



# Legal

## Copyright

This XMPP Extension Protocol is copyright © 1999 - 2011 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	Protocol	1
3	Examples	2
4	Internationalization Considerations	14
5	Security Considerations	14
6	IANA Considerations	14
7	XMPP Registrar Considerations	15
8	XML Schema	15
9	Acknowledgements	15

## 1 Introduction

Everyone knows that binary data encodings are much more efficient than the Extensible Markup Language (XML). Given that XMPP is an application profile of XML, it is a wonder that XMPP works at all! Clearly, XMPP would benefit from a binary representation. Therefore, this specification defines Binary XMPP.

## 2 Protocol

[XMPP Core](#)<sup>1</sup> defines the traditional representation of XMPP. For instance, an empty presence stanza (such as might be provided on login) is as follows:

Listing 1: Traditional XMPP stanza

```
<presence/>
```

That string can be represented in binary as follows:

Listing 2: Binary representation

```
001111100011110000001111001001100101011110011011001010110111001100001101100101001001011100111
```

The bit sequence is therefore represented in Binary XMPP as follows (line breaks are provided only for the purpose of readability):

Listing 3: Binary XMPP representation

```
<zero/><zero/><one/><one/><one/><one/><zero/><zero/>
<zero/><one/><one/><one/><zero/><zero/><zero/><zero/>
<zero/><one/><one/><one/><zero/><zero/><one/><zero/>
<zero/><one/><one/><zero/><zero/><one/><zero/><one/>
<zero/><one/><one/><one/><zero/><zero/><one/><one/>
<zero/><one/><one/><zero/><zero/><one/><zero/><one/>
<zero/><one/><one/><zero/><one/><one/><one/><zero/>
<zero/><one/><one/><zero/><zero/><zero/><one/><one/>
<zero/><one/><one/><zero/><zero/><one/><zero/><one/>
<zero/><zero/><one/><zero/><one/><one/><one/><one/>
<zero/><zero/><one/><one/><one/><one/><one/><zero/>
```

This is an obvious improvement over the traditional representation. It might be objected that the Binary XMPP representation requires a large number of XML elements. However, because all data is represented using only `<zero/>` and `<one/>`, the protocol can be significantly compressed using the standard ZLIB compression algorithm as

<sup>1</sup>RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <<http://tools.ietf.org/html/rfc6120>>.

defined in [RFC 1950](#)<sup>2</sup>, which SHOULD be enabled by default on the bxmpp port. Evidence from preliminary testing indicates that compression of Binary XMPP results in a representation less than 2% of the original size!

As a further simplification, Binary XMPP gets rid of the complicated stream negotiation process defined in RFC 3920. Instead, if a server supports Binary XMPP it advertises a special DNS SRV record `_bxmpp._tcp` in accordance with [RFC 2782](#)<sup>3</sup>. Furthermore, only one record is needed, thus saving precious port space. The RECOMMENDED port for Binary XMPP communication is 10110, but any port MAY be used.

To start a Binary XMPP stream, the sender simply connects via TCP at the advertised port and starts sending `<zero/>` and `<one/>` elements. The lack of a stream header simplifies XML parsing, introducing further performance improvements.

### 3 Examples

The following examples showcase the tremendous efficiency of Binary XMPP using a complete session between a client and a server.

Listing 4: Client sends initial stream header (XMPP)

```
<stream:stream
  from='juliet@example.com'
  to='example.com'
  version='1.0'
  xml:lang='en'
  xmlns='jabber:client'
  xmlns:stream='http://etherx.jabber.org/streams'>
```

Listing 5: Binary XMPP representation

```
<zero/><zero/><one/><one/><one/><one/><zero/><zero/><zero/><one/><one/>
<one/><zero/><zero/><one/><one/><zero/><one/><one/><one/><zero/><zero/><one/>
<zero/><zero/><one/><one/><zero/><zero/><one/><zero/><one/><zero/><zero/>
<one/><one/><zero/><zero/><zero/><zero/><one/><zero/><one/><one/><zero/><one/>
<zero/><one/><one/><zero/><one/><zero/><zero/><one/><one/><one/><zero/><zero/><one/>
<one/><zero/><one/><one/><one/><zero/><one/><zero/><zero/><zero/><one/><one/><zero/>
<zero/><zero/><one/><zero/><one/><zero/><one/><one/><zero/><zero/><zero/><one/>
<one/><one/><one/><zero/><zero/><one/><zero/><zero/><one/><one/><zero/><zero/>
<zero/><zero/><one/><zero/><one/><zero/><one/><one/><zero/><zero/><zero/><zero/>
<zero/><zero/><one/><zero/><one/><one/><zero/><one/><one/><zero/><one/><zero/>
<zero/><zero/><zero/><one/><zero/><one/><zero/><zero/><zero/><one/>
<zero/><zero/><zero/><zero/><zero/><zero/><zero/><one/><zero/><zero/><zero/>
```

<sup>2</sup>RFC 1950: ZLIB Compressed Data Format Specification version 3.3 <<http://tools.ietf.org/html/rfc1950>>.

<sup>3</sup>RFC 2782: A DNS RR for specifying the location of services (DNS SRV) <<http://tools.ietf.org/html/rfc2782>>.























```

zero/><one/><one/><one/><zero/><one/><zero/><zero/><one/><one/><
zero/><one/><one/><one/><zero/><zero/><one/><one/><one/><zero/><
zero/><one/><one/><zero/><zero/><one/><one/><one/><zero/><one/><
zero/><zero/><one/><one/><one/><one/><zero/><zero/><zero/><zero/><
one/><one/><zero/><one/><one/><zero/><one/><zero/><one/><one/><one
/><zero/><zero/><zero/><zero/><zero/><one/><one/><one/><zero/><
zero/><zero/><zero/><zero/><zero/><one/><zero/><one/><one/><zero/>
<one/><zero/><one/><one/><one/><zero/><one/><zero/><zero/><zero/><
one/><one/><zero/><one/><one/><zero/><zero/><zero/><one/><one/><
one/><zero/><zero/><one/><one/><zero/><zero/><one/><zero/><zero/><
one/><one/><one/><zero/><zero/><one/><zero/><one/><one/><one/><one
/><zero/><zero/><one/><one/><one/><one/><one/><zero/>

```

And so on. Much more efficient, eh?

## 4 Internationalization Considerations

The `<zero/>` and `<one/>` elements use English-language words as the element names. Clearly it would have been preferable to define an i18n-friendly binding, such that German-language applications could encode Binary XMPP using the `<null/>` and `<eins/>` elements, Greek-language applications could use the `<μηδέν/>` and `<ένα/>` elements, etc. Flexibility regarding internationalization of the element names may be added in Binary XMPP 2.0.

## 5 Security Considerations

Because it is simply an alternative representation of XMPP, Binary XMPP inherits all of the security characteristics of XMPP (the good, the bad, and the ugly).

Use of port 10110 is obviously secure, since 10110 in base 2 is 22 in base 10, the same default port as Secure Shell (see [RFC 4251](#)<sup>4</sup>).

## 6 IANA Considerations

The [XMPP Registrar](#)<sup>5</sup> shall work with the [Internet Assigned Numbers Authority \(IANA\)](#)<sup>6</sup> to register "bxmpp" as a keyword for port 10111.

<sup>4</sup>RFC 4251: The Secure Shell (SSH) Protocol Architecture <<http://tools.ietf.org/html/rfc4251>>.

<sup>5</sup>The 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 <<http://xmpp.org/registrar/>>.

<sup>6</sup>The 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/>>.

## 7 XMPP Registrar Considerations

The XMPP Registrar shall include 'urn:xmpp:bxmpp' in its registry of protocol namespaces (see <<http://xmpp.org/registrar/namespaces.html>>).

## 8 XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>
<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='urn:xmpp:bxmpp'
  xmlns='urn:xmpp:bxmpp'
  elementFormDefault='qualified'>
  <xs:element name='one' type='empty' />
  <xs:element name='zero' type='empty' />
  <xs:simpleType name='empty'>
    <xs:restriction base='xs:string'>
      <xs:enumeration value='' />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

## 9 Acknowledgements

Thanks to Kevin Smith for his feedback.