@chat.shakespeare.lit' />
    <item jid='henry-the-fourth-two@chat.shakespeare.lit' />
    <item jid='henry-the-fifth@chat.shakespeare.lit' />
    <subscription
      xmlns='http://jabber.org/protocol/pubsub'
      jid='bill@shakespeare.lit'
      subid='some-long-sub-id'
      subscription='subscribed' />
  </query>
</iq>
```

The responding entity then **SHOULD** send subsequent notifications to the requesting entity when associated items are added to or deleted from the potential result set.

Listing 4: A notification (new item)

```
<message from='chat.shakespeare.lit' to='bill@shakespeare.lit' id='foo'>
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
    <items node='http://jabber.org/protocol/disco#items'>
      <item id='ae890ac52d0df67ed7cfd51b644e901'>
        <item xmlns='http://jabber.org/protocol/disco#items'
          jid='henry-the-fifth@chat.shakespeare.lit' />
      </item>
    </items>
  </event>
</message>
```

Listing 5: A notification (deleted item)

```
<message from='chat.shakespeare.lit' to='bill@shakespeare.lit' id='bar
'>
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
    <items node='http://jabber.org/protocol/disco#items' />
      <retract id='fa890ca52d0df67de7fcfd51b644c701'>
        <item xmlns='http://jabber.org/protocol/disco#items'
          jid='cardenio@chat.shakespeare.lit' />
      </retract>
    </items>
  </event>
</message>
```

When the requesting entity goes offline, the responding entity will receive unavailable presence.

Listing 6: Responding entity receives unavailable presence

```
<presence from='bill@shakespeare.lit/globe'
  to='chat.shakespeare.lit'
  type='unavailable' />
```

The responding entity then MUST NOT send further notifications.

3 Security Considerations

This document introduces no new security considerations above and beyond those already defined for service discovery and publish-subscribe.

4 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)⁴.

⁴The 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/>.

5 XMPP Registrar Considerations

This document requires no interaction with the [XMPP Registrar](#)⁵.

⁵The 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/>.