



XMPP

XEP-0280: Message Carbons

Joe Hildebrand

<mailto:jhildebr@cisco.com>

<xmpp:jhildebr@cisco.com>

Matthew Miller

<mailto:linuxwolf@outer-planes.net>

<xmpp:linuxwolf@outer-planes.net>

2017-02-16

Version 0.12.0

Status	Type	Short Name
Proposed	Standards Track	carbons

In order to keep all IM clients for a user engaged in a conversation, outbound messages are carbon-copied to all interested resources.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2018 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	Discovering Support	1
4	Enabling Carbons	2
4.1	Recommended Error Conditions	3
5	Disabling Carbons	3
5.1	Recommended Error Conditions	4
6	Messages Eligible for Carbons Delivery	4
7	Receiving Messages	5
8	Sending Messages	6
9	Avoiding Carbons for a single message	8
10	Business Rules	9
10.1	Handling Multiple Enable/Disable Requests	9
10.2	Interaction with Chat States	9
10.3	Handling of Errors	10
10.4	Auto-responses	10
10.5	Mobile Considerations	10
11	Security Considerations	10
12	IANA Considerations	11
13	XMPP Registrar Considerations	11
13.1	Protocol Namespaces	11
13.2	Protocol Versioning	11
14	XML Schema	12
15	Acknowledgements	12

1 Introduction

At the time of original writing of this XEP, many XMPP servers handle message stanzas sent to a user@host (or "bare") JID with no resource by delivering that message only to the resource with the highest priority for the target user. Some server implementations, however, have chosen to send these messages to all of the online resources for the target user. If the target user is online with multiple resources when the original message is sent, a conversation ensues on one of the user's devices; if the user subsequently switches devices, parts of the conversation may end up on the alternate device, causing the user to be confused, misled, or annoyed.

This XEP defines an approach for ensuring that all of my devices get both sides of all conversations in order to avoid user confusion. As a pleasant side-effect, information about the current state of a conversation is shared between all of a user's clients that implement this protocol.

2 Requirements

- Large changes SHOULD NOT be required to existing servers
- Clients that do not implement the new protocol MUST be able participate in conversations
- Clients that do not implement the new protocol MUST NOT receive a large number of new partial conversations
- Clients that do not implement the new protocol MUST NOT receive protocol they do not expect
- All clients that turn on the new protocol MUST be able to see all inbound instant messaging messages.
- All clients that turn on the new protocol MUST be able to see all outbound instant messaging messages from all of the resources of the user, regardless of whether the clients for the other resources have implemented the new protocol.

3 Discovering Support

An entity advertises support for this protocol by including the "urn:xmpp:carbons:2" feature in its service discovery information features as specified in [Service Discovery \(XEP-0030\)](#)¹ or section 6.3 of [Entity Capabilities \(XEP-0115\)](#)².

¹XEP-0030: Service Discovery <<https://xmpp.org/extensions/xep-0030.html>>.

²XEP-0115: Entity Capabilities <<https://xmpp.org/extensions/xep-0115.html>>.

Listing 1: Client requests information about its own server

```
<iq xmlns='jabber:client'
  from='romeo@montague.example/garden'
  id='info1'
  to='montague.example'
  type='get'>
  <query xmlns='http://jabber.org/protocol/disco#info' />
</iq>
```

Listing 2: Server responds with Carbons feature

```
<iq xmlns='jabber:client'
  from='montague.example'
  id='info1'
  to='romeo@montague.example/garden'
  type='result'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    ...
    <feature var='urn:xmpp:carbons:2' />
    ...
  </query>
</iq>
```

4 Enabling Carbons

When a client wants to participate in the Carbons protocol, it enables the protocol by sending an IQ-set containing a child element `<enable/>` qualified by the namespace `"urn:xmpp:carbons:2"`:

Listing 3: Client enables Carbons

```
<iq xmlns='jabber:client'
  from='romeo@montague.example/garden'
  id='enable1'
  type='set'>
  <enable xmlns='urn:xmpp:carbons:2' />
</iq>
```

The server will respond with an IQ-result when Carbons are enabled:

Listing 4: Server acknowledges enabling Carbons

```
<iq xmlns='jabber:client'
  from='romeo@montague.example'
  id='enable1'
  to='romeo@montague.example/garden'
  type='result' />
```

If the server cannot enable Carbons for this client, it sends an IQ-error to the client, with an appropriate error condition (e.g., `<forbidden/>` if local policy forbids the client from enabling):

Listing 5: Server forbids client from enabling Carbons

```
<iq xmlns='jabber:client'
  from='romeo@montague.example'
  id='enable1'
  to='romeo@montague.example/garden'
  type='error'>
  <error type='auth'>
    <forbidden xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>
```

4.1 Recommended Error Conditions

There are various reasons why a server might not be able to enable Carbons for a client. The RECOMMENDED error conditions to return for some reasons are:

- `<forbidden/>` if the server's policy forbids the client from enabling Carbons.
- `<not-allowed/>` if the request is from a client that is not hosted on this server.

See the section [Handling Multiple Enable/Disable Requests](#) for considerations when a client attempts to enable Carbons multiple times.

5 Disabling Carbons

Some clients might want to disable Carbons. To disable Carbons, the client sends an IQ-set containing a child element `<disable/>` qualified by the namespace "urn:xmpp:carbons:2":

Listing 6: Client disables Carbons

```
<iq xmlns='jabber:client'
  from='romeo@montague.example/garden'
  id='disable1'
  type='set'>
  <disable xmlns='urn:xmpp:carbons:2' />
</iq>
```

The server will respond with an IQ-result when Carbons are disabled:

Listing 7: Server acknowledges disabling Carbons

```
<iq xmlns='jabber:client'
  from='romeo@montague.example'
  id='disable1'
  to='romeo@montague.example/garden'
  type='result' />
```

If the server cannot disable Carbons for this client, it sends an IQ-error to the client, with an appropriate error condition (e.g., <not-allowed/> if trying to disable another client's Carbons):

Listing 8: Server informs client cannot disable Carbons

```
<iq xmlns='jabber:client'
  from='romeo@montague.example'
  id='disable1'
  to='juliet@capulet.example/balcony'
  type='error'>
  <error type='cancel'>
    <not-allowed xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>
```

5.1 Recommended Error Conditions

There are various reasons why a server might not be able to disable Carbons for a client. The RECOMMENDED error conditions to return for some reasons are:

- <not-allowed/> if the request is from a client that is not hosted on this server.

See the section [Handling Multiple Enable/Disable Requests](#) for considerations when a client attempts to disable Carbons multiple times.

6 Messages Eligible for Carbons Delivery

The focus of this specification is instant messaging applications and so those (and only those) <message/> stanzas used for instant messaging SHOULD be delivered as Carbons.

Defining precisely which messages are used for instant messaging and which are not is difficult, as future specifications may add additional payloads used for, or not used for, instant messaging; as such, the rules for which messages are eligible for carbons delivery is left as an implementation detail for servers.

The following is a suggested set of rules a server MAY use, or it MAY use its own; in either case it SHOULD follow the general intent of these rules:

- A `<message/>` is eligible for carbons delivery if it is of type "chat".
- A `<message/>` is eligible for carbons delivery if it is of type "normal" and it contains a `<body>` element.
- A `<message/>` is eligible for carbons delivery if it is of type "error" and sent in response to a `<message/>` that was itself eligible for carbons delivery (Note that as this would require message tracking and correlation on the server, it is unlikely to be generally appropriate for most implementations).
- A `<message/>` is not eligible for carbons delivery if it is determined to have been sent by a MUC room or service, even if it would be otherwise eligible (this also includes private messages from MUC participants).
- A `<message/>` is not eligible for carbons delivery if it does not meet any of these criteria.

As this is a implementation detail of servers, clients MUST NOT rely on the server implementing a particular set of rules for which messages are eligible for Carbons delivery.

Future specifications may have more precise requirements on which messages need to be eligible for carbons delivery; such future specifications will provide their own discovery and negotiation mechanisms, such that a client negotiating Carbons using the protocol defined in this specification will cause the server to consider messages eligible for Carbons delivery based on the requirements described herein.

Note: previous versions of this specification limited eligible messages to those of type "chat" - however, this was generally found to be inadequate due to the proliferation of type "normal" messages used in instant messaging.

7 Receiving Messages

When the server receives a `<message/>` [eligible for carbons delivery](#) addressed to a client JID (either bare or full), it delivers the `<message/>` according to RFC 6121 § 8.5.3, and then delivers a forwarded copy to each Carbons-enabled resource for the matching bare JID recipient that did not receive it under the RFC 6121 delivery rules.

Each forwarded copy is wrapped using [Stanza Forwarding \(XEP-0297\)](#)³ with the following properties:

- The wrapping message SHOULD maintain the same 'type' attribute value;
- the 'from' attribute MUST be the Carbons-enabled user's bare JID (e.g., "local-part@domainpart");
- and the 'to' attribute MUST be the full JID of the resource receiving the copy.

³XEP-0297: Stanza Forwarding <https://xmpp.org/extensions/xep-0297.html>.

- The content of the wrapping message MUST contain a <received/> element qualified by the namespace "urn:xmpp:carbons:2", which itself contains a <forwarded/> element qualified by the namespace "urn:xmpp:forward:0" that contains the original <message/>.

The receiving server MUST NOT send a forwarded copy to the client(s) the original <message/> stanza was addressed to, as these recipients receive the original <message/> stanza.

Listing 9: Juliet sends Romeo a directed message

```
<message xmlns='jabber:client'
  from='juliet@capulet.example/balcony'
  to='romeo@montague.example/garden'
  type='chat'>
  <body>What man art thou that, thus bescreen'd_in_night,_so_stumblest
    _on_my_counsel?</body>
  <<thread>0e3141cd80894871a68e6fe6b1ec56fa</thread>
</message>
```

Listing 10: Server sends carbon to Romeo's other clients

```
<message xmlns='jabber:client'
  from='romeo@montague.example'
  to='romeo@montague.example/home'
  type='chat'>
  <received xmlns='urn:xmpp:carbons:2'>
    <forwarded xmlns='urn:xmpp:forward:0'>
      <message xmlns='jabber:client'
        from='juliet@capulet.example/balcony'
        to='romeo@montague.example/garden'
        type='chat'>
        <body>What man art thou that, thus bescreen'd_in_night,_so_
          stumblest_on_my_counsel?</body>
        <<thread>0e3141cd80894871a68e6fe6b1ec56fa</thread>
      </message>
    </forwarded>
  </received>
</message>
```

8 Sending Messages

When a client sends a <message/> [eligible for carbons delivery](#), its sending server delivers the <message/> according to RFC 6120 and RFC 6121, and delivers a forwarded copy to each Carbons-enabled resource for the matching bare JID sender, excluding the sending client. Note that this happens irrespective of whether the sending client has carbons enabled. Each forwarded copy is wrapped using [Stanza Forwarding \(XEP-0297\)](#)⁴ with the following

⁴XEP-0297: Stanza Forwarding <<https://xmpp.org/extensions/xep-0297.html>>.

properties:

- The wrapping message SHOULD maintain the same 'type' attribute value;
- the 'from' attribute MUST be the Carbons-enabled user's bare JID (e.g., "local-part@domainpart");
- and the 'to' attribute SHOULD be the full JID of the resource receiving the copy.
- The content of the wrapping message MUST contain a <sent/> element qualified by the namespace "urn:xmpp:carbons:2", which itself contains a <forwarded/> qualified by the namespace "urn:xmpp:forward:0" that contains the original <message/> stanza.

The sending server SHOULD NOT send a forwarded copy to the sending full JID if it is a Carbons-enabled resource.

Listing 11: Romeo responds to Juliet

```
<message xmlns='jabber:client'
  from='romeo@montague.example/home'
  to='juliet@capulet.example/balcony'
  type='chat'>
  <body>Neither, fair saint, if either thee dislike.</body>
  <thread>0e3141cd80894871a68e6fe6b1ec56fa</thread>
</message>
```

Listing 12: Romeo's other Carbons-enabled clients receive a copy

```
<message xmlns='jabber:client'
  from='romeo@montague.example'
  to='romeo@montague.example/garden'
  type='chat'>
  <sent xmlns='urn:xmpp:carbons:2'>
    <forwarded xmlns='urn:xmpp:forward:0'>
      <message xmlns='jabber:client'
        to='juliet@capulet.example/balcony'
        from='romeo@montague.example/home'
        type='chat'>
        <body>Neither, fair saint, if either thee dislike.</body>
        <thread>0e3141cd80894871a68e6fe6b1ec56fa</thread>
      </message>
    </forwarded>
  </sent>
</message>
```

9 Avoiding Carbons for a single message

Some clients might want to avoid Carbons on a single message, while still keeping all of the other semantics of Carbon support. This might be useful for clients sending end-to-end encrypted messages.

- The sending client MAY exclude a <message/> from being forwarded to other Carbons-enabled resources, by adding a <private/> element qualified by the namespace "urn:xmpp:carbons:2" and a <no-copy/> hint as described in [Message Processing Hints \(XEP-0334\)](#)⁵ as child elements of the <message/> stanza.
- The sending server MUST NOT deliver forwarded <message/>s to the other Carbons-enabled resources of the sender.
- The receiving server MUST NOT deliver forwarded <message/>s to the other Carbons-enabled resource of the recipient,
- and the receiving server SHOULD remove the <private/> element before delivering to the recipient.

Note: Use of the private mechanism might lead to partial conversations on other devices. This is the intended effect. If the private <message/> stanza is addressed to a bare JID, the receiving server still delivers it according to RFC 6121. This might result in a copy being delivered to each resource for the recipient, which effectively negates the behavior of the <private/> element for recipients.

Listing 13: Romeo sends to Juliet, excluding Carbons

```
<message xmlns='jabber:client'
  from='romeo@montague.example/home'
  to='juliet@capulet.example/home'
  type='chat'>
  <body>Neither, fair saint, if either thee dislike.</body>
  <thread>0e3141cd80894871a68e6fe6b1ec56fa</thread>
  <private xmlns='urn:xmpp:carbons:2' />
  <no-copy xmlns='urn:xmpp:hints' />
</message>
```

Listing 14: Romeo's server delivers original message, without creating Carbon copies

```
<message xmlns='jabber:client'
  from='romeo@montague.example/home'
  to='juliet@capulet.example/home'
  type='chat'>
  <body>Neither, fair saint, if either thee dislike.</body>
```

⁵XEP-0334: Message Processing Hints <<https://xmpp.org/extensions/xep-0334.html>>.

```
<thread>0e3141cd80894871a68e6fe6b1ec56fa</thread>
<private xmlns='urn:xmpp:carbons:2' />
<no-copy xmlns='urn:xmpp:hints' />
</message>
```

10 Business Rules

10.1 Handling Multiple Enable/Disable Requests

Handling multiple enable/disable request must adhere to the following rules:

- If a client is permitted to enable Carbons during its login session, the server **MUST** allow the client to enable and disable the protocol multiple times within a session.
- The server **SHOULD NOT** treat multiple enable requests (without an intermediate disable request) as an error;
- the server **SHOULD** simply return an IQ-result (if the protocol is already enabled) or an IQ-error (if the client is not permitted to enable Carbons) for any subsequent requests after the first.
- Similarly, the server **SHOULD NOT** treat multiple disable requests (without an intermediate enable request) as an error;
- the server **SHOULD** return an IQ-result (if the protocols is already disabled) or an IQ-error (if the client's request failed previously) for any subsequent requests after the first.

10.2 Interaction with Chat States

Note: [Chat State Notifications \(XEP-0085\)](#) ⁶ recommends sending chat state notifications as chat type messages, which means that they will be subject to Carbon-copying. This is intentional.

Additionally, there are other considerations for clients that implement Carbons and XEP-0085:

- Upon receiving an inbound or outbound `<gone/>` chat state (as a carbon copy) for a given conversation, the client **SHOULD** visually indicate the conversation is terminated.
- In order to prevent unwanted termination of conversations on other resources, clients **SHOULD NOT** send `<gone/>` chat states on logout, instead
- clients **SHOULD** count on the broadcast of unavailable presence to convey the change in attention.

⁶XEP-0085: Chat State Notifications <https://xmpp.org/extensions/xep-0085.html>.

- Upon receiving an outbound notification of any chat state other than <gone/>, the copied client MAY conclude that the sending client has taken responsibility for the conversation, and make appropriate user interface modifications. For example, notifications could be suppressed on devices receiving the Carbon copies.

10.3 Handling of Errors

The following rules prevent some of the half-failure modes that have been an issue in other protocols:

- When a server attempts to deliver a (locally generated) carbon copy, and that carbon copy bounces with an error for any reason, the server MUST NOT forward that error back to the original sender.
- The server MUST NOT rely on the <sent/> or <received/> elements in the bounce to determine that an error is from a carbon-copied message, because entities are not required to include the original XML in their error replies as per RFC 6120, §8.3.1.

10.4 Auto-responses

Clients that automatically respond to messages for any reason (e.g., when in the "dnd" presence show state) MUST take adequate care when enabling Carbons in order to prevent storms or loops.

Forwarded outbound messages MUST NOT be auto-replied to under any circumstances.

Forwarded inbound messages MUST NOT be auto-replied to unless the client has some way of ensuring no more than one auto-reply is sent from all of its user's resources.

10.5 Mobile Considerations

Enabling this protocol on mobile devices needs to be undertaken with care. This protocol can result in additional bandwidth and power usage, possibly decreasing battery lifetime and increasing monetary costs. Additional mechanisms for controlling the Carbon-copying of individual conversations might need to be added to deal with mobile clients in the future.

11 Security Considerations

The security model assumed by this document is that all of the resources for a single user are in the same trust boundary.

- Any forwarded copies received by a Carbons-enabled client MUST be from that user's bare JID;

- any copies that do not meet this requirement MUST be ignored.

Outbound chat messages that are encrypted end-to-end are not often useful to receive on other resources. As such, they should use the `<private/>` element specified above to avoid such copying, unless the encryption mechanism is able to accommodate this protocol.

12 IANA Considerations

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

13 XMPP Registrar Considerations

13.1 Protocol Namespaces

This specification defines the following XML namespace:

- `urn:xmpp:carbons:2`

Upon advancement of this specification from a status of Experimental to a status of Draft, the [XMPP Registrar](#)⁸ 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 \(XEP-0053\)](#)⁹.

13.2 Protocol 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.

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/>](http://www.iana.org/).

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/>](https://xmpp.org/registrar/).

⁹XEP-0053: XMPP Registrar Function [<https://xmpp.org/extensions/xep-0053.html>](https://xmpp.org/extensions/xep-0053.html).

14 XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>
<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='urn:xmpp:carbons:2'
  xmlns='urn:xmpp:carbons:2'
  elementFormDefault='qualified'>

  <xs:element name='disable' type='empty' />

  <xs:element name='enable' type='empty' />

  <xs:element name='private' type='empty' />

  <xs:element name='received' type='forward-container' />

  <xs:element name='sent' type='forward-container' />

  <xs:simpleType name='empty' abstract='true'>
    <xs:restriction base='xs:string'>
      <xs:enumeration value='' />
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name='forward-container' abstract='true'>
    <xs:choice>
      <xs:element name='forwarded'
        namespace='urn:xmpp:forward:0'
        minOccurs='1'
        maxOccurs='1' />
    </xs:choice>
  </xs:complexType>

</xs:schema>
```

15 Acknowledgements

The authors wish to thank Patrick Barry, Teh Chang, Jack Erwin, Craig Kaes, Kathleen McMurry, Tory Patnoe, Peter Saint-Andre, Ben Schumacher, and Kevin Smith for their feedback.