



# XMPP

## XEP-0128: Service Discovery Extensions

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

2004-10-20  
Version 1.0

| Status | Type          | Short Name |
|--------|---------------|------------|
| Active | Informational | N/A        |

This document specifies best practices for including extended information in Service Discovery results.

# 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

|          |                                      |          |
|----------|--------------------------------------|----------|
| <b>1</b> | <b>Introduction</b>                  | <b>1</b> |
| <b>2</b> | <b>Recommendations</b>               | <b>1</b> |
| <b>3</b> | <b>Examples</b>                      | <b>2</b> |
| 3.1      | IM Server . . . . .                  | 2        |
| 3.2      | Multi-User Chat Room . . . . .       | 3        |
| <b>4</b> | <b>Implementation Notes</b>          | <b>4</b> |
| <b>5</b> | <b>Security Considerations</b>       | <b>4</b> |
| <b>6</b> | <b>IANA Considerations</b>           | <b>4</b> |
| <b>7</b> | <b>XMPP Registrar Considerations</b> | <b>5</b> |

## 1 Introduction

Developers periodically wonder why [Service Discovery \(XEP-0030\)](#)<sup>1</sup> does not include more bits of information. For example, why does the `<identity/>` element not include a 'description' attribute, and can we add one now? The answer is: well, it just doesn't, and at this point it's too late to make further changes (since XEP-0030 is Final). So the best approach is to specify a well-defined extension mechanism.

Let us consider an example. A [Multi-User Chat \(XEP-0045\)](#)<sup>2</sup> room might want to include additional information in its service discovery results, such as the full room description, the current discussion topic (room subject), the number of occupants in the room, and the JID of the room owner.

Adding one new attribute to the service discovery schema (even if that were an option) would not solve the problem, since a MUC service might want to provide certain bits of information, whereas a [Publish-Subscribe \(XEP-0060\)](#)<sup>3</sup> service might want to provide other bits.

A better solution would be to include extended information qualified by a namespace that provides a way to flexibly define structured data formats. Thankfully, we already possess such a protocol: [Data Forms \(XEP-0004\)](#)<sup>4</sup>. In addition, we possess a way to define common fields used in data forms: [Field Standardization for Data Forms \(XEP-0068\)](#)<sup>5</sup>. Using these building blocks, we can define some best practices for extending service discovery results.

## 2 Recommendations

If an entity desires to provide extended information about itself in an IQ results stanza within the context of the Service Discovery protocol, it SHOULD do so by including each bit of information as the XML character data of the `<value/>` child of a distinct `<field/>` element, with the entire set of fields contained within an `<x/>` element of type "result" qualified by the 'jabber:x:data' namespace; this `<x/>` element SHOULD be a child of the `<query/>` element qualified by the 'http://jabber.org/protocol/disco#info' namespace. Thus the IQ result SHOULD be of the following form:

```
<iq type='result'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    ...
    <x type='result' xmlns='jabber:x:data'>
      <field var='[var-name]' label='[optional]'>
        <value>[var-value]</value>
      </field>
    ...
  </query>
</iq>
```

<sup>1</sup>XEP-0030: Service Discovery <<https://xmpp.org/extensions/xep-0030.html>>.

<sup>2</sup>XEP-0045: Multi-User Chat <<https://xmpp.org/extensions/xep-0045.html>>.

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

<sup>4</sup>XEP-0004: Data Forms <<https://xmpp.org/extensions/xep-0004.html>>.

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

```

    </x>
  </query>
</iq>

```

Note: A <field/> element MAY contain more than one <value/> child if appropriate. If the data fields are to be used in the context of a protocol approved by the XMPP Standards Foundation, they SHOULD be described in the relevant XMPP Extension Protocol specification and registered in accordance with the rules defined in XEP-0068, resulting in the inclusion of a <field/> element whose 'var' attribute has a value of "FORM\_TYPE" and whose 'type' attribute has a value of "hidden".

An entity MUST NOT supply extended information about associated children communicated via the 'http://jabber.org/protocol/disco#items' namespace, since a core principle of Service Discovery is that an entity must define its own identity only and must not define the identity of any children associated with the entity.

## 3 Examples

### 3.1 IM Server

The following is an example of including a disco extension in the IQ result sent by a standard instant messaging server.

Listing 1: Entity Queries Server for Information

```

<iq type='get'
  from='capulet.com'
  to='shakespeare.lit'
  id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info' />
</iq>

<iq type='result'
  from='shakespeare.lit'
  to='capulet.com'
  id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    <identity
      category='server'
      type='im'
      name='shakespeare.lit_jabber_server' />
    <feature var='jabber:iq:register' />
    <x xmlns='jabber:x:data' type='result'>
      <field var='FORM_TYPE' type='hidden'>
        <value>http://jabber.org/network/serverinfo</value>
      </field>
      <field var='c2s_port'>

```

```

    <value>5222</value>
  </field>
  <field var='c2s_port_ssl'>
    <value>5223</value>
  </field>
  <field var='http_access'>
    <value>http://shakespeare.lit/jabber</value>
  </field>
  <field var='ip_version'>
    <value>ipv4</value>
    <value>ipv6</value>
  </field>
  <field var='info_url'>
    <value>http://shakespeare.lit/support.php</value>
  </field>
</x>
</query>
</iq>

```

### 3.2 Multi-User Chat Room

The following is an example of including a disco extension in the IQ result sent by a Multi-User Chat room.

Listing 2: User Queries Room for Information

```

<iq type='get'
  from='hag66@shakespeare.lit/pda'
  to='darkcave@macbeth.shakespeare.lit'
  id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info' />
</iq>

<iq type='result'
  from='darkcave@macbeth.shakespeare.lit'
  to='hag66@shakespeare.lit/pda'
  id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    <identity
      category='conference'
      type='text'
      name='A_Dark_Cave' />
    <feature var='http://jabber.org/protocol/muc' />
    <feature var='jabber:iq:register' />
    <x xmlns='jabber:x:data' type='result'>
      <field var='FORM_TYPE' type='hidden'>
        <value>http://jabber.org/protocol/muc#roominfo</value>
      </field>
    </x>
  </query>
</iq>

```

```
<field var='muc#roominfo_description' label='Description'>
  <value>The place for all good witches!</value>
</field>
<field var='muc#roominfo_subject' label='Subject'>
  <value>Spells</value>
</field>
<field var='muc#roominfo_occupants' label='Number_of_occupants'>
  <value>3</value>
</field>
<field var='muc#roominfo_lang' label='Language_of_discussion'>
  <value>en</value>
</field>
</x>
</query>
</iq>
```

## 4 Implementation Notes

In general, the XMPP Standards Foundation may choose to define at most one `FORM_TYPE` for each service discovery identity (category+type) registered with the XMPP Registrar. In addition, particular applications may define application-specific `FORM_TYPES` as well, and one entity may have multiple service discovery identities (e.g., an XMPP server might also function as a publish-subscribe service). Therefore, it is possible (and allowed) for a single service discovery result to contain multiple service discovery extension elements (potentially up to two elements for each identity). However, in practice it is unlikely that any given service discovery result will contain more than one service discovery extension element.

## 5 Security Considerations

Applications SHOULD ensure that information disclosed in a disco extension is appropriate for discovery by any entity on the network.

## 6 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)<sup>6</sup>.

---

<sup>6</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

This document requires no interaction with the [XMPP Registrar](#)<sup>7</sup>; however, specifications following the best practices defined herein may register FORM\_TYPES and field values with the XMPP Registrar.

---

<sup>7</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 <https://xmpp.org/registrar/>.