



# XMPP

## XEP-0297: Message Forwarding

Matthew Wild

<mailto:me@matthewwild.co.uk>

<xmpp:me@matthewwild.co.uk>

Kevin Smith

<mailto:kevin@kismith.co.uk>

<xmpp:kevin@doomsong.co.uk>

2011-07-11

Version 0.3

Status	Type	Short Name
Proposed	Standards Track	forwarding

This document defines a protocol to forward a message from one entity to another.

# Legal

## Copyright

This XMPP Extension Protocol is copyright © 1999 - 2012 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 <http://xmpp.org/about-xmpp/xsf/xsf-ipr-policy/> or obtained by writing to XMPP Standards Foundation, 1899 Wynkoop Street, Suite 600, Denver, CO 80202 USA).

## Contents

1	Introduction	1
2	Requirements	1
3	Forwarding a message	1
3.1	Overview . . . . .	1
3.2	Business rules . . . . .	2
4	Security Considerations	2
4.1	As-is . . . . .	3
4.2	Extensions . . . . .	3
5	XML Schema	3
6	Acknowledgements	4

## 1 Introduction

There are many situations in which an entity needs to forward a previously sent message to another entity, such as forwarding an interesting message to a friend, or a server forwarding stored messages from an archive. Here we specify a simple encapsulation method for such forwarded messages. This format can be used in other specifications requiring the forwarding of messages, or used in isolation for a user to forward a message to another user (rather like email forwards).

## 2 Requirements

Several properties are desirable when forwarding messages:

- The original sender and receiver should be identified.
- Extension payloads should be included (not only a message <body/>
- It should be possible to annotate the message (e.g. with a timestamp) without ambiguity as to the original message.
- It should allow features such as encryption or signing of the original message to survive the forwarding.

## 3 Forwarding a message

### 3.1 Overview

Let us suppose that a Romeo receives a message from Juliet:

Listing 1: Receiving a message

```
<message to='romeo@montague.lit' from='juliet@capulet.lit/orchard'  
  type='chat' id='0202197'>  
  <body>Yet I should kill thee with much cherishing.</body>  
  <mood xmlns='http://jabber.org/protocol/mood'>  
    <amorous/>  
  </mood>  
</message>
```

To forward this to Mercutio, Romeo would send a new message with a <forwarded/> payload of namespace 'urn:xmpp:forward:0'.

Listing 2: Forwarding a message

```
<message to='mercutio@verona.lit' from='romeo@montague.lit/orchard'
  type='chat' id='28gs'>
  <body>A most courteous exposition!</body>
  <forwarded xmlns='urn:xmpp:forward:0'>
    <delay xmlns='urn:xmpp:delay' stamp='2010-07-10T23:08:25Z' />
    <message to='romeo@montague.lit' from='juliet@capulet.lit/
      orchard' type='chat' id='0202197' xmlns='jabber:client'>
      <body>Yet I should kill thee with much cherishing.</body
        >
      <mood xmlns='http://jabber.org/protocol/mood'>
        <amorous />
      </mood>
    </message>
  </forwarded>
</message>
```

### 3.2 Business rules

- Forwarded messages SHOULD include all relevant child elements of the original message by default. However, an implementation MAY omit elements it deems irrelevant and safe to discard. An example would be omitting [Chat State Notifications](#)<sup>1</sup> elements which typically do not make sense outside the context of a conversation session. However it should be noted that removing such elements can invalidate any digital signature on a stanza. If preserving a signature is important in the context this extension is used then child elements SHOULD NOT be removed.
- The forwarding entity SHOULD add a <delay/> child to the <forwarded/> element to indicate to the recipient the date/time that the forwarding entity received the original message. The format of this element is described in [Delayed Delivery](#)<sup>2</sup>.
- The namespace of the forwarded stanza MUST be preserved (this is typically 'jabber:client'). If no 'xmlns' is set for the stanza then as per XML namespacing rules it would inherit the 'urn:xmpp:forward:0' namespace, which is wrong.
- When this extension is employed simply for a user to forward a given message to a contact, the outer <message/> SHOULD contain a body (even if empty) and a receiving client should pay particular attention to ensure it renders both the sender's text and the forwarded message unambiguously.

## 4 Security Considerations

Forwarding messages can reveal information about the original sender, including possible presence leaks as well as the message payloads themselves. Any extensions using this format

<sup>1</sup>XEP-0085: Chat State Notifications <<http://xmpp.org/extensions/xep-0085.html>>.

<sup>2</sup>XEP-0203: Delayed Delivery <<http://xmpp.org/extensions/xep-0203.html>>.

must therefore consider the implications of this.

Forwarding can either be used as-is, or in the context of another specification, with different security considerations:

#### 4.1 As-is

Receipt of a forwarded message from a third-party does not guarantee that the original message was actually received, or that the content has not been modified, by the forwarder. Integrity of the original message can only be determined through digital signing mechanisms such as described in [Encapsulating Digital Signatures in XMPP](#)<sup>3</sup> and [Encapsulated Digital Signatures in XMPP](#)<sup>4</sup>.

Considering the above an end-user client should take special care in its rendering of a forwarded message to ensure that the user cannot mistake it for a message received directly from the original sender.

An entity SHOULD NOT trust that forwards are genuine when receiving them unprovoked (i.e. outside the scope of another specification).

#### 4.2 Extensions

While security considerations are ultimately dependent upon the specifications using the format defined herein, forwarding introduces scope for message forgery such that authors of derivative specifications will need to address security considerations themselves. These need to cover which entities a client should accept forwards from, and which entities those are permitted to forward messages for. For example, a specification may choose to only trust forwards if they are received from the user's client, another client on the bare JID, or the user's server.

## 5 XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>
<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='urn:xmpp:forward:0'
  xmlns='urn:xmpp:forward:0'
  elementFormDefault='qualified'>
  <xs:annotation>
    <xs:documentation>
```

---

<sup>3</sup>XEP-0285: Encapsulating Digital Signatures in XMPP <<http://xmpp.org/extensions/xep-0285.html>>.

<sup>4</sup>XEP-0290: Encapsulated Digital Signatures in XMPP <<http://xmpp.org/extensions/xep-0290.html>>.

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

<xs:element name='forwarded'>
  <xs:complexType>
    <xs:sequence xmlns:delay='urn:xmpp:delay'>
      <xs:element ref='delay:delay' minOccurs='0' maxOccurs='1' />
      <xs:any namespace='##other' minOccurs='1' maxOccurs='1' />
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

## 6 Acknowledgements

Thanks to Matt Miller and Florian Zeitz for feedback.