XMPP	P. Saint-Andre
Internet-Draft	Cisco
Obsoletes: 6122 (if approved)	November 17, 2011
Intended status: Standards Track	
Expires: May 20, 2012	

Extensible Messaging and Presence Protocol (XMPP): Address Format
draft-ietf-xmpp-6122bis-00

Abstract

This document defines the address format for the Extensible Messaging and Presence Protocol (XMPP), including support for code points outside the US-ASCII range. This document obsoletes RFC 6122.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

1.1. Overview

The Extensible Messaging and Presence Protocol [XMPP] is an application profile of the Extensible Markup Language [XML] for streaming XML data in close to real time between any two or more network-aware entities.

The address format for XMPP entities was originally developed in the Jabber open-source community in 1999, first described by [XEP-0029] in 2002, and then defined canonically by [RFC3920] in 2004 and [RFC6122] in 2011.

As specified in RFC 3920 and RFC 6122, the XMPP address format used the "stringprep" technology for preparation of non-ASCII characters [STRINGPREP]. This document defines the XMPP address format in a way that no longer depends on stringprep. Instead, this document depends on the internationalization framework defined by the IETF's PRECIS Working Group [FRAMEWORK].

This document obsoletes RFC 6122.

1.2. Terminology

Many important terms used in this document are defined in [FRAMEWORK], [I18N-TERMS], [IDNA-DEFS], [UNICODE], and [XMPP].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [KEYWORDS].

2. Addresses

2.1. Fundamentals

An XMPP entity is anything that is network-addressable and that can communicate using XMPP. For historical reasons, the native address of an XMPP entity is called a Jabber Identifier ("JID"). A valid JID is a string of [UNICODE] code points, encoded using [UTF-8], and structured as an ordered sequence of localpart, domainpart, and resourcepart (where the first two parts are demarcated by the '@' character used as a separator, and the last two parts are similarly demarcated by the '/' character).

The syntax for a JID is defined as follows using the Augmented Backus-Naur Form as specified in [ABNF].

```
   jid           = [ localpart "@" ] domainpart [ "/" resourcepart ]
   localpart     = 1*(localpoint)
                   ;
                   ; a "localpoint" is a UTF-8 encoded Unicode
                   ; code point that conforms to the localpart
                   ; subclass of the "NameClass" string class
                   ; defined in draft-blanchet-precis-framework
                   ;
   domainpart    = IP-literal / IPv4address / ifqdn
                   ;
                   ; the "IPv4address" and "IP-literal" rules are
                   ; defined in RFC 3986, and the first-match-wins
                   ; (a.k.a. "greedy") algorithm described in RFC
                   ; 3986 applies to the matching process
                   ;
                   ; note well that reuse of the IP-literal rule
                   ; from RFC 3986 implies that IPv6 addresses are
                   ; enclosed in square brackets (i.e., beginning
                   ; with '[' and ending with ']')
                   ;
   ifqdn         = 1*(domainpoint)
                   ;
                   ; a "domainpoint" is a UTF-8 encoded Unicode
                   ; code point that conforms to the "domain name"
                   ; string class effectively defined in RFC 5890
                   ;
   resourcepart  = 1*(resourcepoint)
                   ;
                   ; a "resourcepoint" is a UTF-8 encoded Unicode
                   ; code point that conforms to the localpart
                   ; subclass of the "FreeClass" string class
                   ; defined in draft-blanchet-precis-framework
                   ;
```

All JIDs are based on the foregoing structure.

Each allowable portion of a JID (localpart, domainpart, and resourcepart) MUST NOT be zero bytes in length and MUST NOT be more than 1023 bytes in length, resulting in a maximum total size (including the '@' and '/' separators) of 3071 bytes.

For the purposes of communication over an XMPP network (e.g., in the 'to' or 'from' address of an XMPP stanza), an entity's address MUST be represented as a JID, not as a Uniform Resource Identifier [URI] or Internationalized Resource Identifier [IRI]. An XMPP URI or IRI [XMPP-URI] is in essence a JID prepended with 'xmpp:'; however, the native addressing format used in XMPP is that of a mere JID without a URI scheme. [XMPP-URI] is provided only for identification and interaction outside the context of XMPP itself, for example when linking to a JID from a web page. See [XMPP-URI] for information about securely extracting a JID from an XMPP URI or IRI.

- *Implementation Note: When dividing a JID into its component parts, an implementation needs to match the separator characters '@' and '/' before applying any transformation algorithms, which might decompose certain Unicode code points to the separator characters (e.g., U+FE6B SMALL COMMERCIAL AT might decompose to U+0040 COMMERCIAL AT).

2.2. Domainpart

The domainpart of a JID is that portion after the '@' character (if any) and before the '/' character (if any); it is the primary identifier and is the only REQUIRED element of a JID (a mere domainpart is a valid JID). Typically a domainpart identifies the "home" server to which clients connect for XML routing and data management functionality. However, it is not necessary for an XMPP domainpart to identify an entity that provides core XMPP server functionality (e.g., a domainpart can identify an entity such as a multi-user chat service [XEP-0045], a publish-subscribe service [XEP-0060], or a user directory).

The domainpart for every XMPP service MUST be a fully qualified domain name (FQDN; see [DNS]), IPv4 address, IPv6 address, or unqualified hostname (i.e., a text label that is resolvable on a local network).

- *Interoperability Note: Domainparts that are IP addresses might not be accepted by other services for the sake of server-to-server communication, and domainparts that are unqualified hostnames cannot be used on public networks because they are resolvable only on a local network.

If the domainpart includes a final character considered to be a label separator (dot) by [DNS], this character MUST be stripped from the domainpart before the JID of which it is a part is used for the purpose of routing an XML stanza, comparing against another JID, or constructing an [XMPP-URI]. In particular, the character MUST be stripped before any other canonicalization steps are taken.

A domainpart MUST NOT be zero bytes in length and MUST NOT be more than 1023 bytes in length. This rule is to be enforced after any mapping or normalization of code points. Naturally, the length limits of [DNS] apply, and nothing in this document is to be interpreted as overriding those more fundamental limits.

In the terms of IDNA2008 [IDNA-DEFS], the domainpart of a JID is a "domain name slot".

A domainpart consisting of a fully qualified domain name MUST be an "internationalized domain name" as defined in [IDNA-DEFS] and MUST consist only of Unicode code points that conform to the rules specified in [IDNA-CODE].

For the purposes of communication over XMPP, the domainpart of a JID MUST be treated as follows, where the operations specified MUST be completed in the order shown:

1. Uppercase and titlecase characters MUST be mapped to their lowercase equivalents.

2. All characters MUST be mapped using Unicode Normalization Form C (NFC). [[OPEN ISSUE: Use NFD instead?]]

3. Each A-label SHOULD be converted to a U-label (however, if it is not converted then the application MUST apply the Punycode algorithm [PUNYCODE] to each A-label and prepend the ACE prefix ("xn--") to the resulting DNS domain name).

With regard to directionality, the "Bidi Rule" provided in [IDNA-BIDI] applies.

2.3. Localpart

The localpart of a JID is an optional identifier placed before the domainpart and separated from the latter by the '@' character. Typically a localpart uniquely identifies the entity requesting and using network access provided by a server (i.e., a local account), although it can also represent other kinds of entities (e.g., a chat room associated with a multi-user chat service [XEP-0045]). The entity represented by an XMPP localpart is addressed within the context of a specific domain (i.e., <localpart@domainpart>).
A localpart MUST NOT be zero bytes in length and MUST NOT be more than 1023 bytes in length. This rule is to be enforced after any mapping or normalization of code points.
A localpart MUST consist only of Unicode code points that conform to the "NameClass" base string class defined in [FRAMEWORK], with the exception of the following characters that are explicitly disallowed in XMPP localparts:

U+0022 (QUOTATION MARK), i.e., "

U+0026 (AMPERSAND), i.e., &

U+0027 (APOSTROPHE), i.e., '

U+002F (SOLIDUS), i.e., /

U+003A (COLON), i.e., :

U+003C (LESS-THAN SIGN), i.e., <

U+003E (GREATER-THAN SIGN), i.e., >

U+0040 (COMMERCIAL AT), i.e., @

For the purposes of communication over XMPP, the localpart of a JID MUST be treated as follows, where the operations specified MUST be completed in the order shown:

1. Uppercase and titlecase characters MUST be mapped to their lowercase equivalents.

2. All characters MUST be mapped using Unicode Normalization Form C (NFC). [[OPEN ISSUE: Use NFD instead?]]

With regard to directionality, the "Bidi Rule" provided in [IDNA-BIDI] applies.

2.4. Resourcepart

The resourcepart of a JID is an optional identifier placed after the domainpart and separated from the latter by the '/' character. A resourcepart can modify either a <localpart@domainpart> address or a mere <domainpart> address. Typically a resourcepart uniquely identifies a specific connection (e.g., a device or location) or object (e.g., an occupant in a multi-user chat room [XEP-0045]) belonging to the entity associated with an XMPP localpart at a domain (i.e., <localpart@domainpart/resourcepart>).
A resourcepart MUST NOT be zero bytes in length and MUST NOT be more than 1023 bytes in length. This rule is to be enforced after any mapping or normalization of code points.
A resourcepart MUST consist only of Unicode code points that conform to the "FreeClass" base string class defined in [FRAMEWORK].
For the purposes of communication over XMPP, the localpart of a JID MUST be treated as follows, where the operations specified MUST be completed in the order shown:

1. Uppercase and titlecase characters MAY be mapped to their lowercase equivalents.

2. All characters MUST be mapped using Unicode Normalization Form C (NFC). [[OPEN ISSUE: Use NFD instead?]]

With regard to directionality, the "Bidi Rule" provided in [IDNA-BIDI] applies.
XMPP entities SHOULD consider resourceparts to be opaque strings and SHOULD NOT impute meaning to any given resourcepart. In particular:

- *Use of the '/' character as a separator between the domainpart and the resourcepart does not imply that XMPP addresses are hierarchical in the way that, say, HTTP addresses are hierarchical; thus for example an XMPP address of the form <localpart@domainpart/foo/bar> does not identify a resource "bar" that exists below a resource "foo" in a hierarchy of resources associated with the entity "localpart@domainpart".

- *The '@' character is allowed in the resourcepart and is often used in the "nick" shown in XMPP chatrooms [XEP-0045]. For example, the JID <room@chat.example.com/user@host> describes an entity who is an occupant of the room <room@chat.example.com> with an (asserted) nick of <user@host>. However, chatroom services do not necessarily check such an asserted nick against the occupant's real JID.

3. Internationalization Considerations

XMPP applications MUST support IDNA2008 for domainparts, the "NameClass" string class from [FRAMEWORK] for localparts (with the exception of certain ASCII characters specified under Section 2.3), and the "FreeClass" string class from [FRAMEWORK] for resourceparts. This enables XMPP addresses to include a wide variety of characters outside the US-ASCII range. Rules for enforcement of the XMPP address format are provided in [XMPP] and specifications for various XMPP extensions. For backward compatibility, many XMPP applications support [IDNA2003] for domainparts, and the [STRINGPREP] profiles Nodeprep and Resourceprep [RFC3920] for localparts and resourceparts.

4. Security Considerations

4.1. Reuse of PRECIS

The security considerations described in [FRAMEWORK] apply to the "NameClass" and "FreeClass" base string classes used in this document for XMPP localparts and resourceparts. The security considerations described in [IDNA-DEFS] apply to internationalized domain names, which are used here for XMPP domainparts.

4.2. Reuse of Unicode

The security considerations described in [UTR39] apply to the use of Unicode characters in XMPP addresses.

4.3. Address Spoofing

There are two forms of address spoofing: forging and mimicking.

4.3.1. Address Forging

In the context of XMPP technologies, address forging occurs when an entity is able to generate an XML stanza whose 'from' address does not correspond to the account credentials with which the entity authenticated onto the network (or an authorization identity provided during negotiation of SASL authentication [SASL] as described in [XMPP]). For example, address forging occurs if an entity that authenticated as "juliet@im.example.com" is able to send XML stanzas from "nurse@im.example.com" or "romeo@example.net".
Address forging is difficult in XMPP systems, given the requirement for sending servers to stamp 'from' addresses and for receiving servers to verify sending domains via server-to-server authentication (see [XMPP]). However, address forging is possible if:

- *A poorly implemented server ignores the requirement for stamping the 'from' address. This would enable any entity that authenticated with the server to send stanzas from any localpart@domainpart as long as the domainpart matches the sending domain of the server.

- *An actively malicious server generates stanzas on behalf of any registered account at the domain or domains hosted at that server.

Therefore, an entity outside the security perimeter of a particular server cannot reliably distinguish between JIDs of the form <localpart@domainpart> at that server and thus can authenticate only the domainpart of such JIDs with any level of assurance. This specification does not define methods for discovering or counteracting the kind of poorly implemented or rogue servers just described. However, the end-to-end authentication or signing of XMPP stanzas could help to mitigate this risk, since it would require the rogue server to generate false credentials for signing or encryption of each stanza, in addition to modifying 'from' addresses.

Furthermore, it is possible for an attacker to forge JIDs at other domains by means of a DNS poisoning attack if DNS security extensions [DNSSEC] are not used.

4.3.2. Address Mimicking

Address mimicking occurs when an entity provides legitimate authentication credentials for and sends XML stanzas from an account whose JID appears to a human user to be the same as another JID. Because many characters are visually similar, it is relatively easy to mimic JIDs in XMPP systems. As one simple example, the localpart "ju1iet" (using the Arabic numeral one as the third character) might appear the same as the localpart "juliet" (using lowercase "L" as the third character).
As explained in [IDNA-DEFS], [FRAMEWORK], [UTR36], and [UTR39], there is no straightforward solution to the problem of visually similar characters. Furthermore, IDNA and PRECIS technologies do not attempt to define such a solution. As a result, XMPP domainparts, localparts, and resourceparts could contain such characters, leading to security vulnerabilities such as the following:

- A domainpart is always employed as one part of an entity's address in XMPP. One common usage is as the address of a server or server-side service, such as a multi-user chat service [XEP-0045]. The security of such services could be compromised based on different interpretations of the internationalized domainpart; for example, a user might authorize a malicious entity at a fake server to view the user's presence information, or a user could join chatrooms at a fake multi-user chat service.

- A localpart can be employed as one part of an entity's address in XMPP. One common usage is as the username of an instant messaging user; another is as the name of a multi-user chat room; and many other kinds of entities could use localparts as part of their addresses. The security of such services could be compromised based on different interpretations of the internationalized localpart; for example, a user entering a single internationalized localpart could access another user's account information, or a user could gain access to a hidden or otherwise restricted chat room or service.

- A resourcepart can be employed as one part of an entity's address in XMPP. One common usage is as the name for an instant messaging user's connected resource; another is as the nickname of a user in a multi-user chat room; and many other kinds of entities could use resourceparts as part of their addresses. The security of such services could be compromised based on different interpretations of the internationalized resourcepart; for example, two or more confusable resources could be bound at the same time to the same account (resulting in inconsistent authorization decisions in an XMPP application that uses full JIDs), or a user could send a message to someone other than the intended recipient in a multi-user chat room.

XMPP services and clients are strongly encouraged to define and implement consistent policies regarding the registration, storage, and presentation of visually similar characters in XMPP systems. In particular, service providers and software implementers are strongly encouraged to use the policies recommended in [FRAMEWORK].

5. IANA Considerations

5.1. Use of NameClass

The IANA shall add an entry to the PRECIS Usage Registry for reuse of the PRECIS NameClass in XMPP, as follows:

Application Protocol: XMPP.

Base Class: NameClass.

Subclassing: Yes. See Section 2.3 of RFC XXXX.

Directionality: If the string contains at least one right-to-left code point, the entire string is considered to be right-to-left.

Casemapping: Uppercase and titlecase code points are mapped to their lowercase equivalents.

Normalization: NFC.

Specification: RFC XXXX.

5.2. Use of FreeClass

The IANA shall add an entry to the PRECIS Usage Registry for reuse of the PRECIS FreeClass in XMPP, as follows:

Application Protocol: XMPP.

Base Class: FreeClass

Subclassing: No.

Directionality: If the string contains at least one right-to-left code point, the entire string is considered to be right-to-left.

Casemapping:
None.

Normalization: NFC.

Specification: RFC XXXX.

6. Conformance Requirements

This section describes a protocol feature set that summarizes the conformance requirements of this specification. This feature set is appropriate for use in software certification, interoperability testing, and implementation reports. For each feature, this section provides the following information:

- A human-readable name

- An informational description

- A reference to the particular section of this document that normatively defines the feature

- Whether the feature applies to the Client role, the Server role, or both (where "N/A" signifies that the feature is not applicable to the specified role)

- Whether the feature MUST or SHOULD be implemented, where the capitalized terms are to be understood as described in [KEYWORDS]

The feature set specified here attempts to adhere to the concepts and formats proposed by Larry Masinter within the IETF's NEWTRK Working Group in 2005, as captured in [INTEROP]. Although this feature set is more detailed than called for by [REPORTS], it provides a suitable basis for the generation of implementation reports to be submitted in support of advancing this specification from Proposed Standard to Draft Standard in accordance with [PROCESS].

Feature: address-domain-length

Description: Ensure that the domainpart of an XMPP address is at least one byte in length and at most 1023 bytes in length, and conforms to the underlying length limits of the DNS.

Section: Section 2.2

Roles: Both MUST.

Feature:
address-domain-prep

Description: Ensure that the domainpart of an XMPP address conforms to IDNA2008, mapped to lowercase and normalized using NFC.

Section: Section 2.2

Roles: Both MUST.

Feature: address-localpart-length

Description: Ensure that the localpart of an XMPP address is at least one byte in length and at most 1023 bytes in length.

Section: Section 2.3

Roles: Both MUST.

Feature: address-localpart-prep

Description: Ensure that the localpart of an XMPP address conforms to the "NameClass" base string class from the PRECIS framework, excluding the eight XMPP prohibited code points (U+0022, U+0026, U+0027, U+002F, U+003A, U+003C, U+003E, and U+0040), with all code points mapped to lowercase and normalized using NFC.

Section: Section 2.3

Roles: Both MUST.

Feature: address-resource-length

Description: Ensure that the resourcepart of an XMPP address is at least one byte in length and at most 1023 bytes in length.

Section: Section 2.4

Roles: Both MUST.

Feature: address-resource-prep

Description: Ensure that the resourcepart of an XMPP address conforms to the "FreeClass" base string class from the PRECIS framework, with all code points normalized using NFC.

Section: Section 2.4

Roles: Both MUST.

7. References

7.1. Normative References

[ABNF]	Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.
[DNS]	Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.
[FRAMEWORK]	Blanchet, M and P Saint-Andre, "Precis Framework: Handling Internationalized Strings in Protocols", Internet-Draft draft-blanchet-precis-framework-03, August 2011.
[IDNA-BIDI]	Alvestrand, H. and C. Karp, "Right-to-Left Scripts for Internationalized Domain Names for Applications (IDNA)", RFC 5893, August 2010.
[IDNA-CODE]	Faltstrom, P., "The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)", RFC 5892, August 2010.
[IDNA-DEFS]	Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.
[IDNA-PROTO]	Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, August 2010.
[KEYWORDS]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
[UNICODE]	The Unicode Consortium, "The Unicode Standard, Version 3.2.0", 2000. The Unicode Standard, Version 3.2.0 is defined by The Unicode Standard, Version 3.0 (Reading, MA, Addison-Wesley, 2000. ISBN 0-201-61633-5), as amended by the Unicode Standard Annex #27: Unicode 3.1 (http://www.unicode.org/reports/tr27/) and by the Unicode Standard Annex #28: Unicode 3.2 (http://www.unicode.org/reports/tr28/).
[UTF-8]	Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.
[UTR36]	The Unicode Consortium, "Unicode Technical Report #36: Unicode Security Considerations", 2008.
[XMPP]	Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 2011.

7.2. Informative References

[DNSSEC]	Arends, R., Austein, R., Larson, M., Massey, D. and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[I18N-TERMS]	Hoffman, P and J Klensin, "Terminology Used in Internationalization in the IETF", Internet-Draft draft-hoffman-rfc3536bis-02, April 2011.
[IDNA2003]	Faltstrom, P., Hoffman, P. and A. Costello, "Internationalizing Domain Names in Applications (IDNA)", RFC 3490, March 2003. See Section 1 for an explanation of why the normative reference to an obsoleted specification is needed.
[IDNA-RATIONALE]	Klensin, J., "Internationalized Domain Names for Applications (IDNA): Background, Explanation, and Rationale", RFC 5894, August 2010.
[INTEROP]	Masinter, L, "Formalizing IETF Interoperability Reporting", Work in Progress, October 2005.
[IRI]	Duerst, M. and M. Suignard, "Internationalized Resource Identifiers (IRIs)", RFC 3987, January 2005.
[PROCESS]	Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.
[PUNYCODE]	Costello, A., "Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)", RFC 3492, March 2003.
[REPORTS]	Dusseault, L. and R. Sparks, "Guidance on Interoperation and Implementation Reports for Advancement to Draft Standard", BCP 9, RFC 5657, September 2009.
[RFC3920]	Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 3920, October 2004.
[RFC6122]	Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Address Format", RFC 6122, March 2011.
[SASL]	Melnikov, A. and K. Zeilenga, "Simple Authentication and Security Layer (SASL)", RFC 4422, June 2006.
[STRINGPREP]	Hoffman, P. and M. Blanchet, "Preparation of Internationalized Strings ("stringprep")", RFC 3454, December 2002.
[URI]	Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.
[UTR39]	The Unicode Consortium, "Unicode Technical Report #39: Unicode Security Mechanisms", August 2010.
[XEP-0029]	Kaes, C., "Definition of Jabber Identifiers (JIDs)", XSF XEP 0029, October 2003.
[XEP-0045]	Saint-Andre, P., "Multi-User Chat", XSF XEP 0045, July 2008.
[XEP-0060]	Millard, P., Saint-Andre, P. and R. Meijer, "Publish-Subscribe", XSF XEP 0060, July 2010.

[XEP-0165]	Saint-Andre, P., "Best Practices to Discourage JID Mimicking", XSF XEP 0165, December 2007.
[XML]	Paoli, J., Maler, E., Sperberg-McQueen, C., Yergeau, F. and T. Bray, "Extensible Markup Language (XML) 1.0 (Fourth Edition)", World Wide Web Consortium Recommendation REC-xml-20060816, August 2006.
[XMPP-URI]	Saint-Andre, P., "Internationalized Resource Identifiers (IRIs) and Uniform Resource Identifiers (URIs) for the Extensible Messaging and Presence Protocol (XMPP)", RFC 5122, February 2008.

Appendix A. Differences from RFC 6122

Based on consensus derived from implementation and deployment experience as well as formal interoperability testing, the following substantive modifications were made from RFC 3920.

*Changed domainpart preparation to use IDNA2008 instead of IDNA2003.

*Changed localpart preparation to use PRECIS instead of the Nodeprep profile of Stringprep.

*Changed resourcepart preparation to use PRECIS instead of the Resourceprep profile of Stringprep.

Appendix B. Acknowledgements

Some text in this document was borrowed or adapted from [IDNA-DEFS], [IDNA-PROTO], [IDNA-RATIONALE], and [XEP-0165].

Author's Address

Peter Saint-Andre Saint-Andre Cisco 1899 Wynkoop Street, Suite 600 Denver, CO 80202 USA Phone: +1-303-308-3282 EMail: psaintan@cisco.com