



XMPP

XEP-0231: Bits of Binary

Peter Saint-Andre
<mailto:stpeter@stpeter.im>
<xmpp:stpeter@jabber.org>
<https://stpeter.im/>

Pavel Šimerda
<xmpp:pavlix@pavlix.net>
<http://www.pavlix.net/>

2022-07-25
Version 1.1

Status	Type	Short Name
Draft	Standards Track	bob

This specification defines an XMPP protocol extension for including or referring to small bits of binary data in an XML stanza.

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	Protocol	1
2.1	Data Exchange	1
2.2	Referencing Data	2
2.3	Retrieving Uncached Data	2
2.4	Caching Data	3
2.5	Format of the <data/> Element	4
2.6	Algorithm Names	6
3	Determining Support	6
4	Security Considerations	7
5	IANA Considerations	7
6	XMPP Registrar Considerations	7
6.1	Protocol Namespaces	7
7	XML Schema	7
8	Acknowledgements	8

1 Introduction

Sometimes it is desirable to include a small bit of binary data in an XMPP stanza. Typical use cases might be to include icon or emoticon in a message, a thumbnail in a file transfer request, a rasterized image in a whiteboarding session, or a small bit of media in a data form. Currently, there is no lightweight method for including such data in an XMPP stanza, since existing methods (e.g., [In-Band Bytestreams \(XEP-0047\)](https://xmpp.org/extensions/xep-0047.html)¹) are designed for larger blobs of data and therefore require some form of negotiation (e.g., via [SI File Transfer \(XEP-0096\)](https://xmpp.org/extensions/xep-0096.html)² or [Jingle File Transfer \(XEP-0234\)](https://xmpp.org/extensions/xep-0234.html)³).

This document specifies just such a lightweight method. The key building blocks are:

1. A Content-ID ("cid") that uniquely identifies the data.
2. A `<data/>` element (similar to the data: URL scheme defined in [RFC 2397](http://tools.ietf.org/html/rfc2397)⁴) that enables the sender and recipient to exchange the data identified by the cid.

2 Protocol

2.1 Data Exchange

The RECOMMENDED approach is for the sender to include the cid when communicating with the recipient. The recipient SHOULD then check its cache of data to determine if the data identified by that cid is cached. If the data is cached, the recipient would then load its cached data. If the data is not cached, the recipient would then retrieve the data by sending an IQ-get to the sender (or potentially some other entity) containing an empty `<data/>` element whose 'cid' attribute specifies the data to be retrieved, to which the sender would reply with an IQ-result containing a `<data/>` element whose XML character data provides the binary data. The `<data/>` element MUST be used only to encapsulate small bits of binary data and MUST NOT be used for large data transfers. Naturally the definitions of "small" and "large" are rather loose. In general, the data SHOULD NOT be more than 8 kilobytes, and dedicated file transfer methods (e.g., [SOCKS5 Bytestreams \(XEP-0065\)](https://xmpp.org/extensions/xep-0065.html)⁵ or [In-Band Bytestreams \(XEP-0047\)](https://xmpp.org/extensions/xep-0047.html)⁶) SHOULD be used for exchanging blobs of data larger than 8 kilobytes. However, implementations or deployments MAY impose their own limits.

If the data to be shared is particularly small (e.g., less than 1k), then the sender MAY send it directly by including a `<data/>` element directly in a `<message/>`, `<presence/>`, or `<iq/>` stanza. The following rules apply:

¹XEP-0047: In-Band Bytestreams <<https://xmpp.org/extensions/xep-0047.html>>.

²XEP-0096: SI File Transfer <<https://xmpp.org/extensions/xep-0096.html>>.

³XEP-0234: Jingle File Transfer <<https://xmpp.org/extensions/xep-0234.html>>.

⁴RFC 2397: The data: URL scheme <<http://tools.ietf.org/html/rfc2397>>.

⁵XEP-0065: SOCKS5 Bytestreams <<https://xmpp.org/extensions/xep-0065.html>>.

⁶XEP-0047: In-Band Bytestreams <<https://xmpp.org/extensions/xep-0047.html>>.

Listing 2: Requesting data

```

<iq from='doctor@shakespeare.lit/pda'
  id='get-data-1'
  to='lady-macbeth@shakespeare.lit/castle'
  type='get'>
  <data xmlns='urn:xmpp:bob'
    cid='sha1+8f35fef110ffc5df08d579a50083ff9308fb6242@bob.xmpp.
      org' />
</iq>

```

The recipient then would either return an error (e.g., <item-not-found/> if it does not have data matching the Content-ID) or return the data.

Listing 3: Returning data

```

<iq from='lady-macbeth@shakespeare.lit/castle'
  id='get-data-1'
  to='doctor@shakespeare.lit/pda'
  type='result'>
  <data xmlns='urn:xmpp:bob'
    cid='sha1+8f35fef110ffc5df08d579a50083ff9308fb6242@bob.xmpp.
      org'
    max-age='86400'
    type='image/png'>
iVBORw0KGgoAAAANSUgAAAAoAAAAKCAyAAACNMs+9AAAABGdBTUEAALGP
C/xhBQAAAA1wSF1zAAALEwAACxMBAJqcGAAAAAd0SU1FB9YGARc5KB0XV+IA
AAAddEVYdENvbW1lbnQAQ3JlYXRlZCB3aXRoIFRoZSBHSU1Q72Q1bgAAAF1J
REFUGN09zL0Ng1AAxPEfdLTs4BZM4DI04C70wQg2JoQ9LE1exdlYvBBBeZ7jq
ch9//q1uH4TLzw4d6+ErXMMcXuHwXId3K0ETnnXXV6MJpcq2MLaI97CER3N0
vr4MkhoXe0rZigAAAABJRU5ErkJggg==
  </data>
</iq>

```

2.4 Caching Data

It is RECOMMENDED for the recipient to cache data; however, the recipient MAY opt not to cache data, for example because it runs on a device that does not have sufficient space for data storage.

The default behavior is for the recipient to cache the data only for the life of the entity's application session (not a client's presence session with the server or the controlling user's communication session with the contact from whom the user received the data); that is, the recipient would clear the cache when the application is terminated or restarted.

As a hint regarding the suggested period for caching the data, the sender MAY include a 'max-age' attribute whenever it sends a <data/> element. The meaning of the 'max-age' attribute exactly matches that of the Max-Age attribute from RFC 2965⁹.

⁹RFC 2965: HTTP State Management Mechanism <<http://tools.ietf.org/html/rfc2965>>.

If it is not suggested to cache the data (e.g., because it is ephemeral), the value of the 'max-age' attribute MUST be "0" (the number zero).

A recipient SHOULD cache data based on the hash of the data as encapsulated in the cid. However, if a hash cannot be extracted from the cid, if the recipient does not support the hashing algorithm used, or the recipient does not support hashes, then the recipient SHOULD cache based on the JID of the sender.

2.5 Format of the <data/> Element

To exchange binary data, the data is encapsulated as the XML character data of a <data/> element qualified by the 'urn:xmpp:bob' namespace, where the data MUST be encoded as Base64 in accordance with Section 4 of RFC 4648¹⁰ (note: the Base64 output MUST NOT include whitespace and MUST set the number of pad bits to zero).

The following attributes are defined for the <data/> element.

Attribute	Description	Inclusion
cid	A Content-ID that can be mapped to a cid: URL as specified in RFC 2111 RFC 2111: Content-ID and Message-ID Uniform Resource Locators < http://tools.ietf.org/html/rfc2111 >.. The 'cid' value SHOULD be of the form algo+hash@bob.xmpp.org, where the "algo" is the hashing algorithm used (e.g., "sha1" for the SHA-1 algorithm as specified in RFC 3174 RFC 3174: US Secure Hash Algorithm 1 (SHA1) < http://tools.ietf.org/html/rfc3174 >., for more information see the next section) and the "hash" is the hex output of the algorithm applied to the binary data itself.	REQUIRED
max-age	A suggestion regarding how long (in seconds) to cache the data; the meaning matches the Max-Age attribute from RFC 2965 RFC 2965: HTTP State Management Mechanism < http://tools.ietf.org/html/rfc2965 >..	RECOMMENDED

¹⁰RFC 4648: The Base16, Base32, and Base64 Data Encodings <<http://tools.ietf.org/html/rfc4648>>.

Attribute	Description	Inclusion
type	The value of the 'type' attribute MUST match the syntax specified in RFC 2045 RFC 2045: Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies < http://tools.ietf.org/html/rfc2045 >.. That is, the value MUST include a top-level media type, the "/" character, and a subtype; in addition, it MAY include one or more optional parameters (e.g., the "audio/ogg" MIME type in the example shown below includes a "codecs" parameter as specified in RFC 4281 RFC 4281: The Codecs Parameter for "Bucket" Media Types < http://tools.ietf.org/html/rfc4281 >.). The "type/subtype" string SHOULD be registered in the IANA MIME Media Types Registry IANA registry of MIME media types < http://www.iana.org/assignments/media-types >., but MAY be an unregistered or yet-to-be-registered value.	REQUIRED if the <data/> element is non-empty

The following example illustrates the format (line endings are provided for readability only).

Listing 4: Data element format

```
<data xmlns='urn:xmpp:bob'
  cid='sha1+8f35fef110ffc5df08d579a50083ff9308fb6242@bob.xmpp.org'
  max-age='86400'
  type='image/png'>
iVBORw0KGgoAAAANSUgAAAAoAAAAKCAyAAACNM+s+9AAAABGdBTUEAALGP
C/xhBQAAAA1wSFlzAAALEwAACxMBAJqcGAAAAAd0SU1FB9YGARc5KB0XV+IA
AAAddEVYdENvbW1lbnQAQ3JlYXRlZCB3aXRoIFRoZSBHSU1Q72Q1bgAAAF1J
REFUGN09zL0Ng1AAxPEfdLTs4BZM4DI04C70wQg2JoQ9LE1exd1YvBBEz7jq
ch9//q1uH4TLzw4d6+ErXMMcXuHWxId3KOETnnXXV6MJpcq2MLaI97CER3N0
vr4MkhoXe0rZigAAAABJRU5ErkJggg==
</data>
```

2.6 Algorithm Names

The value of the 'algo' parameter MUST be one of the values from the [IANA Hash Function Textual Names Registry](#) ¹¹ maintained by the [Internet Assigned Numbers Authority \(IANA\)](#) ¹², or one of the values defined in [Use of Cryptographic Hash Functions in XMPP \(XEP-0300\)](#) ¹³ unless the hash is SHA-1 in which case the label "sha1" MUST be used for historical reasons.

3 Determining Support

If an entity supports the protocol specified herein, it MUST advertise that fact by returning a feature of "urn:xmpp:bob" in response to [Service Discovery \(XEP-0030\)](#) ¹⁴ information requests.

Listing 5: Service discovery information request

```
<iq from='doctor@shakespeare.lit/pda'
  id='disco1'
  to='ladymacbeth@shakespeare.lit/castle'
  type='get'>
  <query xmlns='http://jabber.org/protocol/disco#info' />
</iq>
```

Listing 6: Service discovery information response

```
<iq from='ladymacbeth@shakespeare.lit/castle'
  id='disco1'
  to='doctor@shakespeare.lit/pda'
  type='result'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    ...
    <feature var='urn:xmpp:bob' />
    ...
  </query>
</iq>
```

In order for an application to determine whether an entity supports this protocol, where possible it SHOULD use the dynamic, presence-based profile of service discovery defined in [Entity Capabilities \(XEP-0115\)](#) ¹⁵. However, if an application has not received entity

¹¹IANA registry of Hash Function Textual Names <<http://www.iana.org/assignments/hash-function-text-names>>.

¹²The 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/>>.

¹³XEP-0300: Use of Cryptographic Hash Functions in XMPP <<https://xmpp.org/extensions/xep-0300.html>>.

¹⁴XEP-0030: Service Discovery <<https://xmpp.org/extensions/xep-0030.html>>.

¹⁵XEP-0115: Entity Capabilities <<https://xmpp.org/extensions/xep-0115.html>>.

capabilities information from an entity, it SHOULD use explicit service discovery instead.

4 Security Considerations

The ability to include arbitrary binary data implies that it is possible to send scripts, applets, images, and executable code, which may be potentially harmful. To reduce the risk of such exposure, an implementation MAY choose to not display or process such data but instead either completely ignore the data, show only the value of the 'alt' attribute, or prompt a human user for approval (either explicitly via user action or implicitly via a list of approved entities from whom the user will accept binary data without per-event approval).

5 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#) ¹⁶.

6 XMPP Registrar Considerations

6.1 Protocol Namespaces

The [XMPP Registrar](#) ¹⁷ includes "urn:xmpp:bob" in its registry of protocol namespaces (see <https://xmpp.org/registrar/namespaces.html>).

7 XML Schema

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

¹⁶The 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/>.

¹⁷The 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/>.

```
<xs:annotation>
  <xs:documentation>
    The protocol documented by this schema is defined in
    XEP-0231: http://www.xmpp.org/extensions/xep-0231.html
  </xs:documentation>
</xs:annotation>

<xs:element name='data'>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base='xs:base64Binary'>
        <xs:attribute name='cid' type='xs:string' use='required' />
        <xs:attribute name='max-age' type='xs:nonNegativeInteger'
          use='optional' />
        <xs:attribute name='type' type='xs:string' use='optional' />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

</xs:schema>
```

8 Acknowledgements

Thanks to Rachel Blackman, Dave Cridland, Zenon Kuder, and Tomasz Sterna for their feedback.