



XMPP

XEP-0118: User Tune

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

2008-01-30
Version 1.2

Status	Type	Short Name
Draft	Standards Track	tune

This specification defines a payload format for communicating information about music to which a user is listening, including the title, track number, collection, performer, composer, length, and user rating. The payload format is typically transported using the personal eventing protocol, a profile of XMPP publish-subscribe specified in XEP-0163.

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	Implementation Notes	4
4	Security Considerations	4
5	IANA Considerations	4
6	XMPP Registrar Considerations	4
6.1	Protocol Namespaces	4
7	XML Schema	4

1 Introduction

This document defines a protocol for communicating information about the music to which a user is listening. Such information may be seen as a kind of "extended presence", and users may want to communicate such information to their contacts on the network as a fun add-on to traditional IM applications or to provide integration with common music-player applications.

2 Protocol

2.1 Container Element and Child Elements

Information about tunes is provided by the user and propagated on the network by the user's client. The information container for tune data is a <tune/> element that is qualified by the 'http://jabber.org/protocol/tune' namespace. The tune information itself is provided as the XML character data of the following children of the <tune/> element:

Element	Description	Example	Datatype
artist	The artist or performer of the song or piece	Yes	xs:string
length	The duration of the song or piece in seconds	686	xs:unsignedShort
rating	The user's rating of the song or piece, from 1 (lowest) to 10 (highest).	8	xs:positiveInteger
source	The collection (e.g., album) or other source (e.g., a band website that hosts streams or audio files)	Yessongs	xs:string
title	The title of the song or piece	Heart of the Sunrise	xs:string
track	A unique identifier for the tune; e.g., the track number within a collection or the specific URI for the object (e.g., a stream or audio file)	3	xs:string
uri	A URI or URL pointing to information about the song, collection, or artist	http://www.yesworld.com/lyrics/Fragile.html#9	xs:string

NOTE: The datatypes specified above are defined in [XML Schema Part 2](#)¹.

2.2 Transport Mechanism

Tune information SHOULD be communicated and transported by means of the [Publish-Subscribe \(XEP-0060\)](#)² subset specified in [Personal Eventing Protocol \(XEP-0163\)](#)³. Because tune information is not pure presence information and can change independently of the user's availability, it SHOULD NOT be provided as an extension to <presence/>.

Listing 1: User Publishes Tune Information

```
<iq type='set'
  from='stpeter@jabber.org/14793c64-0f94-11dc-9430-000bcd821bfb'
  id='tunes123'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='http://jabber.org/protocol/tune'>
      <item>
        <tune xmlns='http://jabber.org/protocol/tune'>
          <artist>Yes</artist>
          <length>686</length>
          <rating>8</rating>
          <source>Yessongs</source>
          <title>Heart of the Sunrise</title>
          <track>3</track>
          <uri>http://www.yesworld.com/lyrics/Fragile.html#9</uri>
        </tune>
      </item>
    </publish>
  </pubsub>
</iq>
```

The tune information is then delivered to all subscribers:

Listing 2: Tune 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='http://jabber.org/protocol/tune'>
      <item id='bffe6584-0f9c-11dc-84ba-001143d5d5db'>
        <tune xmlns='http://jabber.org/protocol/tune'>
```

¹XML Schema Part 2: Datatypes <<http://www.w3.org/TR/xmlschema11-2/>>.

²XEP-0060: Publish-Subscribe <<https://xmpp.org/extensions/xep-0060.html>>.

³XEP-0163: Personal Eventing Protocol <<https://xmpp.org/extensions/xep-0163.html>>.

```

    <artist>Yes</artist>
    <length>686</length>
    <rating>8</rating>
    <source>Yessongs</source>
    <title>Heart of the Sunrise</title>
    <track>3</track>
    <uri>http://www.yesworld.com/lyrics/Fragile.html#9</uri>
  </tune>
</item>
</items>
</event>
</message>

```

In order to indicate that the user is no longer listening to any tunes (or has simply disabled publication), the user's client shall send an empty <tune/> element, which can be considered a "stop command" for user tunes:

Listing 3: User Disables Publishing

```

<iq type='set'
  from='stpeter@jabber.org/14793c64-0f94-11dc-9430-000bcd821bfb'
  id='tunes345'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='http://jabber.org/protocol/tune'>
      <item>
        <tune xmlns='http://jabber.org/protocol/tune' />
      </item>
    </publish>
  </pubsub>
</iq>

```

Listing 4: Empty Tune 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='http://jabber.org/protocol/tune'>
      <item id='cbbb3f5a-0f9c-11dc-afe0-001143d5d5db'>
        <tune xmlns='http://jabber.org/protocol/tune' />
      </item>
    </items>
  </event>
</message>

```

3 Implementation Notes

To prevent a large number of updates when a user is skipping through tracks, an implementation SHOULD wait several seconds before publishing new tune information.

If the length is unknown (e.g., the user is listening to a stream), the <length/> element SHOULD NOT be included.

A typical interface for user ratings is to show one to five star icons such as . If this interface is used, the numbers 2, 4, 6, 8, and 10 should be mapped to one, two, three, four, and five stars respectively, and odd numbers should be mapped to half stars (e.g., the number 9 would be mapped to four-and-a-half stars).

4 Security Considerations

The publication of user tune information is not known to introduce any new security considerations above and beyond those defined in XEP-0060: Publish-Subscribe.

5 IANA Considerations

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

6 XMPP Registrar Considerations

6.1 Protocol Namespaces

The [XMPP Registrar](#) ⁵ includes 'http://jabber.org/protocol/tune' in its registry of protocol namespaces.

7 XML Schema

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

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/>.

```
xmlns:xs='http://www.w3.org/2001/XMLSchema'
targetNamespace='http://jabber.org/protocol/tune'
xmlns='http://jabber.org/protocol/tune'
elementFormDefault='qualified'>

<xs:annotation>
  <xs:documentation>
    The protocol documented by this schema is defined in
    XEP-0118: http://www.xmpp.org/extensions/xep-0118.html
  </xs:documentation>
</xs:annotation>

<xs:element name='tune'>
  <xs:complexType>
    <xs:sequence minOccurs='0'>
      <xs:element name='artist' type='xs:string' minOccurs='0' />
      <xs:element name='length' type='xs:unsignedShort' minOccurs='0' />
      <xs:element name='rating'>
        <xs:simpleType>
          <xs:restriction base='xs:positiveInteger'>
            <xs:maxInclusive value='10' />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name='source' type='xs:string' minOccurs='0' />
      <xs:element name='title' type='xs:string' minOccurs='0' />
      <xs:element name='track' type='xs:string' minOccurs='0' />
      <xs:element name='uri' type='xs:anyURI' minOccurs='0' />
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```