



XMPP

XEP-0012: Last Activity

Jeremie Miller
<mailto:jer@jabber.org>
<xmpp:jer@jabber.org>

Thomas Muldowney
<mailto:temas@jabber.org>
<xmpp:temas@jabber.org>

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

2008-11-26
Version 2.0

Status	Type	Short Name
Final	Standards Track	iq-last

This specification defines an XMPP protocol extension for communicating information about the last activity associated with an XMPP entity. It is typically used by an IM client to retrieve the most recent presence information from an offline contact by sending a last activity request to the server that hosts the account controlled by the contact.

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	Protocol	1
3	Offline User Query	2
4	Online User Query	3
5	Server and Component Query	4
6	Determining Support	5
7	Implementation Notes	6
8	Security Considerations	6
9	IANA Considerations	6
10	XMPP Registrar Considerations	7
11	XML Schema	7

1 Introduction

It is often helpful to know the time of the last activity associated with an entity. The canonical usage is to discover when a disconnected user last accessed its server. The 'jabber:iq:last' namespace provides a method for retrieving that information. The 'jabber:iq:last' namespace can also be used to discover or publicize when a connected user was last active on the server (i.e., the user's idle time) or to query servers and components about their current uptime.

2 Protocol

In order to request last activity information regarding another entity, the requesting entity sends an <iq/> stanza of type "get" to the target entity, containing a <query/> element qualified by the 'jabber:iq:last' namespace:

Listing 1: Last Activity Query

```
<iq from='romeo@montague.net/orchard'
  id='last1'
  to='juliet@capulet.com'
  type='get'>
  <query xmlns='jabber:iq:last' />
</iq>
```

The target entity MUST return either an IQ-result or an IQ-error. When returning an IQ-result, the target entity sends an <iq/> stanza of type='result' containing a <query/> element with a REQUIRED 'seconds' attribute and OPTIONAL XML character data.

Listing 2: Last Activity Response

```
<iq from='juliet@capulet.com'
  id='last1'
  to='romeo@montague.net/orchard'
  type='result'>
  <query xmlns='jabber:iq:last' seconds='903' />
</iq>
```

The requesting entity interprets the IQ-result based on the responding entity's JID type in order to determine the meaning of the information. Specifically, the information means something different depending on the identity of the responding entity:

1. An account registered on an XMPP server, with a JID of the form <localpart@domain.tld>.
2. A connected or available resource associated with such an account, with a JID of the form <localpart@domain.tld/resource>.

3. An "always-on" XMPP server or service, with a JID of the form <domain.tld>.

These differences are explained more fully below.

3 Offline User Query

The primary usage of the 'jabber:iq:last' namespace is to find out how long ago a user logged out (and, additionally, what their status message was at that time). This primary usage assumes that the IQ-get is sent to a bare JID <localpart@domain.tld>. When used in this way, the <query/> element contained in the IQ-result has a 'seconds' attribute, which is the number of seconds that have passed since the user last logged out. In addition, the element MAY contain XML character data that specifies the status message of the last unavailable presence received from the user. An example is shown below:

Listing 3: Last Activity Query

```
<iq from='romeo@montague.net/orchard'
  id='last1'
  to='juliet@capulet.com'
  type='get'>
  <query xmlns='jabber:iq:last' />
</iq>
```

As specified in [XMPP Core](#)¹ and [XMPP IM](#)², an IQ stanza of type "get" sent to a bare JID <localpart@domain.tld> is handled by the user's server on the user's behalf, not delivered to one or more connected or available resources.

If the requesting entity is not authorized to view the user's presence information (normally via a presence subscription as defined in XMPP-IM), the user's server MUST NOT return last activity information but instead MUST return a <forbidden/> error in response to the last activity request.

Listing 4: Requesting Entity is Not Authorized to Retrieve Last Activity Information

```
<iq from='juliet@capulet.com'
  id='last1'
  to='romeo@montague.net/orchard'
  type='error'>
  <error type='auth'>
    <forbidden xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>
```

¹RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <<http://tools.ietf.org/html/rfc6120>>.

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

If the requesting entity is authorized to view the user's presence information, the server shall return information about the last presence activity recorded by the server for that user.

Listing 5: Last Activity Response by Server

```
<iq from='juliet@capulet.com'
  id='last1'
  to='romeo@montague.net/orchard'
  type='result'>
  <query xmlns='jabber:iq:last' seconds='903'>Heading Home</query>
</iq>
```

In this example, the user logged out fifteen minutes and three seconds ago, and when they logged out they sent a presence stanza of type='unavailable' whose <status/> element contained the text "Heading Home".

If the user has at least one connected or available resource when the server receives the request, the response MUST (subject to local security policies) contain an empty <query/> element whose 'seconds' attribute is set to a value of '0'.

4 Online User Query

A requesting entity can send a last activity query to the full JID <local-part@domain.tld/resource> of a connected or available resource associated with an account.

Listing 6: Last Activity Query Sent to Full JID

```
<iq from='romeo@montague.net/orchard'
  id='last2'
  to='juliet@capulet.com/balcony'
  type='get'>
  <query xmlns='jabber:iq:last' />
</iq>
```

In this case, the user's server shall either deliver the IQ to an available resource or respond on behalf of the user.

In particular, as with the offline query use case above, if the requesting entity is not authorized to view the user's presence information (normally via a presence subscription as defined in XMPP IM), the user's server MUST NOT deliver the IQ-get to an available resource but instead MUST return a <forbidden/> error in response to the last activity request.

Listing 7: Requesting Entity is Not Authorized to Retrieve Last Activity Information

```
<iq from='juliet@capulet.com'
  id='last1'
  to='romeo@montague.net/orchard'
```

```

    type='error'>
  <error type='auth'>
    <forbidden xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>

```

If the user's server delivers the IQ-get to one of the user's available resources, the user's client MAY respond with the idle time of the user (i.e., the last time that a human user interacted with the client application).

Listing 8: Last Activity Response by Client

```

<iq from='juliet@capulet.com/balcony'
  id='last2'
  to='romeo@montague.net/orchard'
  type='result'>
  <query xmlns='jabber:iq:last' seconds='123' />
</iq>

```

In the foregoing example, the user has been idle for about two minutes. Support for this functionality is OPTIONAL. A client that does not support the protocol, or that does not wish to divulge this information, MUST return a <service-unavailable/> error.

Listing 9: Service Unavailable Error

```

<iq from='juliet@capulet.com/balcony'
  id='last2'
  to='romeo@montague.net/orchard'
  type='error'>
  <error type='cancel'>
    <service-unavailable xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>

```

If there is no available resource matching the <localpart@domain.tld> in the 'to' attribute of the request, the server MUST follow the rules in XMPP IM in order to determine what error stanza to return.

5 Server and Component Query

When the last activity query is sent to a server or component (i.e., to a JID of the form <domain.tld>), the information contained in the IQ reply reflects the uptime of the JID sending the reply. The seconds attribute specifies how long the host has been running since it was last (re-)started. The <query/> element SHOULD NOT contain XML character data.

Listing 10: Last Activity Query Sent to Server or Service

```
<iq from='romeo@montague.net/orchard'
  id='last3'
  to='capulet.com'
  type='get'>
  <query xmlns='jabber:iq:last' />
</iq>
```

Listing 11: Last Activity Response from Server or Service

```
<iq from='capulet.com'
  id='last3'
  to='romeo@montague.net/orchard'
  type='result'>
  <query xmlns='jabber:iq:last' seconds='123456' />
</iq>
```

In this example, the server has been running for a little more than 34 hours.

6 Determining Support

If an entity supports the Last Activity protocol, it MUST advertise that fact in its responses to [Service Discovery \(XEP-0030\)](#)³ information ("disco#info") requests by returning a feature of "jabber:iq:last".

Listing 12: Requesting entity queries responding entity regarding protocol support

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

Listing 13: Responding entity communicates protocol support

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

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

In order for an application to determine whether an entity supports this protocol, where possible it SHOULD use the dynamic, presence-based profile of service discovery defined in [Entity Capabilities \(XEP-0115\)](#)⁴. However, if an application has not received entity capabilities information from an entity, it SHOULD use explicit service discovery instead.

7 Implementation Notes

The information contained in an IQ reply for this namespace is inherently ambiguous. Specifically, for a bare JID <localpart@domain.tld> the information is the time since the JID was last connected to its server; for a full JID <localpart@domain.tld/resource> the information is the time since the resource was last active in the context of an existing session; and for a bare domain the information is the uptime for the server or component. An application MUST take these differences into account when presenting the information to a human user (if any).

8 Security Considerations

A server MUST NOT allow an unauthorized entity to learn a user's network availability by sending a Last Activity request to a JID of the form <localpart@domain.tld> or <localpart@domain.tld/resource>, since doing so would constitute a "presence leak" as described in RFC 6120. That is, Last Activity information MAY be divulged only to those entities that have permission to view the user's presence via a presence subscription (potentially as restricted by [Privacy Lists \(XEP-0016\)](#)⁵ or [Blocking Command \(XEP-0191\)](#)⁶).

A client MUST provide a way for a human user to disable sending of Last Activity responses from the client's full JID <localpart@domain.tld/resource>.

9 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)⁷.

⁴XEP-0115: Entity Capabilities <<https://xmpp.org/extensions/xep-0115.html>>.

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

⁶XEP-0191: Blocking Command <<https://xmpp.org/extensions/xep-0191.html>>.

⁷The 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/>>.

10 XMPP Registrar Considerations

No action on the part of the [XMPP Registrar](#)⁸ is necessary as a result of this document, since 'jabber:iq:last' is already registered in the protocol namespaces registry located at <https://xmpp.org/registrar/namespaces.html>.

11 XML Schema

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

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

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-0012: http://www.xmpp.org/extensions/xep-0012.html
    </xs:documentation>
  </xs:annotation>

  <xs:element name='query'>
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base='xs:string'>
          <xs:attribute
            name='seconds'
            type='xs:unsignedLong'
            use='optional' />
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

⁸The 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/>.