



XMPP

XEP-0186: Invisible Command

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

2008-10-07
Version 0.9

Status	Type	Short Name
Deferred	Standards Track	invisible

This document specifies an XMPP-compatible protocol for user invisibility.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 - 2011 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 <http://xmpp.org/about-xmpp/xsf/xsf-ipr-policy/> or obtained by writing to XMPP Standards Foundation, 1899 Wynkoop Street, Suite 600, Denver, CO 80202 USA).

Contents

1	Introduction	1
2	Requirements	1
3	Use Cases	2
3.1	User Becomes Invisible	2
3.1.1	Server Handling	2
3.1.2	Client Handling	3
3.2	User Becomes Visible	4
4	Discovering Support	5
5	Integration With Privacy Lists	5
6	Security Considerations	6
7	IANA Considerations	6
8	XMPP Registrar Considerations	6
8.1	Protocol Namespaces	6
8.2	Protocol Versioning	6
9	XML Schema	7

1 Introduction

Some instant messaging implementations of the Jabber/XMPP protocols have long supported the ability for IM users to be online but appear invisible. The existing protocols for doing so are:

- [Invisible Presence](#)¹ -- this protocol is not compatible with [XMPP Core](#)² and [XMPP IM](#)³, and the specification does not provide reliable documentation of the protocol in use since many server implementations support presence of type "invisible" but not presence of type "visible".
- [Invisibility](#)⁴ -- this protocol is a somewhat complicated use of [Privacy Lists](#)⁵ for the temporary purpose of appearing invisible rather than the intended purpose of permanently blocking communications; however, the invisible command defined herein can provide a client-friendly interface to the same data store used for privacy lists.

In order to provide a standards-compliant protocol that can be used in the long term, this document defines an IQ-based protocol that enables an IM user to become "invisible" and "visible" at will within the context of a given session. This protocol is intended to supersede the protocol described in XEP-0018 and to provide a more client-friendly approach to invisibility than XEP-0126.

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 directed presence to particular contacts.

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

²RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <<http://tools.ietf.org/html/rfc6120>>.

³RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <<http://tools.ietf.org/html/rfc6121>>.

⁴XEP-0126: Invisibility <<http://xmpp.org/extensions/xep-0126.html>>.

⁵XEP-0016: Privacy Lists <<http://xmpp.org/extensions/xep-0016.html>>.

3 Use Cases

3.1 User Becomes Invisible

In order for a client to go invisible, it shall send an IQ-set with no 'to' address (thus handled by the user's server) containing an <invisible/> element qualified by the 'urn:xmpp:invisible:0' namespace (see [Namespace Versioning](#) regarding the possibility of incrementing the version number).

Listing 1: Invisible command

```
<iq from='bilbo@tolkien.lit/shire'  
  id='inv1'  
  type='set'>  
  <invisible xmlns='urn:xmpp:invisible:0' />  
</iq>
```

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

Listing 2: Invisible command is successful

```
<iq to='bilbo@tolkien.lit/shire'  
  id='inv1'  
  type='result' />
```

(Standard XMPP stanza errors apply; see RFC 3920 and [Error Condition Mappings](#) ⁶.)

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

The following sections define how the server and the client shall handle inbound and out-bound 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 presence notifications as a result of receiving any subsequent undirected presence notifications from the client.
2. MUST deliver directed presence stanzas generated by the client.

⁶XEP-0086: Error Condition Mappings <<http://xmpp.org/extensions/xep-0086.html>>.

3. MUST deliver inbound <presence/> stanzas.
4. 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).
5. 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.
6. 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).
7. 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](#)⁷ request, the last activity time MUST be the time when the client went invisible.
 - If the server responds to a [Service Discovery](#)⁸ items request, the response MUST NOT include the invisible resource as one of the account's available items.
8. 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.

3.1.2 Client Handling

While the client is in invisible mode, the client:

- MUST 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; the list MAY be auto-populated with trusted entities if so configured by the user.

⁷XEP-0012: Last Activity <<http://xmpp.org/extensions/xep-0012.html>>.

⁸XEP-0030: Service Discovery <<http://xmpp.org/extensions/xep-0030.html>>.

- MUST prompt the user before sending any outbound traffic (message, presence, or IQ stanza) to a contact even if the user generated such traffic; upon receiving authorization from the user, the client SHOULD 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 shall send an IQ-set with no 'to' address (thus handled by the user's server) containing a <visible/> element qualified by the 'urn:xmpp:invisible:0' namespace (see [Namespace Versioning](#) regarding the possibility of incrementing the version number).

Listing 3: Visible command

```
<iq from='bilbo@tolkien.lit/shire'  
  id='vis1'  
  type='set'>  
  <visible xmlns='urn:xmpp:invisible:0' />  
</iq>
```

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

Listing 4: Visible command is successful

```
<iq to='bilbo@tolkien.lit/shire'  
  id='vis1'  
  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.

It is the responsibility of the client to send an undirected presence notification to the server.

Listing 5: Client sends presence

```
<presence />
```

The server then MUST broadcast that presence to all entities who would normally receive presence broadcasts from the client (as well as any other entities to which the client sent directed presence while invisible).

4 Discovering Support

In order for a client to discover whether its server supports the protocol defined herein, it MUST send a [Service Discovery](#)⁹ information request to the server:

Listing 6: Service discovery request

```
<iq from='bilbo@tolkien.lit/shire'
  id='disco1'
  to='tolkien.lit'
  type='get'>
  <query xmlns='http://jabber.org/protocol/disco#info' />
</iq>
```

If the server supports the protocol defined herein, it MUST return a feature of "urn:xmpp:invisible:0" (see [Namespace Versioning](#) regarding the possibility of incrementing the version number).

Listing 7: Service discovery response

```
<iq from='bilbo@tolkien.lit/shire'
  id='disco1'
  to='tolkien.lit'
  type='result'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    <feature var='urn:xmpp:invisible:0' />
  </query>
</iq>
```

A client SHOULD complete this service discovery process before sending initial presence to its server.

5 Integration With Privacy Lists

A server MAY use the same backend data store for invisibility mode as defined herein and [Privacy Lists](#)¹⁰ as used for invisibility (see XEP-0126). If so, the server MUST update the relevant privacy lists on behalf of the user when the client requests initiation or termination of invisible mode.

⁹XEP-0030: Service Discovery <<http://xmpp.org/extensions/xep-0030.html>>.

¹⁰XEP-0016: Privacy Lists <<http://xmpp.org/extensions/xep-0016.html>>.

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 guarantee that presence leaks will not occur, 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 \(IANA\)](#)¹¹ 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:0

Upon advancement of this specification from a status of Experimental to a status of Draft, the [XMPP Registrar](#)¹² shall add the foregoing namespace to the registry located at <http://xmpp.org/registrar/namespaces.html>, as described in Section 4 of [XMPP Registrar Function](#)¹³.

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.

¹¹The 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/>.

¹²The 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/>.

¹³XEP-0053: XMPP Registrar Function <http://xmpp.org/extensions/xep-0053.html>.

9 XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>
<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='urn:xmpp:invisible:0'
  xmlns='urn:xmpp:invisible:0'
  elementFormDefault='qualified'>
  <xs:element name='invisible' type='empty' />
  <xs:element name='visible' type='empty' />
  <xs:simpleType name='empty'>
    <xs:restriction base='xs:string'>
      <xs:enumeration value='' />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```