



# XMPP

## XEP-0352: Client State Indication

Matthew Wild

<mailto:mwild1@gmail.com>

<xmpp:me@matthewwild.co.uk>

2020-10-14

Version 1.0.0

Status	Type	Short Name
Draft	Standards Track	csi

This document defines a way for the client to indicate its active/inactive state.

# 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

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Requirements</b>	<b>1</b>
<b>3</b>	<b>Use Cases</b>	<b>1</b>
3.1	User and client behaviour . . . . .	1
3.2	Server behaviour . . . . .	1
<b>4</b>	<b>Protocol</b>	<b>2</b>
4.1	Discovering support . . . . .	2
4.2	Indicating state . . . . .	2
<b>5</b>	<b>Business Rules</b>	<b>3</b>
5.1	In-order processing . . . . .	3
5.2	Interaction with Stream Resumption . . . . .	4
<b>6</b>	<b>Security Considerations</b>	<b>4</b>
<b>7</b>	<b>IANA Considerations</b>	<b>4</b>
<b>8</b>	<b>XMPP Registrar Considerations</b>	<b>4</b>
<b>9</b>	<b>XML Schema</b>	<b>4</b>
<b>10</b>	<b>Acknowledgements</b>	<b>5</b>

## 1 Introduction

It is common for IM clients to be logged in and 'online' even while the user is not interacting with the application. This protocol allows the client to indicate to the server when the user is not actively using the client, allowing the server to optimise traffic to the client accordingly. This can save bandwidth and resources on both the client and server.

## 2 Requirements

The aim of this specification is to provide a simple and efficient protocol for the client to report its state to the server. Exactly how the server uses this information is beyond the scope of this document, although some examples are given.

Other extensions exist, such as [Stanza Interception and Filtering Technology \(XEP-0273\)](#)<sup>1</sup>, which also aim to optimise the traffic between the client and server. A notable difference is that instead of being client-controlled, CSI shifts the responsibility to the server, and aims to just provide the server with enough information to implement various optimisations itself.

## 3 Use Cases

### 3.1 User and client behaviour

Juliet has an XMPP client on her phone, which is available to receive messages. However most of the time Juliet has her phone screen turned off and is not interested in the status of her contacts unless they are communicating with her.

Juliet's client informs the server when Juliet is not interacting with it. The server uses this information to suppress or reduce stanzas that are unimportant, such as status updates.

When Juliet returns to her IM client, the client again informs the server, this time to report that it is active again. The server then disables its traffic optimisations and restores the stream to its normal state.

### 3.2 Server behaviour

When the server knows that the user is not engaging with their client many optimisations become possible. For example a server could:

- Suppress presence updates until the client becomes active again. On becoming active, push the latest presence from each contact.
- Discard messages containing only [Chat State Notifications \(XEP-0085\)](#)<sup>2</sup> payloads.

---

<sup>1</sup>XEP-0273: Stanza Interception and Filtering Technology <<https://xmpp.org/extensions/xep-0273.html>>.

<sup>2</sup>XEP-0085: Chat State Notifications <<https://xmpp.org/extensions/xep-0085.html>>.

- Defer or discard unimportant PEP notifications, possibly unsubscribe from certain PEP nodes until the client becomes active again.

This list is for example only, a server is not required to implement all or any of these, nor is it prevented from implementing other behaviour not listed here. Regardless of what optimisations a server implements, it SHOULD provide a way for administrators to configure them, and MAY provide such configuration to users also (e.g., through an ad-hoc command).

## 4 Protocol

### 4.1 Discovering support

If the server supports CSI, it advertises it in the stream features after the client has authenticated:

Listing 1: Server indicates support for CSI

```
<stream:features>
  <bind xmlns='urn:iETF:params:xmPP-bind' />
  <csi xmlns='urn:xmpp:csi:0' />
</stream:features>
```

### 4.2 Indicating state

A stream always begins in 'active' state. If a client wishes to inform the server that it has become inactive, it sends an <inactive/> element in the 'urn:xmpp:csi:0' namespace:

Listing 2: Client indicates it is inactive

```
<inactive xmlns='urn:xmpp:csi:0' />
```

As might be anticipated, when the client is active again it sends an <active/> element:

Listing 3: Client indicates it is active

```
<active xmlns='urn:xmpp:csi:0' />
```

There is no reply from the server to either of these elements (though they may indirectly cause the server to send stanzas, e.g., to update presence information when the client becomes active after a period of inactivity).

## 5 Business Rules

As this protocol is for indication only, clients MUST NOT make assumptions about how the server will use the active/inactive state information.

The server MUST assume all clients to be in the 'active' state until the client indicates otherwise. Also the CSI active/inactive state is unrelated to the user's presence, the server MUST treat the two independently.

This protocol is intended primarily for clients with human interaction. Due to the open-ended nature of the possible optimisations implemented by the server, it may not be suitable for non-IM purposes where the fully standard behaviour of XMPP is required.

### 5.1 In-order processing

XMPP requires stanzas to be processed in order as per RFC 6120<sup>3</sup> 10.1. Especially "If the server's processing of a particular request could have an effect on its processing of subsequent data it might receive over that input stream..., it MUST suspend processing of subsequent data until it has processed the request.". As a result, all actions triggered by a CSI nonza sent to the server must happen before processing further requests from the same client to the server.

For example: A client sends a CSI active nonza, followed by an XMPP Ping request to the server. The server first changes the CSI state to active and flushes all eventually queued stanzas. After the state has been restored to 'active' and all resulting stanzas have been put on the wire, the server sends the pong.

Listing 4: In-order processing

```

<!--{}- Client sends 'active' and a ping to the server -{}->
<active xmlns='urn:xmpp:csi:0' />
<iq to='capulet.lit' from='juliet@capulet.lit/balcony' id='ping1'
  type='get'>
  <ping xmlns='urn:xmpp:ping' />
</iq>

<!--{}- Server restores stream state to active,
e.g. by flushing out queued stanzas to the client.
and responds to the ping with a pong -{}->
<iq to='juliet@capulet.lit/baclony' from='capulet.lit' id='ping1' type
  ='result' />
<!--{}- Stream state is now 'active' -{}->

```

<sup>3</sup>RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <<http://tools.ietf.org/html/rfc6120>>.

## 5.2 Interaction with Stream Resumption

After a previous stream was resumed using mechanisms like [Stream Management \(XEP-0198\)](#)<sup>4</sup>, the CSI state is *not* restored. That is, stream resumption does not affect the current CSI state, which always defaults to 'active' for new and resumed streams. Clients wishing to immediately go to the inactive state should do so after stream resumption.

## 6 Security Considerations

To protect the privacy of users, servers MUST NOT reveal the clients active/inactive state to other entities on the network.

## 7 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)<sup>5</sup>.

## 8 XMPP Registrar Considerations

This document requires no interaction with [XMPP Registrar](#)<sup>6</sup>.

## 9 XML Schema

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

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

  <xs:element name='csi' type='empty'/>
  <xs:element name='active' type='empty'/>
```

<sup>4</sup>XEP-0198: Stream Management <<https://xmpp.org/extensions/xep-0198.html>>.

<sup>5</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/>>.

<sup>6</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/>>.

```
<xs:element name='inactive' type='empty' />

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

</xs:schema>
```

## 10 Acknowledgements

Thanks to Florian Schmaus for his feedback.