



XMPP

XEP-0197: User Viewing

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

Kevin Hoekman
<mailto:kevin.hoekman@pandora.be>
<xmpp:kevinh@jabber.org>

2008-09-25
Version 0.3

Status	Type	Short Name
Deferred	Standards Track	NOT_YET_ASSIGNED

This document defines an XMPP protocol extension for communicating information about the television shows, movies, or other videos that a user watches.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2017 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
2.1	Container Element and Child Elements	1
2.2	Transport Mechanism	2
3	Security Considerations	4
4	IANA Considerations	4
5	XMPP Registrar Considerations	4
5.1	Protocol Namespaces	4
5.2	Namespace Versioning	4
6	XML Schema	5

1 Introduction

[Publish-Subscribe \(XEP-0060\)](#)¹ and [Personal Eventing Protocol \(XEP-0163\)](#)² can be used to publish a wide variety of "extended presence" information about users. This document specifies an extended presence payload format that communicates information about the television shows, movies, or other videos that a user watches. This information may be of interest to a user's contacts and can also be used in social networking applications (e.g., IPTV systems).

2 Protocol

2.1 Container Element and Child Elements

Information about video viewing is provided by the user (or automated integration with video systems) and is propagated on the network by the user's client. The information container for video viewing data is a <video/> element that is qualified by the 'urn:xmpp:viewing:0' namespace. The video information itself is provided as the XML character data of the following children of the <video/> element:

Element	Description	Example	Datatype
author	The author of the program	Robert Duncan	xs:string
cast	The cast members of the program	Trace Beaulieu, Frank Conniff, Joel Hodgson, Jim Mal- lon, Kevin Murphy, Michael J. Nelson, Mary Jo Pehl	xs:string
channel_name	The name of the channel showing the program	Sci-Fi Channel	xs:string
description	A description of the program	Terror From The Year 5000	xs:string
duration	The duration of the program (in seconds)	1800	xs:positiveInteger
end_time	The end time of the program	2006-08-29T03:30Z	xs:dateTime
episode	The episode number of the program	807	xs:string
program_name	The name of the program	Mystery Science The- ater 3000	xs:string

¹XEP-0060: Publish-Subscribe <<https://xmpp.org/extensions/xep-0060.html>>.

²XEP-0163: Personal Eventing Protocol <<https://xmpp.org/extensions/xep-0163.html>>.

Element	Description	Example	Datatype
program_type	The type of the program	Comedy	xs:string
start_time	The start time of the program	2006-08-29T03:00Z	xs:dateTime
subprogram_type	The type of the sub-program (e.g., outtakes on an extended DVD)	Cast interview	xs:string
uri	A URI for the video or relevant service	http://www.mst3kinfo.com/	anyURI
user_comment	A user's free-form comments about the program	This is one of my favorite episodes! MST3K forever!	xs:string
user_rating	The user's personal rating of the program (on a scale of 1 to 10, with 10 being highest)	9	xs:string

All elements are OPTIONAL.

NOTE: The datatypes specified above are defined in [XML Schema Part 2](#)³.

2.2 Transport Mechanism

When a user starts watching a video, its client may publish that fact to a PEP node whose NodeID is "urn:xmpp:viewing:0" (see Protocol Namespaces regarding issuance of one or more permanent namespaces) or to a generic pubsub node. Because viewing information is not pure presence information and can change independently of the user's availability, it SHOULD NOT be provided as an extension to the <presence/> stanza type.

Listing 1: User Publishes Viewing Information

```
<iq type='set' from='stpeter@jabber.org/work' id='viewing1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='urn:xmpp:viewing:0'>
      <item id='8eaa8ae16651455376dab4eae45772337e9ec372'>
        <video xmlns='urn:xmpp:viewing:0'>
          <name>Mystery Science Theater 3000</name>
          <episode>807</episode>
          <description>Terror From The Year 5000</description>
          <uri>http://www.mst3kinfo.com/</uri>
        </video>
      </item>
    </publish>
  </pubsub>
</iq>
```

³XML Schema Part 2: Datatypes <<http://www.w3.org/TR/xmlschema11-2/>>.

```

        </video>
      </item>
    </publish>
  </pubsub>
</iq>

```

The viewing information is then delivered to all subscribers:

Listing 2: Viewing Information is Delivered to All Subscribers

```

<message from='stpeter@jabber.org' to='maineboy@jabber.org'>
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
    <items node='urn:xmpp:viewing:0'>
      <item id='8eaa8ae16651455376dab4eae45772337e9ec372'>
        <video xmlns='urn:xmpp:viewing:0'>
          <name>Mystery Science Theater 3000</name>
          <episode>807</episode>
          <description>Terror From The Year 5000</description>
          <uri>http://www.mst3kinfo.com/</uri>
        </video>
      </item>
    </items>
  </event>
</message>

```

When the user stops watching the video, the user's client SHOULD send an empty <video/> element with the same ItemID:

Listing 3: User Publishes Stop Information

```

<iq type='set' from='stpeter@jabber.org/work' id='viewing2'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='urn:xmpp:viewing:0'>
      <item id='8eaa8ae16651455376dab4eae45772337e9ec372'>
        <video xmlns='urn:xmpp:viewing:0' />
      </item>
    </publish>
  </pubsub>
</iq>

```

Listing 4: Stop Information is Delivered to All Subscribers

```

<message from='stpeter@jabber.org' to='maineboy@jabber.org'>
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
    <items node='urn:xmpp:viewing:0'>
      <item id='8eaa8ae16651455376dab4eae45772337e9ec372'>
        <video xmlns='urn:xmpp:viewing:0' />
      </item>
    </items>
  </event>
</message>

```

```
</event>  
</message>
```

3 Security Considerations

The videos that a user watches may be sensitive. A client MUST provide a way for a user to configure which videos or types of videos will not be published (e.g., via user preferences).

4 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)⁴.

5 XMPP Registrar Considerations

5.1 Protocol Namespaces

This specification defines the following XML namespace:

- urn:xmpp:viewing:0

Upon advancement of this specification from a status of Experimental to a status of Draft, the [XMPP Registrar](#)⁵ shall add the foregoing namespace to the registry located at [<https://xmpp.org/registrar/namespaces.html>](https://xmpp.org/registrar/namespaces.html), as described in Section 4 of [XMPP Registrar Function \(XEP-0053\)](#)⁶.

5.2 Namespace 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.

⁴The 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/>.

⁵The 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/>.

⁶XEP-0053: XMPP Registrar Function [<https://xmpp.org/extensions/xep-0053.html>](https://xmpp.org/extensions/xep-0053.html).

6 XML Schema

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

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

  <xs:element name='room'>
    <xs:complexType>
      <xs:sequence minOccurs='0'>
        <xs:element name='author' type='xs:string' minOccurs='0'/>
        <xs:element name='cast' type='xs:string' minOccurs='0'/>
        <xs:element name='channel_name' type='xs:string' minOccurs='0'
          />
        <xs:element name='description' type='xs:string' minOccurs='0'
          >
        <xs:element name='duration' type='xs:positiveInteger'
          minOccurs='0'/>
        <xs:element name='end_time' type='xs:dateTime' minOccurs='0'/>
        <xs:element name='episode' type='xs:string' minOccurs='0'/>
        <xs:element name='program_name' type='xs:string' minOccurs='0'
          />
        <xs:element name='program_type' type='xs:string' minOccurs='0'
          />
        <xs:element name='start_time' type='xs:dateTime' minOccurs='0'
          />
        <xs:element name='subprogram_type' type='xs:string' minOccurs='0'
          />
        <xs:element name='uri' type='xs:anyURI' minOccurs='0'/>
        <xs:element name='user_comment' type='xs:string' minOccurs='0'
          />
        <xs:element name='user_rating' type='xs:string' minOccurs='0'
          >
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```