XEP-0092: Software Version

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

Status	Type	Short Name
Draft	Standards Track	iq-version

This specification defines an XMPP protocol extension for retrieving information about the software application associated with an XMPP entity. The protocol enables one entity to explicitly query another entity, where the response can include the name of the software application, the version of the software application, and the operating system on which the application is running.

Legal

Copyright

Permissions

Warranty

Liability

Conformance

Contents

1 Introduction

The Jabber protocols have long included a method for discovering version information about the software running at another entity's JID. This method makes use of the 'jabber:iq:version' namespace and has been documented variously in Internet-Drafts and elsewhere. Because this protocol is not required by RFC 2779 [1], the 'jabber:iq:version' namespace was removed from XMPP IM [2]. This specification fills the void for canonical documentation.

Note Well: The jabber:iq:version protocol SHOULD NOT be used to determine the identity of entities from which an application receives presence (e.g., contacts in a user's roster and certain kinds of gateways); Entity Capabilities (XEP-0115) [3] SHOULD be used instead. However, the jabber:iq:version protocol MAY be used to determine the identity of entities from which an application does not receive presence (e.g., servers and many kinds of components). The jabber:iq:version protocol MAY also be used to determine information available only via jabber:iq:version (e.g., operating system information) for contacts from which a user receives presence, but only if the user specifically requests such information for a particular contact.

2 Protocol

The 'jabber:iq:version' namespace provides a standard way for Jabber entities to exchange information about the software version used by the entities. The information is communicated in a request/response pair using an <iq/> element that contains a <query/> scoped by the 'jabber:iq:version' namespace. The following children of the <query/> are allowed in an IQ result:

- <name/> -- The natural-language name of the software. This element is REQUIRED in a result.

- <version/> -- The specific version of the software. This element is REQUIRED in a result.

- <os/> -- The operating system of the queried entity. This element is OPTIONAL in a result (see also the Security Considerations).

3 Examples

Listing 1: Querying Another Entity for its Software Version

```
<iq
    type='get'
    from='romeo@montague.net/orchard'
```

[1] RFC 2779: A Model for Presence and Instant Messaging <http://tools.ietf.org/html/rfc2779>.

[2] RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <http://tools.ietf.org/html/rfc6121>.

[3] XEP-0115: Entity Capabilities <http://xmpp.org/extensions/xep-0115.html>.

```
    to='juliet@capulet.com/balcony'
    id='version_1'>
  <query xmlns='jabber:iq:version'/>
</iq>
```

Listing 2: Receiving a Reply Regarding Software Version

```
<iq
    type='result'
    to='romeo@montague.net/orchard'
    from='juliet@capulet.com/balcony'
    id='version_1'>
  <query xmlns='jabber:iq:version'>
    <name>Exodus</name>
    <version>0.7.0.4</version>
    <os>Windows-XP 5.01.2600</os>
  </query>
</iq>
```

The standard error conditions described in Error Condition Mappings (XEP-0086) [4] apply (e.g., service unavailable if the entity does not support the namespace).

4 Determining Support

In order for a requesting entity to determine if a responding entity supports this protocol, it SHOULD send a Service Discovery (XEP-0030) [5] information request to the responding entity:

Listing 3: Requesting entity queries responding entity regarding protocol support

```
<iq from='stpeter@jabber.org/roundabout'
    to='conference.jabber.org'
    id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info'/>
</iq>
```

Listing 4: Responding entity communicates protocol support

```
<iq from='conference.jabber.org'
    to='stpeter@jabber.org/roundabout'
    id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    ...
    <feature var='jabber:iq:version'/>
    ...
```

---

[4]XEP-0086: Error Condition Mappings <http://xmpp.org/extensions/xep-0086.html>.
[5]XEP-0030: Service Discovery <http://xmpp.org/extensions/xep-0030.html>.

```
    </query>
</iq>
```

5 Security Considerations

Revealing the application's underlying operating system may open the user or system to attacks directed against that operating system; therefore, an application MUST provide a way for a human user or administrator to disable sharing of information about the operating system.

6 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [6].

7 XMPP Registrar Considerations

The 'jabber:iq:version' namespace is registered in the protocol namespaces registry maintained by the XMPP Registrar [7].

8 XML Schema

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

<xs:schema
    xmlns:xs='http://www.w3.org/2001/XMLSchema'
    targetNamespace='jabber:iq:version'
    xmlns='jabber:iq:version'
    elementFormDefault='qualified'>

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-0092: http://www.xmpp.org/extensions/xep-0092.html
```

---

[6]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]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 <http://xmpp.org/registrar/>.

```
    </xs:documentation>
  </xs:annotation>

  <xs:element name='query'>
    <xs:complexType>
      <xs:sequence minOccurs='0'>
        <xs:element name='name' type='xs:string' minOccurs='1'/>
        <xs:element name='version' type='xs:string' minOccurs='1'/>
        <xs:element name='os' type='xs:string' minOccurs='0'/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```