Interoperability Report for the Extensible Messaging and Presence Protocol (XMPP)
draft-saintandre-xmpp-interop-report-00

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on April 16, 2007.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This document provides an interoperability report regarding the Extensible Messaging and Presence Protocol (XMPP), as that technology is specified in RFCs 3920 and 3921 (and their proposed successors). This report specifies both a protocol feature set and a protocol implementation report, in accordance with the concepts and formats proposed by Larry Masinter within the NEWTRK Working Group.
Table of Contents

1. Introduction ...................................................... 3
2. Feature Set ...................................................... 4
  2.1. Addresses .................................................... 4
    2.1.1. Domain Identifier ........................................ 4
    2.1.2. Node Identifier .......................................... 4
    2.1.3. Resource Identifier ...................................... 5
  2.2. XML Streams .................................................. 5
    2.2.1. TCP Binding ............................................... 5
    2.2.2. Namespaces ............................................... 5
    2.2.3. Attributes ............................................... 6
    2.2.4. Stream Features ......................................... 7
    2.2.5. Stream Errors ........................................... 7
  2.3. TLS Negotiation ............................................... 7
  2.4. SASL Negotiation ............................................. 7
  2.5. Mandatory TLS and SASL Technologies ......................... 8
  2.6. Resource Binding ............................................. 8
  2.7. Server Dialback ............................................... 9
  2.8. XML Usage .................................................... 9
  2.9. XML Stanzas ................................................ 9
    2.9.1. Attributes ............................................... 9
    2.9.2. Message Stanzas ......................................... 10
    2.9.3. Presence Stanzas ....................................... 11
    2.9.4. IQ Stanzas .............................................. 11
    2.9.5. Stanza Errors .......................................... 11
    2.9.6. Extended Namespaces .................................... 12
    2.9.7. Stanza Handling ........................................ 12
  2.10. Rosters .................................................... 13
3. Implementation Reports ........................................ 13
4. Security Considerations ......................................... 13
5. Informative References .......................................... 13
Author’s Address .................................................. 14
Intellectual Property and Copyright Statements .................. 15
1. Introduction

The Extensible Messaging and Presence Protocol (XMPP) is an Extensible Markup Language technology for near-real-time messaging, presence, and request-response services. The core XML streaming technology is specified in [RFC3920] and the features needed to implement basic instant messaging and presence applications are specified in [RFC3921]. The basic syntax and semantics of XMPP were developed originally within the Jabber open-source community, mainly in 1999. In November 2002, the XMPP WG was chartered with developing an adaptation of the core Jabber protocol that would be suitable as an IETF instant messaging (IM) and presence technology. In October 2004, the IETF published the XMPP RFCs. Since then, existing server, client, and library implementations (which previously used "XMPP 0.9") have been upgraded to conform to XMPP 1.0 as specified in RFC 3920 and RFC 3921, and new implementations also have been developed. Many of these implementations have been widely deployed within organizations and by service providers (there are at this time estimated to be well over 50,000 server deployments and perhaps 40 to 50 million end users). Therefore, the Internet community has quite a bit of implementation and deployment experience with XMPP. Furthermore, in July 2006, the first in-person interoperability testing event was held, and both online interoperability testing processes and future in-person testing events are planned as well.

Work has begun on updating the XMPP specifications (see [rfc3920bis] and [rfc3921bis]). The proposed changes are based on updates to several of the technologies upon which XMPP depends, interoperability and deployment experience, and formal interoperability testing. Where appropriate, this interoperability report discusses the relevant feature as specified in RFC 3920 or RFC 3921, experience and testing results related to that feature, and modifications to the feature as specified in rfc3920bis or rfc3921bis.

This interoperability report attempts to adhere to the concepts and formats proposed by Larry Masinter within the IETF’s NEWTRK Working Group in 2005 (see [INTEROP]). Therefore this document contains two main sections:

- Protocol Feature Set -- this section describes the set of specifications and the features defined therein that constitute the Extensible Messaging and Presence Protocol for the purpose of interoperability testing.
- Protocol Implementation Reports -- this section describes the results of implementation and deployment experience and interoperability testing for each feature, with one report for each tested codebase.
Although the core XML streaming layer specified in RFC 3920 is not necessarily tied to the instant messaging and presence semantics specified in RFC 3921, this interoperability report treats them as a single protocol.

2. Feature Set

For the purpose of interoperability testing, the Extensible Messaging and Presence Protocol is taken to be defined by [RFC3920] and [RFC3921], where appropriate as updated by [rfc3920bis] and [rfc3921bis].

The following subsections provide a first attempt at specifying the particular features of XMPP. Because XMPP uses a client-server architecture, each feature is labelled as applying to the client role, the server role, or both. In addition, each feature is labelled as REQUIRED, RECOMMENDED, or OPTIONAL, where those terms are to be understood as described in [RFC2119].

2.1. Addresses

XMPP addresses are of the form [node@]domain[/resource]. Rules for each portion of the address are specified in Section 3 of RFC 3920. The following features apply.

2.1.1. Domain Identifier

As specified in Section 3.2 of RFC 3920:

1. The domain identifier portion of an XMPP address must conform to the Nameprep profile of Stringprep. Conformance with this feature is REQUIRED for servers and RECOMMENDED for clients.
2. The domain identifier portion of an XMPP address must not be more than 1023 bytes in length. Conformance with this feature is REQUIRED for clients and for servers.

2.1.2. Node Identifier

As specified in Section 3.3 of RFC 3920:

1. The node identifier portion of an XMPP address must conform to the Nodeprep profile of Stringprep. Conformance with this feature is REQUIRED for servers and RECOMMENDED for clients.
2. The node identifier portion of an XMPP address must not be more than 1023 bytes in length. Conformance with this feature is REQUIRED for clients and for servers.
2.1.3. Resource Identifier

As specified in Section 3.4 of RFC 3920:

1. The resource identifier portion of an XMPP address must conform to the Resourceprep profile of Stringprep. Conformance with this feature is REQUIRED for servers and RECOMMENDED for clients.
2. The resource identifier portion of an XMPP address must not be more than 1023 bytes in length. Conformance with this feature is REQUIRED for clients and for servers.

2.2. XML Streams

At root, XMPP is a technology for streaming XML data between a client and a server or between two servers. Thus the management of XML streams is a core aspect of XMPP. The following features apply.

2.2.1. TCP Binding

As specified in Section 4.2 of RFC 3920:

1. XML streams are communicated over a TCP connection. Conformance with this feature is REQUIRED for clients and for servers.

2.2.2. Namespaces

As specified in Section 4.5 and Section 11.2.2 of RFC 3920:

1. An XML stream must be qualified by a streams namespace of 'http://etherx.jabber.org/streams'. Conformance with this feature is REQUIRED for clients and for servers.
2. All elements within the streams namespace must be prefixed with a namespace prefix. Conformance with this feature is REQUIRED for clients and for servers.
3. The streams namespace prefix should be "stream:". Conformance with this feature is RECOMMENDED for clients and for servers.
4. An XML stream must have a default namespace other than the streams namespace. Conformance with this feature is REQUIRED for clients and for servers.
5. An implementation must support ‘jabber:client’ as a default namespace. Conformance with this feature is REQUIRED for clients and for servers.
6. An implementation must support ‘jabber:server’ as a default namespace. Conformance with this feature is REQUIRED for servers (the feature does not apply to clients).
2.2.3. Attributes

As specified in Section 4.4 and Section 4.4.1 of RFC 3920:

1. An initiating entity should include a ‘from’ attribute in the initial stream header it sends to a receiving entity. Conformance with this feature is RECOMMENDED for clients and for servers. (Note: This feature was modified in rfc3920bis as compared to RFC 3920, since implementation and deployment experience has shown that including the ‘from’ attribute makes stream establishment more efficient.)

2. A receiving entity must include a ‘from’ attribute in the response stream header it sends to an initiating entity. Conformance with this feature is REQUIRED for servers (the feature does not apply to clients).

3. An initiating entity should include a ‘to’ attribute in the initial stream header it sends to a receiving entity. Conformance with this feature is RECOMMENDED for clients and for servers.

4. A receiving entity should include a ‘to’ attribute in the response stream header it sends to an initiating entity. Conformance with this feature is RECOMMENDED for servers (the feature does not apply to clients). (Note: This feature was modified in rfc3920bis as compared to RFC 3920, since implementation and deployment experience has shown that including the ‘to’ attribute makes stream establishment more efficient.)

5. A receiving entity must include an ‘id’ attribute in the header for the response stream it sends to an initiating entity. Conformance with this feature is REQUIRED for servers (the feature does not apply to clients).

6. The value of the ‘id’ attribute must be unique within the receiving entity. Conformance with this feature is REQUIRED for servers (the feature does not apply to clients).

7. An initiating entity should include an ‘xml:lang’ attribute in the initial stream headers that it generates. Conformance with this feature is RECOMMENDED for clients and for servers.

8. An initiating entity must include a ‘version’ attribute whose value is "1.0" (for XMPP 1.0 support) in the initial stream headers it generates. Conformance with this feature is REQUIRED for clients and for servers.

9. If the stream header that a receiving entity receives from an initiating entity includes a ‘version’ attribute whose value is "1.0", the receiving entity must include a ‘version’ attribute whose value is "1.0" in the response stream headers it generates. Conformance with this feature is REQUIRED for servers.
2.2.4. Stream Features

As specified in Section 4.6 of RFC 3920:

1. A receiving entity shall advertise the stream-related features it supports after sending a response stream header. Conformance with this feature is REQUIRED for servers (the feature does not apply to clients).

2.2.5. Stream Errors

As specified in Section 4.7 of RFC 3920:

1. An entity shall generate a stream error (followed by a closing stream tag and termination of the TCP connection) when it detects a stream-related error condition. Conformance with this feature is REQUIRED for clients and for servers.

2. The syntax for stream errors shall follow the definition in Section 4.7.2 of RFC 3920. Conformance with this feature is REQUIRED for clients and for servers.

2.3. TLS Negotiation

As specified in Section 5 of RFC 3920:

1. An implementation must support Transport Layer Security (TLS) for channel encryption of XML streams. Conformance with this feature is REQUIRED for clients and for servers.

2. TLS negotiation between two servers must not proceed until the DNS hostnames are resolved. Conformance with this feature is REQUIRED for servers (the feature does not apply to clients).

3. There must be no whitespace between XML elements sent during TLS negotiation. Conformance with this feature is REQUIRED for clients and for servers.

4. Certificate validation must follow the rules in Section 14.2 of RFC 3920. Conformance with this feature is REQUIRED for clients and for servers.

5. Upon successful TLS negotiation, the initiating entity must send a new initial stream header to the receiving entity. Conformance with this feature is REQUIRED for clients and for servers.

2.4. SASL Negotiation

As specified in Section 6 of RFC 3920:

1. An implementation must support the Simple Authentication and Security Layer (SASL) for authentication of XML streams. Conformance with this feature is REQUIRED for clients and for
servers.

2. SASL negotiation between two servers must not proceed until the
DNS hostnames are resolved. Conformance with this feature is
REQUIRED for servers (the feature does not apply to clients).

3. There must be no whitespace between XML elements sent during SASL
negotiation. Conformance with this feature is REQUIRED for
clients and for servers.

4. Upon successful SASL negotiation, the initiating entity must send
a new initial stream header to the receiving entity. Conformance
with this feature is REQUIRED for clients and for servers.

5. An implementation must support the SAL error conditions specified
in Section 6.4 of RFC 3920. Conformance with this feature is
REQUIRED for clients and for servers.

2.5. Mandatory TLS and SASL Technologies

As specified in Section 14.7 of RFC 3920:

1. An implementation must support the SASL DIGEST-MD5 mechanism.
Conformance with this feature is REQUIRED for clients and for
servers.

2. An implementation must support the TLS_RSA_WITH_3DES_EDE_CBC_SHA
cipher. Conformance with this feature is REQUIRED for clients
and for servers.

3. An implementation must support TLS plus SASL PLAIN. Conformance
with this feature is REQUIRED for clients and for servers.
(Note: This feature was added in rfc3920bis as compared to RFC
3920, since implementation of SASL EXTERNAL is uncommon in XMPP
clients, in part because underlying security features such as
X.509 certificates are not yet widely deployed.)

4. An implementation must support TLS plus SASL EXTERNAL for server-
to-server connections. Conformance with this feature is REQUIRED
for servers.

2.6. Resource Binding

As specified in Section 7 of RFC 3920:

1. An implementation must support resource binding for client-to-
server connections. Conformance with this feature is REQUIRED
for clients and for servers.

2. An implementation must be able to request generation of a
resource (rather than providing it). Conformance with this
feature is RECOMMENDED for clients (the feature does not apply to
servers).

3. An implementation must be able to generate a resource on request.
Conformance with this feature is REQUIRED for servers (the
feature does not apply to clients).
4. An implementation should be able to bind multiple resources to an XML stream as specified in Section 7.1 of rfc3920bis. Conformance with this feature is RECOMMENDED for servers and OPTIONAL for clients.

2.7. Server Dialback

As specified in Section 8 of RFC 3920:

1. An implementation should support server dialback for server-to-server connections. Conformance with this feature is RECOMMENDED for servers (the feature does not apply to clients).
2. An implementation should use the HMAC-256 algorithm to generate dialback keys, as specified in Appendix C.4 of rfc3920bis. Conformance with this feature is RECOMMENDED for servers (the feature does not apply to clients).

2.8. XML Usage

1. As specified in Section 11 of RFC 3920, an implementation must not inject XML comments, processing instructions, internal or external DTD subsets, internal or external entity references other than the predefined XML entities, or XML character data or attribute values containing unescaped characters that map to the predefined entities. Conformance with this feature is REQUIRED for clients and for servers.
2. As specified in Section 11.1 of rfc3920bis, an implementation must return a <restricted-xml/> stream error if it receives XML comments, processing instructions, internal or external DTD subsets, internal or external entity references other than the predefined XML entities, or XML character data or attribute values containing unescaped characters that map to the predefined entities. Conformance with this feature is REQUIRED for clients and for servers. (Note: This feature was modified in rfc3920bis as compared to RFC 3920, since ignoring such data rather than returning an error is inconsistent with the stream error handling recommendations in Section 4.7 of RFC 3920.)

2.9. XML Stanzas

2.9.1. Attributes

As specified in Section 9.1 of RFC 3920:

1. An implementation must handle the <presence/>, <message/>, and <iq/> stanza types. Conformance with this feature is REQUIRED for clients and for servers.
2. An implementation must support the ‘to’ attribute on all stanza types to encapsulate the intended recipient’s address, as specified in Section 9.1.1 of RFC 3920. Conformance with this feature is REQUIRED for clients and servers.

3. An implementation must support the ‘from’ attribute on all stanza types to encapsulate the sender’s address, as specified in Section 9.1.2 of RFC 3920. Conformance with this feature is REQUIRED for clients and servers.

4. In streams qualified by the ‘jabber:client’ namespace, the receiving entity must validate the address of the sender by verifying that it is that of a connected resource for the sending entity or by stamping the ‘from’ value itself, as specified in Section 9.1.2 of RFC 3920. Conformance with this feature is REQUIRED for servers (the feature does not apply to clients).

5. In streams qualified by the ‘jabber:server’ namespace, the sending entity must ensure that every stanza it sends possesses a ‘from’ attribute and that the domain identifier portion of the encapsulated JID value matches a hostname of the server, as specified in Section 9.1.2 of RFC 3920. Conformance with this feature is REQUIRED for servers (the feature does not apply to clients).

6. In streams qualified by the ‘jabber:server’ namespace, the receiving entity must ensure that every stanza it receives possesses a ‘from’ attribute and that the domain identifier portion of the encapsulated JID value matches a hostname of the sending entity, as specified in Section 9.1.2 of RFC 3920. Conformance with this feature is REQUIRED for servers (the feature does not apply to clients).

7. An XML stanza should possess an ‘xml:lang’ attribute, as specified in section 9.1.5 of RFC 3920. Conformance with this feature is RECOMMENDED for clients and for servers.

2.9.2. Message Stanzas

As specified in Section 2.1 of RFC 3921 (Section 5 of rfc3921bis):

1. An implementation must differentiate between messages of type "normal", "chat", "groupchat", "headline", and "error". Conformance with this feature is REQUIRED for clients (the feature does not apply to servers).

2. An implementation must support the <body/> child element of the <message/> stanza. Conformance with this feature is REQUIRED for clients (the feature does not apply to servers).

3. An implementation should support the <subject/> and <thread/> child elements of the <message/> stanza. Conformance with this feature is RECOMMENDED for clients (the feature does not apply to servers).
2.9.3. Presence Stanzas

As specified in Section 6 of RFC 3921:

1. An implementation must support <presence/> stanzas with no 'type' attribute to signal availability and of type "unavailable" to signal lack of availability. Conformance with this feature is REQUIRED for clients and for servers.
2. An implementation must support <presence/> stanzas of type "subscribe", "unsubscribe", "subscribed", and "unsubscribed" to manage presence subscriptions. Conformance with this feature is REQUIRED for clients and for servers.
3. An implementation must support <presence/> stanzas of type "probe" to discover the presence of subscribed entities. Conformance with this feature is REQUIRED for servers (the feature does not apply to clients).

2.9.4. IQ Stanzas

As specified in Section 9.2.3 of RFC 3920:

1. An <iq/> stanza must possess an 'id' attribute. Conformance with this feature is REQUIRED for clients and for servers.
2. The 'type' attribute of an <iq/> stanza must be present and it must have a value of "get", "set", "result", or "error". Conformance with this feature is REQUIRED for clients and for servers.
3. The response to an <iq/> stanza of type "get" or "set" must be an <iq/> stanza of type "result" or "error". Conformance with this feature is REQUIRED for clients and for servers.
4. An <iq/> stanza of type "get" or "set" must contain only one child element. Conformance with this feature is REQUIRED for clients and for servers.
5. An <iq/> stanza of type "result" must contain zero child elements or one child element. Conformance with this feature is REQUIRED for clients and for servers.
6. An <iq/> stanza of type "error" should contain the child element received in the corresponding <iq/> stanza of type "get" or "set". Conformance with this feature is RECOMMENDED for clients and for servers.

2.9.5. Stanza Errors

As specified in Section 9.3 of RFC 3920:

1. A <message/> , <presence/> , or <iq/> stanza whose type is "error" must contain an <error/> child element whose syntax adheres to the definition specified in Section 9.3.2 of RFC 3920.
Conformance with this feature is REQUIRED for clients and for servers.

2. An implementation must not return an error stanza in response to a <message/>, <presence/>, or <iq/> stanza whose type is "error". Conformance with this feature is REQUIRED for clients and for servers.

3. An implementation must support the stanza error conditions specified in Section 9.3.3 of RFC 3920. Conformance with this feature is REQUIRED for clients and for servers.

2.9.6. Extended Namespaces

As specified in Section 2.4 of RFC 3921 (Section 8.4 of rfc3920bis):

1. An implementation must not process XML data qualified by XML namespaces it does not understand. Conformance with this feature is REQUIRED for clients and for servers.

2.9.7. Stanza Handling

As specified in Section 10 of RFC 3920:

1. A server must properly handle an XML stanza with no ‘to’ address, as specified in Section 10.1 of RFC 3920. Conformance with this feature is REQUIRED for servers (the feature does not apply to clients).

2. A server must properly handle an XML stanza with a ‘to’ address whose domain identifier does not match one of the configured hostnames of the server, as specified in Section 10.2 of RFC 3920. Conformance with this feature is REQUIRED for servers (the feature does not apply to clients).

3. A server must properly handle an XML stanza with a ‘to’ address whose domain identifier matches a subdomain of one of the configured hostnames of the server, as specified in Section 10.2 of RFC 3920 and clarified in Section 10.3 of rfc3920bis. Conformance with this feature is REQUIRED for servers (the feature does not apply to clients).

4. A server must properly handle an XML stanza with a ‘to’ address that exactly matches one of the configured hostnames of the server (or such hostname appended by a resource identifier), as specified in Section 10.3 of RFC 3920. Conformance with this feature is REQUIRED for servers (the feature does not apply to clients).

5. A server must properly handle an XML stanza with a ‘to’ address of the form <node@domain> whose domain identifier matches one of the configured hostnames of the server, as specified in Section 10.2 of RFC 3920 and Section 11.1 of RFC 3921. Conformance with this feature is REQUIRED for servers (the feature does not apply to clients).
2.10. Rosters

As specified in Section 7 of RFC 3921:

1. An implementation must use <iq/> stanzas containing <query/> elements qualified by the 'jabber:iq:roster' namespace in order to manage contact lists (in XMPP, "rosters"). Conformance with this feature is REQUIRED for clients and for servers.
2. An implementation should retrieve its roster after sending initial presence. Conformance with this feature is REQUIRED for clients (the feature does not apply to servers).

3. Implementation Reports

A future version of this document will provide one implementation report for each tested or reporting codebase.

4. Security Considerations

This document does not directly discuss security issues, since they are discussed in the relevant sections of RFC 3920, RFC 3921, rfc3920bis, and rfc3921bis.

RFC 3921 requires client implementations to use the technology specified in [RFC3923] for end-to-end signing and object encryption of XML stanzas in the context of instant messaging and presence applications. Unfortunately, there exist no implementations of the protocol specified in RFC 3923, as a result of which no implementation and deployment experience exists and no interoperability testing could be performed.

5. Informative References


[rfc3920bis]


[RFC3921]


[rfc3921bis]


[RFC3923]


Author’s Address

Peter Saint-Andre
Jabber Software Foundation
P.O. Box 1641
Denver, CO  80201
US

Email: stpeter@jabber.org
URI:   xmpp:stpeter@jabber.org