



XMPP

XEP-0186: Invisible Command

Peter Saint-Andre
<mailto:xsf@stpeter.im>
<xmpp:peter@jabber.org>
<http://stpeter.im/>

2018-12-06
Version 0.14.0

Status	Type	Short Name
Deferred	Standards Track	invisible

This document specifies an XMPP protocol extension for user invisibility.

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	1
3.1	User Becomes Invisible	1
3.1.1	Server Handling	2
3.1.2	Client Handling	4
3.2	User Becomes Visible	4
4	Discovering Support	5
5	Interoperability Considerations	6
6	Security Considerations	6
7	IANA Considerations	6
8	XMPP Registrar Considerations	6
8.1	Protocol Namespaces	6
8.2	Protocol Versioning	7
9	XML Schema	7
10	Acknowledgements	7

1 Introduction

Some XMPP-based instant messaging systems have long supported the ability for users to be online but to appear offline to other users. This "invisibility" feature was previously defined in nonstandard or complicated ways via [Invisible Presence \(XEP-0018\)](#)¹ and [Invisibility \(XEP-0126\)](#)² (the latter was a profile of [Privacy Lists \(XEP-0016\)](#)³, which is now deprecated). By contrast, this specification defines a standards-compliant protocol extension that can be used over the long term, using an IQ-based protocol that enables an IM user to become "invisible" and "visible" at will within the context of a given session.

2 Requirements

The requirements for invisible mode are straightforward:

1. A user can become visible or invisible at any time within an XMPP session.
2. Invisible mode is active only for the current session; if the user ends that session and starts another session, the invisibility mode set for the previous session does not carry over to the new session.
3. When in invisible mode, a user can send messages and directed presence to particular contacts.

3 Use Cases

3.1 User Becomes Invisible

In order for a client to go invisible, it sends an IQ-set with no 'to' address (thus handled by the user's server) containing an <invisible/> element qualified by the 'urn:xmpp:invisible:1' namespace (see [Namespace Versioning](#) regarding the possibility of incrementing the version number).

The <invisible/> element SHOULD include a 'probe' attribute, which specifies whether the server shall or shall not send presence probes to entities in the user's roster (thus determining whether the user does or does not automatically receive presence notifications from contacts). This attribute is a boolean⁴, where a logical value of TRUE (lexical value of "true" or "1") indicates that the server shall send presence probes and where a logical value of FALSE (lexical value of "false" or "0") indicates that the server shall not send presence probes. The

¹XEP-0018: Invisible Presence <<https://xmpp.org/extensions/xep-0018.html>>.

²XEP-0126: Invisibility <<https://xmpp.org/extensions/xep-0126.html>>.

³XEP-0016: Privacy Lists <<https://xmpp.org/extensions/xep-0016.html>>.

⁴In accordance with Section 3.2.2.1 of XML Schema Part 2: Datatypes, the allowable lexical representations for the xs:boolean datatype are the strings "0" and "false" for the concept 'false' and the strings "1" and "true" for the concept 'true'; implementations MUST support both styles of lexical representation.

default logical value is FALSE.

Listing 1: Invisible command with indication to send presence probes

```
<iq from='bilbo@tolkien.example/shire'  
  id='d1s4pp34r1'  
  type='set'>  
  <invisible probe='true' xmlns='urn:xmpp:invisible:1' />  
</iq>
```

Although the default value is false (thus protecting the user from leaking presence information), the client SHOULD always include the 'probe' attribute.

Listing 2: Invisible command with indication to not send presence probes

```
<iq from='bilbo@tolkien.example/shire'  
  id='d1s4pp34r2'  
  type='set'>  
  <invisible probe='false' xmlns='urn:xmpp:invisible:1' />  
</iq>
```

If the server can successfully process the invisible command, it MUST return an IQ-result.

Listing 3: Invisible command is successful

```
<iq to='bilbo@tolkien.example/shire'  
  id='d1s4pp34r2'  
  type='result' />
```

(Standard XMPP stanza errors apply; see RFC 6120.)

When the client enters invisible mode during a presence session (i.e., after having previously sent undirected available presence with no 'type' attribute), the server MUST send <presence type='unavailable' /> from the specified resource to all contacts who would receive unavailable presence if the client itself were to send <presence type='unavailable' />.

The following sections define server and client handling of inbound and outbound XML stanzas while the client is invisible.

3.1.1 Server Handling

While the client is in invisible mode, the server:

1. MUST NOT broadcast outbound presence notifications as a result of receiving any subsequent undirected presence notifications from the client.

2. MUST deliver outbound directed presence stanzas generated by the client.
3. MUST generate or not generate outbound presence probes in accordance with the value of the 'probe' attribute.
4. MUST deliver inbound <presence/> stanzas.
5. SHOULD deliver inbound <message/> stanzas whose 'to' address is the bare JID <localpart@domain.tld> of the user (subject to standard XMPP stanza handling rules from RFC 6120 and RFC 6121).
6. MUST deliver inbound <message/> and <iq/> stanzas whose 'to' address is the full JID <localpart@domain.tld/resource> corresponding to the resource of the client.
7. MUST deliver outbound <message/> and <iq/> stanzas generated by the client (for an important note regarding presence leaks, see the [Security Considerations](#) section of this document).
8. If there are no other available resources, MUST respond to all IQ-get requests and presence probes sent to the account's bare JID as if the account were offline; this includes but is not limited to the following:
 - If the server responds to a presence probe, the last available presence MUST indicate that the user is unavailable, and if a time is indicated it MUST be the time when the client went invisible.
 - If the server responds to a [Last Activity \(XEP-0012\)](#)⁵ request, the last activity time MUST be the time when the client went invisible.
 - If the server responds to a [Service Discovery \(XEP-0030\)](#)⁶ items request, the response MUST NOT include the invisible resource as one of the account's available items.
9. If after sending directed presence the client then sends <presence type='unavailable'/>, the server MUST deliver that unavailable presence only to the entities to which the client sent directed presence after going invisible.

⁵XEP-0012: Last Activity <<https://xmpp.org/extensions/xep-0012.html>>.

⁶XEP-0030: Service Discovery <<https://xmpp.org/extensions/xep-0030.html>>.

3.1.2 Client Handling

While the client is in invisible mode, it is suggested that the client behave as follows:

- Maintain a temporary list of entities with which communication is allowed and prompt the user before adding any entity to that "communicants list" for this invisibility session; for user convenience, this list might be auto-populated with trusted entities if so configured by the user.
- Prompt the user before sending any outbound traffic (message, presence, or IQ stanza) to another user, even if the user generated such traffic; upon receiving authorization from the user, the client might then add the authorized entity to the communicants list for this invisibility session.

3.2 User Becomes Visible

In order for a client to become visible again, it sends an IQ-set with no 'to' address (thus handled by the user's server) containing a <visible/> element qualified by the 'urn:xmpp:invisible:1' namespace (see [Namespace Versioning](#) regarding the possibility of incrementing the version number).

Listing 4: Visible command

```
<iq from='bilbo@tolkien.example/shire'  
  id='r34pp34r'  
  type='set'>  
  <visible xmlns='urn:xmpp:invisible:1' />  
</iq>
```

If the server can successfully process the visibility command, it MUST return an IQ-result.

Listing 5: Visible command is successful

```
<iq to='bilbo@tolkien.example/shire'  
  id='r34pp34r'  
  type='result' />
```

When the client becomes visible, the server MUST treat that state as equivalent to an active session before receiving initial presence from the client, with one exception: if the client sent directed presence to any entities while in the invisible state, the server MUST treat those entities as under point 2 of Section 4.6.3 of RFC 6121 (i.e., the server MUST ensure that it sends unavailable presence to those entities if the client subsequently goes offline after becoming visible).

If the user wishes to then send presence to all contacts in the roster, it is the responsibility of

the client to send an undirected available presence notification to the server.

Listing 6: Client sends available presence for roster broadcast

```
<presence />
```

The server would then broadcast that presence notification to all entities who would normally receive presence broadcasts from the client (the server MAY also send that notification to any entities to which the client sent directed presence while invisible, whether or not they are in the user's roster).

4 Discovering Support

In order for a client to discover whether its server supports the invisible command, it sends a [Service Discovery \(XEP-0030\)](#)⁷ information request to the server:

Listing 7: Service discovery request

```
<iq from='bilbo@tolkien.example/shire'
  id='d1sc0v3ry'
  to='tolkien.example'
  type='get'>
  <query xmlns='http://jabber.org/protocol/disco#info' />
</iq>
```

If the server supports the invisible command, it MUST return a feature of "urn:xmpp:invisible:1" (see [Namespace Versioning](#) regarding the possibility of incrementing the version number).

Listing 8: Service discovery response

```
<iq from='tolkien.example'
  id='d1sc0v3ry'
  to='bilbo@tolkien.example/shire'
  type='result'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    <feature var='urn:xmpp:invisible:1' />
  </query>
</iq>
```

A client SHOULD complete this service discovery process before sending initial presence to its server (as specified in [Entity Capabilities \(XEP-0115\)](#)⁸, a server can include entity capabilities information in a stream feature, which obviates the need for explicit service discovery as

⁷XEP-0030: Service Discovery <<https://xmpp.org/extensions/xep-0030.html>>.

⁸XEP-0115: Entity Capabilities <<https://xmpp.org/extensions/xep-0115.html>>.

shown above).

5 Interoperability Considerations

Implementers need to be aware that use of the 'probe' attribute is not consistent with the older privacy lists approach defined in XEP-0126.

6 Security Considerations

No matter how it is implemented, invisibility can be defeated and presence leaks can occur without careful stanza handling on the part of the server and the client. Use of the protocol defined here does not necessarily prevent presence leaks, either technically or socially (e.g., if the user reveals his presence to one contact but not another and those contacts are in communication).

7 IANA Considerations

No interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)⁹ is required as a result of this document.

8 XMPP Registrar Considerations

8.1 Protocol Namespaces

This specification defines the following XML namespace:

- urn:xmpp:invisible:1

Upon advancement of this specification from a status of Experimental to a status of Draft, the [XMPP Registrar](#)¹⁰ shall add the foregoing namespace to the registry located at <https://xmpp.org/registrar/namespaces.html>, as described in Section 4 of [XMPP Registrar Function \(XEP-0053\)](#)¹¹.

⁹The 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/>.

¹⁰The 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/>.

¹¹XEP-0053: XMPP Registrar Function <https://xmpp.org/extensions/xep-0053.html>.

8.2 Protocol Versioning

If the protocol defined in this specification undergoes a revision that is not fully backwards-compatible with an older version, the XMPP Registrar shall increment the protocol version number found at the end of the XML namespaces defined herein, as described in Section 4 of XEP-0053.

9 XML Schema

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

<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='urn:xmpp:invisible:1'
  xmlns='urn:xmpp:invisible:1'
  elementFormDefault='qualified'>

  <xs:element name='invisible'>
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base='empty'>
          <xs:attribute name='probe'
            type='xs:boolean'
            use='optional'
            default='false' />
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

  <xs:element name='visible' type='empty' />

  <xs:simpleType name='empty'>
    <xs:restriction base='xs:string'>
      <xs:enumeration value='' />
    </xs:restriction>
  </xs:simpleType>

</xs:schema>
```

10 Acknowledgements

Thanks to Philipp Hancke, Evgeny Khramtsov, Ruslan Marchenko, Kevin Smith, and Matthew Wild for their feedback.