



XMPP

XEP-0109: Out-of-Office Messages

Darrell Piner
<mailto:dpiner@cisco.com>
<xmpp:dpiner@cisco.com>

Robert Norris
<mailto:rob@cataclysm.cx>
<xmpp:rob@cataclysm.cx>

2022-03-22
Version 0.3.2

Status	Type	Short Name
Deferred	Standards Track	ooo

This document defines an XMPP protocol extension for communicating out-of-office status.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2024 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	Requirements	1
3	Format	1
4	Use cases	2
4.1	Retrieving the current out-of-office settings	2
4.2	Publishing new out-of-office settings	3
4.3	Removing out-of-office settings	5
4.4	Hiding vacation settings	5
5	Server requirements	6
6	Security Considerations	6
7	IANA Considerations	6
8	XMPP Registrar Considerations	7
9	XML Schema	7

1 Introduction

Popular electronic mail systems include features allowing users to set up automated messages that are returned to message senders when the user is not able to deal with the message immediately. This feature is commonly known as "out-of-office" or "vacation" messages, because it is most commonly used when a user is unable to read their messages because they are out of the office (e.g., on vacation).

This document describes a similar system that enables XMPP users to setup out-of-office messages when they are away.

The basis for the e-mail system mechanism is considering an incoming message at the server level and determining whether to respond with a canned out-of-office message. While a system may send a response for each incoming message, many systems provide an optimization where a response to subsequent e-mail messages is sent after some period of time expires.

2 Requirements

The requirements for this feature are fairly straightforward. A user MUST be able to:

- Retrieve their current out-of-office settings.
- Set new out-of-office settings.
- Remove out-of-office settings.
- Notify interested parties of the out-of-office settings when it is in effect.

All these requirements are satisfied by [Personal Eventing Protocol \(XEP-0163\)](#)¹, which is a subset of [Publish-Subscribe \(XEP-0060\)](#)².

3 Format

The out-of-office setting contains a start-time, an end-time and a short message text.

Listing 1: Sample Out-of-Office message

```
<start>2003-07-06T10:30:00+10:00</start>
<end>2003-07-13T08:00:00+10:00</end>
<message>I'm attending OSCON in sunny Portland and won't be able to
  read your message until I get back. If it's urgent, please
  send_email_to_rob@cataclysm.cx.</message>
```

¹XEP-0163: Personal Eventing Protocol <<https://xmpp.org/extensions/xep-0163.html>>.

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

The start and end times are informational only; i.e., the server does nothing with these values other than store them. PEP does not include node lifetime management (i.e., start and end times for the node); therefore, the server is unable to enforce any start and end times included in the settings.

Using the PEP mechanism a client creates out-of-office settings on a PEP node with the desired access model (such as Presence or Open). When a user (or their client) sends presence containing CAPS (see [Entity Capabilities \(XEP-0115\)](#)³) with an entry for out-of-office to a contact with an out-of-office message, the user's client is notified of the out-of-office message and may display, in a client-defined fashion, the out-of-office settings information.

Clients may rely on the PEP node for notifications of changes as well as automatic notification when the user logs-in (i.e., a user is notified of the current item on the out-of-office PEP node of another user on sending initial presence).

4 Use cases

4.1 Retrieving the current out-of-office settings

A user may request their current out-of-office message by sending an IQ get to the local server like so:

Listing 2: Retrieving the current out-of-office settings

```
<iq type='get'
  from='user@example.com'
  to='example.com'
  id='get1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <items node='urn:xmpp:ooo:0' />
  </pubsub>
</iq>
```

Listing 3: Server returns out-of-office settings

```
<iq type='result'
  from='example.com'
  id='get1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <items node='urn:xmpp:ooo:0'>
      <item_id='current'>
        <ooo_xmlns='urn:xmpp:ooo:0'>
          <start>2003-07-06T10:30:00+10:00</start>
          <end>2003-07-13T08:00:00+10:00</end>
          <message>I'm attending OSCON in sunny Portland and won't be_
            able_to
```

³XEP-0115: Entity Capabilities <<https://xmpp.org/extensions/xep-0115.html>>.

```

.....read_your_message_until_I_get_back._If_it's_urgent,
      please
                send_email_to_rob@cataclysm.cx.</message>
      </ooo>
    </item>
  </items>
</pubsub>
</iq>

```

The <start/> and <end/> elements define the times between which this vacation message should be considered valid by a supporting client; the times are in the format specified by [XMPP Date and Time Profiles \(XEP-0082\)](#)⁴.

The <message/> element contains the text of the message which the client may display for the user (when appropriate).

If the user has no stored vacation message, the user will receive a result like the following:

Listing 4: User does not have any out-of-office settings

```
<iq type='result' id='get1' />
```

4.2 Publishing new out-of-office settings

A user may set new vacation message by publishing a new item with an id of 'current' to the out-of-office node:

Listing 5: Publishing new out-of-office settings

```

<iq type='set'
  from='user@example.com/client'
  to='example.com'
  id='publish1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='urn:xmpp:ooo:0'>
      <item id='current'>
        <ooo xmlns='urn:xmpp:ooo:0'>
          <start>2003-07-06T10:30:00+10:00</start>
          <end>2003-07-13T08:00:00+10:00</end>
          <message>I'm_attending_OSCON_in_sunny_Portland_and_won't_be
            able_to
              read_your_message_until_I_get_back.If_it's_urgent,
                _please
          .....send_email_to_rob@cataclysm.cx.</message>
          .....</ooo>
        </item>
      </publish>

```

⁴XEP-0082: XMPP Date and Time Profiles <<https://xmpp.org/extensions/xep-0082.html>>.

```

</pubsub>
</iq>

```

Listing 6: Out-of-office settings published successfully

```

<iq type='result'
  from='example.com'
  to='user@example.com/client'
  id='publish1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='urn:xmpp:ooo:0'>
      <item id='current'>
        <ooo xmlns='urn:xmpp:ooo:0'>
          <start>2003-07-06T10:30:00+10:00</start>
          <end>2003-07-13T08:00:00+10:00</end>
          <message>I'm attending OSCON in sunny Portland and won't be
            able to
              read your message until I get back. If it's urgent
                , please
                  send_email_to_rob@cataclysm.cx.</message>
            </ooo>
          </item>
        </publish>
      </pubsub>
    </iq>

```

And by design, PEP sends a notification to all the user's resources.

Listing 7: PEP node notification sent to user

```

<message from='user@example.com'
  id='notification_1781477179'
  to='user@example.com/client' type='headline'>
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
    <items node='urn:xmpp:ooo:0'>
      <item id='current'>
        <ooo xmlns='urn:xmpp:ooo:0'>
          <start>2003-07-06T10:30:00+10:00</start>
          <end>2003-07-13T08:00:00+10:00</end>
          <message>I'm attending OSCON in sunny Portland and won't be
            able to
              read your message until I get back. If it's urgent
                , please
                  send_email_to_rob@cataclysm.cx.</message>
            </ooo>
          </item>
        </items>
      </event>
    </message>

```

The meaning of each element is as outlined above. All elements are required. Additionally, the <start/> and <end/> elements MAY be empty (i.e., have no CDATA). When <start/> is empty, the client MUST take this to mean that the settings should take effect immediately. Similarly, when <end/> is empty, the client MUST take this to mean that the settings should never expire (unless they are explicitly removed).

4.3 Removing out-of-office settings

A user may remove all stored vacation settings by sending an IQ to the local server like so:

Listing 8: Remove vacation settings

```
<iq type='set'
  from='user@example.com/client'
  to='example.com'
  it='retract1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <retract node='urn:xmpp:ooo:0'>
      <item id='current' />
    </retract>
  </pubsub>
</iq>
```

Listing 9: Vacation settings removed successfully

```
<iq type='result'
  from='example.com'
  to='user@example.com/client'
  id='retract1' />
```

TODO: Is the Delete And Notify functionality described in XEP-0060 7.2.2.1 widely implemented? If so, should that case be included here?

4.4 Hiding vacation settings

In PEP, a node's default access model is 'presence' (i.e., any other user with a subscription type "from" or "both" may subscribe to the node. PEP provides other access models including a 'whitelist.' See [Publish-Subscribe \(XEP-0060\)](#)⁵ section 4.5 for a list of Node Access Models. The whitelist access model can be used to hide the current out-of-office message. With an empty whitelist, no other users receive notification when the node changes. The client may also use this fact to allow the client's user to set up an out-of-office message before it becomes active.

See [Publish-Subscribe \(XEP-0060\)](#)⁶ Section 8.2 for details on configuration the out-of-office

⁵XEP-0060: Publish-Subscribe <<https://xmpp.org/extensions/xep-0060.html>>.

⁶XEP-0060: Publish-Subscribe <<https://xmpp.org/extensions/xep-0060.html>>.

node's access model.

See also [Best Practices for Persistent Storage of Private Data via Publish-Subscribe \(XEP-0223\)](#)⁷.

PEP and PubSub allow for publishing an item and setting configuration on a node. A user may wish to create an initial, hidden out-of-office setting. They may do so by publishing an item with id='archive' and an access model of 'whitelist.' When the user enables the out-of-office settings, a configuration changing the access model to some other mode sends notifications as defined by PEP.

TODO: add an example of publishing an item and setting configuration with a single stanza i.e., find the appropriate example from [Publish-Subscribe \(XEP-0060\)](#)⁸ or [Personal Eventing Protocol \(XEP-0163\)](#)⁹ and include it here.

5 Server requirements

A server implementing out-of-office messages MUST implement [Personal Eventing Protocol \(XEP-0163\)](#)¹⁰.

6 Security Considerations

None yet defined.

7 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)¹¹.

⁷XEP-0223: Best Practices for Persistent Storage of Private Data via Publish-Subscribe <<https://xmpp.org/extensions/xep-0223.html>>.

⁸XEP-0060: Publish-Subscribe <<https://xmpp.org/extensions/xep-0060.html>>.

⁹XEP-0163: Personal Eventing Protocol <<https://xmpp.org/extensions/xep-0163.html>>.

¹⁰XEP-0163: Personal Eventing Protocol <<https://xmpp.org/extensions/xep-0163.html>>.

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/>>.

8 XMPP Registrar Considerations

The 'urn:xmpp:ooo:0' namespace shall be registered with the [XMPP Registrar](#)¹² as a result of this document.

9 XML Schema

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

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

  <xs:element name='item'>
    <xs:complexType>
      <xs:sequence>
        <xs:element name='start' minOccurs='0' maxOccurs='1' type='
          xs:string' />
        <xs:element name='end' minOccurs='0' maxOccurs='1' type='
          xs:string' />
        <xs:element name='message' minOccurs='0' maxOccurs='1' type='
          xs:string' />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

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 <https://xmpp.org/registrar/>.