



XMPP

XEP-0485: PubSub Server Information

Guus der Kinderen

<mailto:guus.der.kinderen@gmail.com>

<xmpp:guus.der.kinderen@igniterealtime.org>

2024-03-10

Version 0.1.0

Status	Type	Short Name
Experimental	Standards Track	serverinfo

This document defines a data format whereby basic information of an XMPP domain can be expressed and exposed over pub-sub.

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	Discovering Support	1
4	Data Format	3
5	Publication	4
6	Implementation Notes	5
7	Privacy Considerations	5
8	XMPP Registrar Considerations	5
8.1	Protocol Namespaces	5
8.2	Field Standardization	6
9	XML Schema	6
10	Acknowledgements	8

1 Introduction

To facilitate discovery of information of individual domains in an XMPP-based network, this specification defines a data format to define basic information for individual XMPP domains. By leveraging [Publish-Subscribe \(XEP-0060\)](#)¹ this information can efficiently be shared with applications that compose an overview of the larger XMPP network.

2 Requirements

- Describe links between nodes in an XMPP-based network, by enumerating connections used for federation between XMPP domains.
- An extensible data format, allowing additional data (such as that defined in [Software Version \(XEP-0092\)](#)²) to be retrievable without requiring additional round-trips.

3 Discovering Support

Domains supporting the publication of Server Information data, as described in this document, MUST advertise the fact by announcing a [Service Discovery \(XEP-0030\)](#)³ feature of 'urn:xmpp:serverinfo:0'. This signifies that an administrative entity approved the publication of data, which is important for the opt-in mechanism described in [Privacy Considerations section](#) of this document.

The pub-sub service address and node in which Server Information data is advertised SHOULD be specified using a [Service Discovery Extensions \(XEP-0128\)](#)⁴, using an URI as specified in section 12.22 of XEP-0060. These pub-sub coordinates MUST be scoped using a FORM_TYPE of "http://jabber.org/network/serverinfo" (as specified in XEP-0157) and data form field registered for this purpose as defined in the [XMPP Registrar Considerations section](#) of this document.

When the 'urn:xmpp:serverinfo:0' feature but no corresponding Service Discovery Extension is advertised, the node that is used will be a first-level leaf node using the name 'serverinfo' on the first pub-sub service advertised through service discovery.

Listing 1: Service Discovery information request

```
<iq type='get'
  from='francisco@denmark.lit/barracks'
  to='shakespeare.lit'
  id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info' />
</iq>
```

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

²XEP-0092: Software Version <<https://xmpp.org/extensions/xep-0092.html>>.

³XEP-0030: Service Discovery <<https://xmpp.org/extensions/xep-0030.html>>.

⁴XEP-0128: Service Discovery Extensions <<https://xmpp.org/extensions/xep-0128.html>>.

```
</iq>
```

Listing 2: Service Discovery information response

```
<iq type='result'
  from='shakespeare.lit'
  to='francisco@denmark.lit/barracks'
  id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    ...
    <feature var='urn:xmpp:serverinfo:0' />
    ...
    <x xmlns='jabber:x:data' type='result'>
      <field var='FORM_TYPE' type='hidden'>
        <value>http://jabber.org/network/serverinfo</value>
      </field>
      <field var='serverinfo-pubsub-node'>
        <value>xmpp:pubsub.shakespeare.lit?;node=serverinfo</value>
      </field>
    </x>
  </query>
</iq>
```

The node MUST reference a first-level leaf node on a pub-sub service.

Listing 3: Entity queries root node for information

```
<iq type='get'
  from='francisco@denmark.lit/barracks'
  to='pubsub.shakespeare.lit'
  id='info1'>
  <query xmlns='http://jabber.org/protocol/disco#info'
    node='serverinfo' />
</iq>
```

Listing 4: Service responds with identity of pubsub/leaf

```
<iq type='result'
  from='pubsub.shakespeare.lit'
  to='francisco@denmark.lit/barracks'
  id='info1'>
  <query xmlns='http://jabber.org/protocol/disco#info'
    node='serverinfo'>
    ...
    <identity category='pubsub' type='leaf' />
    ...
  </query>
</iq>
```

4 Data Format

The data format uses an element named 'serverinfo' in the namespace 'urn:xmpp:serverinfo:0'. In its minimal form, it defines each XMPP domain name served by the local server in an attribute named 'name'.

Listing 5: Minimal Data Format

```
<serverinfo xmlns='urn:xmpp:serverinfo:0'>
  <domain name='shakespeare.lit' />
</serverinfo>
```

The optional 'federation' child element is used to denote remote XMPP domains with which the local domain is federating. Each of them are represented by an element named 'remote-domain'. The domain name of the peer in an optional attribute named 'name'. Optionally, each actual (e.g. TCP) connection from the local server to the peer is added as a 'connection' child-element to the 'remote-domain' element, that has an optional 'type' attribute, defining the directionality of the connection (one of 'incoming', 'outgoing' and 'bidi').

The name of a remote domain MUST only be included if the remote server advertises support for this XEP. This acts as an opt-in mechanism, to address the privacy concern defined in the [Privacy Considerations](#) section of this document.

Listing 6: Data Format with Federated Domains

```
<serverinfo xmlns="urn:xmpp:serverinfo:0">
  <domain name="shakespeare.lit">
    <federation>
      <remote-domain name='denmark.lit'>
        <connection type="incoming" />
        <connection type="outgoing" />
      </remote-domain>
      <remote-domain name='montague.net'>
        <connection type="bidi" />
      </remote-domain>
    </federation>
  </domain>
</serverinfo>
```

Additional data MAY be included as child-elements of the 'serverinfo' element or any of the 'domain' elements. Such data MUST be namespaced appropriately. The example below uses the 'query' element defined in [Software Version \(XEP-0092\)](#) ⁵ to include information about the software application associated with the local server.

Listing 7: Data Format with Software Version

⁵XEP-0092: Software Version <<https://xmpp.org/extensions/xep-0092.html>>.

```
<serverinfo xmlns="urn:xmpp:serverinfo:0">
  <domain name="shakespeare.lit">
    <federation>
      <remote-domain name='montague.net'>
        <connection type="bidi"/>
      </remote-domain>
    </federation>
  </domain>
  <query xmlns='jabber:iq:version'>
    <name>Openfire</name>
    <version>4.8.0</version>
    <os>Windows 11</os>
  </query>
</serverinfo>
```

5 Publication

The data is to be published using a pub-sub node named 'serverinfo' that MUST be a first-level leaf node of a pub-sub service for the domain. It is RECOMMENDED that the leaf-node is configured to have an open access model and contain a maximum of 1 item.

Listing 8: Publish ServerInfo Item

```
<iq type='set'
  from='william@shakespeare.lit/atwork'
  to='pubsub.shakespeare.lit'
  id='publish1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='serverinfo'>
      <item id='current'>
        <serverinfo xmlns="urn:xmpp:serverinfo:0">
          <domain name="shakespeare.lit">
            <federation>
              <remote-domain name='denmark.lit'>
                <connection type="incoming"/>
                <connection type="outgoing"/>
              </remote-domain>
              <remote-domain name='montague.net'>
                <connection type="bidi"/>
              </remote-domain>
            </federation>
          </domain>
        </serverinfo>
      </item>
    </publish>
  </pubsub>
</iq>
```

6 Implementation Notes

As certain information can be expected to be updated continuously and frequently, the server MAY choose to reduce the frequency of updates of the 'serverinfo' pub-sub node.

7 Privacy Considerations

When multiple domains publish their connections to named remote domains, an information leak occurs: by collecting these public statistics, behavioral data of those remote domains can be deduced. To prevent undesired privacy-sensitive information leaks, a domain MUST NOT publish the name of a remote domain, unless that domain advertises support for this XEP, as defined in the [Discovering Support section](#) of this document.

This way, the service discovery mechanism doubles as an opt-in mechanism. Domains that advertise support for this XEP allow other domains to reference them by domain-name in the data that they publish. The mere presence of an applicable pub-sub node MUST NOT be used for Service Discovery purposes, as under common service configuration, non-administrative users are allowed to create such nodes.

8 XMPP Registrar Considerations

Upon advancement of this specification from a status of Experimental to a status of Draft, the [XMPP Registrar](#)⁶ shall include the following information in its registries.

8.1 Protocol Namespaces

This specification defines the following XML namespaces:

- urn:xmpp:serverinfo:0

The [XMPP Registrar](#)⁷ shall add the foregoing namespace to the registry located at [<https://xmpp.org/registrar/namespaces.html>](https://xmpp.org/registrar/namespaces.html), as described in Section 4 of [XMPP Registrar Function \(XEP-0053\)](#)⁸.

⁶The 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/>.

⁷The 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/>.

⁸XEP-0053: XMPP Registrar Function [<https://xmpp.org/extensions/xep-0053.html>](https://xmpp.org/extensions/xep-0053.html).

8.2 Field Standardization

Field Standardization for Data Forms (XEP-0068) ⁹ defines a process for standardizing the fields used within Data Forms qualified by a particular namespace, and Service Discovery Extensions (XEP-0128) ¹⁰ describes how to use field standardization in the context of service discovery. This section registers fields for server information scoped by the "http://jabber.org/network/serverinfo" FORM_TYPE.

```
<form_type>
  <name>http://jabber.org/network/serverinfo</name>
  <doc>XEP-0XXX</doc>
  <desc>
    Forms advertising the coordinates of a pub-sub service and node
    for publication of Server Information data.
  </desc>
  <field
    var='serverinfo-pubsub-node'
    type='text-single'
    label='An_URI_(per_XEP-0060_section_12.22)_identifying_the_pub-
    sub_node_on_which_Server_Information_data_is_published.'/>
</form_type>
```

Note that the FORM_TYPE used by Contact Addresses for XMPP Services (XEP-0157) ¹¹ is purposefully re-used by this XEP, to circumvent the restriction of having at most one XMPP Standards Foundation defined FORM_TYPE for a service discovery identity, as defined in Service Discovery Extensions (XEP-0128) ¹². When a service supports both features, the data in both forms SHOULD be merged into one form.

9 XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>

<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='urn:xmpp:serverinfo:0'
  xmlns='urn:xmpp:serverinfo:0'
  elementFormDefault='qualified'>

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
```

⁹XEP-0068: Field Data Standardization for Data Forms <<https://xmpp.org/extensions/xep-0068.html>>.

¹⁰XEP-0128: Service Discovery Extensions <<https://xmpp.org/extensions/xep-0128.html>>.

¹¹XEP-0157: Contact Addresses for XMPP Services <<https://xmpp.org/extensions/xep-0157.html>>.

¹²XEP-0128: Service Discovery Extensions <<https://xmpp.org/extensions/xep-0128.html>>.

```

    XEP-0XXX: http://www.xmpp.org/extensions/xep-0XXX.html
  </xs:documentation>
</xs:annotation>

<xs:element name="serverinfo" type="urn:serverinfoType" xmlns:urn="
  urn:xmpp:serverinfo:0"/>

<xs:simpleType name="directionType" final="restriction" >
  <xs:restriction base="xs:string">
    <xs:enumeration value="incoming" />
    <xs:enumeration value="outgoing" />
    <xs:enumeration value="bidi" />
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="connectionType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute type="directionType" name="type" use="optional"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:complexType name="remote-domainType">
  <xs:sequence>
    <xs:element type="urn:connectionType" name="connection"
      maxOccurs="unbounded" minOccurs="0" xmlns:urn="
      urn:xmpp:serverinfo:0"/>
  </xs:sequence>
  <xs:attribute type="xs:string" name="name" use="optional"/>
</xs:complexType>

<xs:complexType name="federationType">
  <xs:sequence>
    <xs:element type="urn:remote-domainType" name="remote-domain"
      maxOccurs="unbounded" minOccurs="0" xmlns:urn="
      urn:xmpp:serverinfo:0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="domainType">
  <xs:sequence>
    <xs:element type="urn:federationType" name="federation"
      xmlns:urn="urn:xmpp:serverinfo:0"/>
  </xs:sequence>
  <xs:attribute type="xs:string" name="name"/>
</xs:complexType>

```

```
<xs:complexType name="serverinfoType">
  <xs:sequence>
    <xs:element type="urn:domainType" name="domain" xmlns:urn="
      urn:xmpp:serverinfo:0"/>
  </xs:sequence>
</xs:complexType>

</xs:schema>
```

10 Acknowledgements

Inspiration was taken from the (now defunct) 'server info' crawler by Thomas Leister. Many thanks to Dave Cridland, as well as 'zoidberg' and 'chewie' from the Ignite Realtime community for helping to test the initial implementation of a graphing implementation based on this XEP and to Florian Schmaus, Matthew Wild, Jonas Schäfer and Kevin Smith for their feedback on the earliest drafts of this document.