XMPP
Internet-Draft
Obsoletes: 6122 (if approved)
Intended status: Standards Track
Expires: March 27, 2013

P. Saint-Andre
Cisco Systems, Inc.

September 23, 2012

# Extensible Messaging and Presence Protocol (XMPP): Address Format
# draft-ietf-xmpp-6122bis-04

## Abstract

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

## Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on March 27, 2013.

## Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction . . . . . 3
1.1. Overview . . . . . 3
1.2. Terminology . . . . . 3
2. Addresses . . . . . 3
2.1. Fundamentals . . . . . 3
2.2. Domainpart . . . . . 5
2.3. Localpart . . . . . 6
2.4. Resourcepart . . . . . 7
3. Enforcement in JIDs and JID Parts . . . . . 9
4. Internationalization Considerations . . . . . 11
5. Security Considerations . . . . . 11
5.1. Reuse of PRECIS . . . . . 11
5.2. Reuse of Unicode . . . . . 11
5.3. Address Spoofing . . . . . 12
5.3.1. Address Forging . . . . . 12
5.3.2. Address Mimicking . . . . . 12
6. IANA Considerations . . . . . 14
6.1. Subclasses . . . . . 14
6.2. Usage of NameClass . . . . . 14
6.3. Usage of FreeClass . . . . . 14
7. Conformance Requirements . . . . . 15
8. References . . . . . 16
8.1. Normative References . . . . . 16
8.2. Informative References . . . . . 17
Appendix A. Differences from RFC 6122 . . . . . 20
Appendix B. Acknowledgements . . . . . 21
Author's Address . . . . . 21

# 1. Introduction

## 1.1. Overview

The Extensible Messaging and Presence Protocol (XMPP) [RFC6120] 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 [RFC3454]. Following the migration of internationalized domain names away from stringprep, this document defines the XMPP address format in a way that no longer depends on stringprep. Instead, this document builds upon 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], [RFC5890], [RFC6120], [RFC6365], and [UNICODE].

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 [RFC2119].

# 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 [RFC3629], 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 (ABNF) as specified in [RFC5234].

```
      jid           = [ localpart "@" ] domainpart [ "/" resourcepart ]
      localpart     = 1*(localpoint)
                      ;
                      ; a "localpoint" is a UTF-8 encoded
                      ; Unicode code point that conforms to
                      ; the "LocalpartNameClass" subclass of
                      ; the "NameClass" string class defined
                      ; in draft-ietf-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
                      ; RFC 5890
                      ;
      resourcepart  = 1*(resourcepoint)
                      ;
                      ; a "resourcepoint" is a UTF-8 encoded
                      ; Unicode code point that conforms to
                      ; the "ResourcepartFreeClass" subclass of
                      ; of the "FreeClass" string class defined
                      ; in draft-ietf-precis-framework
                      ;
```

All JIDs are based on the foregoing structure. However, note that the foregoing structure does not capture all of the rules and restrictions that apply to JIDs, which are described below.

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.

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., under Unicode Normalization Form KC U+FE6B SMALL COMMERCIAL AT decomposes to U+0040 COMMERCIAL AT, although this is not true under Unicode Normalization C, which is used in this specification).

This document defines the native format for JIDs; see [RFC5122] for information about the representation of a JID as a Uniform Resource Identifier (URI) [RFC3986] or Internationalized Resource Identifier (IRI) [RFC3987] and the extraction of a JID from an XMPP URI or IRI.

## 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), an IPv4 address, an IPv6 address, or an unqualified hostname (i.e., a text label that is resolvable on a local network).

Informational Note: The term "fully-qualified domain name" is not well defined. In [RFC1034] it also called an absolute domain name, and the two terms are associated in [RFC1535]. The earliest use of the term can be found in [RFC1123]. References to those older specifications ought not to be construed as limiting the characters of a fully-qualified domain name to the ASCII range; for example, [RFC5890] mentions that a fully-qualified domain name can contain one or more U-labels.

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 [RFC1034], 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 or IRI [RFC5122]. In particular, the character MUST be stripped before any other canonicalization steps are taken.

In general, the content of a domainpart is an Internationalized Domain Name ("IDN") as described in the specifications for Internationalized Domain Names in Applications (commonly called "IDNA2008") [RFC5890], and a domainpart is an "IDNA-aware domain name slot". The following rules apply to a domainpart that consists of a fully-qualified domain name:

- The domainpart MUST contain only NR-LDH labels and U-labels as defined in [RFC5890] and MUST consist only of Unicode code points that conform to the rules specified in [RFC5892].

- The domainpart MUST NOT include A-labels as defined in [RFC5890]; each A-label MUST be converted to a U-label during preparation of a domainpart, and comparison MUST be performed using U-labels, not A-labels.

- After conversion of A-labels to U-labels if necessary, all uppercase and titlecase code points within the domainpart MUST be mapped to their lowercase equivalents.

- After (and in addition to) casemapping, other mappings MAY be applied to the domainpart, such as those defined in [I-D.yoneya-precis-mappings] or [RFC5895].

After any and all normalization, conversion, and mapping of code points, a domainpart MUST NOT be zero bytes in length and MUST NOT be more than 1023 bytes in length. (Naturally, the length limits of [RFC1034] apply, and nothing in this document is to be interpreted as overriding those more fundamental limits.)

## 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 normalization and mapping of code points.

A localpart MUST consist only of Unicode code points that conform to the "LocalpartNameClass" subclass of the "NameClass" base string class defined in [FRAMEWORK]. The LocalpartNameClass subclass includes all code points allowed by the NameClass base class, 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., @

The normalization and mapping rules for the LocalpartNameClass are as follows, where the operations specified MUST be completed in the order shown:

1. All characters MUST be mapped using Unicode Normalization Form C (NFC).

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

3. Fullwidth and halfwidth characters SHOULD be mapped to their decomposition mappings (see [I-D.yoneya-precis-mappings]).

4. Additional mappings MAY be applied, such as those defined in [I-D.yoneya-precis-mappings].

With regard to directionality, applications MUST apply the "Bidi Rule" defined in [RFC5893] (i.e., each of the six conditions of the Bidi Rule must be satisfied).

### 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 normalization and mapping of code points.

A resourcepart MUST consist only of Unicode code points that conform to the "FreeClass" base string class defined in [FRAMEWORK]. (Note that there is no XMPP-specific subclass for resourceparts.)

The normalization and mapping rules for the resourcepart of a JID are as follows, where the operations specified MUST be completed in the order shown:

1. All characters MUST be mapped using Unicode Normalization Form C (NFC).

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

3. Fullwidth and halfwidth characters SHOULD be mapped to their decomposition mappings (see [I-D.yoneya-precis-mappings]).

4. Additional mappings MAY be applied, such as those defined in [I-D.yoneya-precis-mappings].

With regard to directionality, applications MUST apply the "Bidi Rule" defined in [RFC5893] (i.e., each of the six conditions of the Bidi Rule must be satisfied).

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.

In some contexts, it might be appropriate to apply more restrictive rules to the preparation and comparison of XMPP resourceparts. For example, in the context of XMPP Multi-User Chat [XEP-0045], it might be appropriate to apply the rules specified in [I-D.ietf-precis-nickname]. However, the application of such more restrictive rules is out of scope for resourceparts in general and is properly defined in other specifications.

## 3. Enforcement in JIDs and JID Parts

Enforcement of the XMPP address format rules is the responsibility of XMPP servers. Although XMPP clients SHOULD prepare complete JIDs and parts of JIDs in accordance with the rules before including them in protocol slots within XML streams (such that JIDs and parts of JIDs are in conformance), XMPP servers MUST enforce the rules wherever possible and reject stanzas and other XML elements that violate the rules (for stanzas, by returning a <jid-malformed/> error to the sender as described in Section 8.3.3.8 of [RFC6120]).

Enforcement applies to complete JIDs and to parts of JIDs. To facilitate implementation, this document defines the concepts of "JID slot", "localpart slot", and "resourcepart slot" (similar to the concept of a "domain name slot" for IDNA2008 defined in Section 2.3.2.6 of [RFC5890]):

JID Slot: An XML element or attribute explicitly designated in XMPP or in XMPP extensions for carrying a complete JID.

Localpart Slot: An XML element or attribute explicitly designated in XMPP or in XMPP extensions for carrying the localpart of a JID.

Resourcepart Slot: An XML element or attribute explicitly designated in XMPP or in XMPP extensions for carrying the resourcepart of a JID.

A server is responsible for enforcing the address format rules when receiving protocol elements from clients where the server is expected to handle such elements directly or to use them for purposes of routing a stanza to another domain or delivering a stanza to a local entity; two examples from [RFC6120] are the 'to' attribute on XML stanzas (which is a JID slot used by XMPP servers for routing of outbound stanzas) and the <resource/> child of the <bind/> element (which is a resourcepart slot used by XMPP servers for binding of a

resource to an account for routing of stanzas between the server and a particular client).

A server is not responsible for enforcing the rules when the protocol elements are intended for communication among other entities, typically within the payload of a stanza that the server is merely routing to another domain or delivering to a local entity, such as a connected client or an add-on service. Two examples are the 'initiator' attribute in the Jingle extension [XEP-0166] (which is a JID slot used for client-to-client coordination of multimedia sessions) and the 'nick' attribute in the Multi-User Chat extension [XEP-0045] (which is a resourcepart slot used for administrative purposes in the context of XMPP chatrooms). In such cases, clients SHOULD enforce the rules themselves and not depend on the server to do so, and client implementers need to understand that not enforcing the rules can lead to a degraded user experience or to security vulnerabilities. However, when an add-on service (e.g., a multi-user chat service) handles a stanza directly, it ought to enforce the rules as well, as defined by the relevant specification for that type of service.

This document does not provide an exhaustive list of JID slots, localpart slots, or resourcepart slots. However, implementers of core XMPP servers are advised to consider as JID slots at least the following elements and attributes when they are handled directly or used for purposes of routing to another domain or delivering to a local entity:

- The 'from' and 'to' stream attributes and the 'from' and 'to' stanza attributes [RFC6120].
- The 'jid' attribute of the roster <item/> element for contact list management [RFC6121].
- The 'value' attribute of the <item/> element for Privacy Lists [RFC3921] [XEP-0016] when the value of the 'type' attribute is "jid".
- The 'jid' attribute of the <item/> element for Service Discovery defined in [XEP-0030].
- The <value/> element for Data Forms [XEP-0004], when the 'type' attribute is "jid-single" or "jid-multi".
- The 'jid' attribute of the <conference/> element for Bookmark Storage [XEP-0048].
- The <JABBERID/> of the <vCard/> element for vCard 3.0 [XEP-0054] and the <uri/> child of the <impp/> element for vCard 4.0 [XEP-0292] when the XML character data identifies an XMPP URI [RFC5122].
- The 'from' attribute of the <delay/> element for Delayed Delivery [XEP-0203].

- The 'jid' attribute of the <item/> element for the Blocking Command [XEP-0191].
- The 'from' and 'to' attributes of the <result/> and <verify/> elements for Server Dialback [RFC3921], [XEP-0220].
- The 'from' and 'to' attributes of the <amp/> element for Advanced Message Processing [XEP-0079].
- The 'from' and 'to' attributes of the <iq/>, <message/>, and <presence/> elements for the Jabber Component Protocol [XEP-0114].

Developers of XMPP clients and specialized XMPP add-on services are advised to check the appropriate specifications for JID slots, localpart slots, and resourcepart slots in XMPP protocol extensions such as Service Discovery [XEP-0030], Multi-User Chat [XEP-0045], Publish-Subscribe [XEP-0060], SOCKS5 Bytestreams [XEP-0065], In-Band Registration [XEP-0077], Roster Item Exchange [XEP-0144], and Jingle [XEP-0166].

## 4. Internationalization Considerations

XMPP applications MUST support IDNA2008 for domainparts as described under Section 2.2, the "LocalpartNameClass" subclass for localparts as described under Section 2.3, and the "FreeClass" base string class for resourceparts as described under Section 2.4. This enables XMPP addresses to include a wide variety of characters outside the ASCII range. Rules for enforcement of the XMPP address format are provided in [RFC6120] and specifications for various XMPP extensions.

> Implementation Note: For backward compatibility, many XMPP applications support IDNA2003 [RFC3490] for domainparts, and the stringprep [RFC3454] profiles Nodeprep and Resourceprep [RFC3920] for localparts and resourceparts.

## 5. Security Considerations

### 5.1. 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 [RFC5890] apply to internationalized domain names, which are used here for XMPP domainparts.

### 5.2. Reuse of Unicode

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

### 5.3. Address Spoofing

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

#### 5.3.1. 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 [RFC4422] as described in [RFC6120]). 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 [RFC6120]). 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.

#### 5.3.2. 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 [RFC5890], [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 private 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].

## 6. IANA Considerations

### 6.1. Subclasses

The IANA shall add the following entry to the PRECIS Subclass Registry:

Subclass: LocalpartNameClass.
Base Class: NameClass.
Exclusions: Eight legacy characters in the ASCII range.
Specification: RFC XXXX. [Note to RFC Editor: please change XXXX to the number issued for this specification.]

### 6.2. Usage of NameClass

The IANA shall add the following entry to the PRECIS Usage Registry:

Applicability: Localparts of XMPP addresses.
Base Class: NameClass.
Subclass: Yes, LocalpartNameClass.
Replaces: The Nodeprep profile of Stringprep.
Normalization: NFC.
Casemapping: Map uppercase and titlecase characters to lowercase.
Additional Mappings: Map fullwidth and halfwidth characters to their decomposition mappings (recommended).
Directionality: The "Bidi Rule" defined in RFC 5893 applies.
Specification: RFC XXXX. [Note to RFC Editor: please change XXXX to the number issued for this specification.]

### 6.3. Usage of FreeClass

The IANA shall add the following entry to the PRECIS Usage Registry:

Applicability: Resourceparts of XMPP addresses.
Base Class: FreeClass
Subclass: No.
Replaces: The Resourceprep profile of Stringprep.
Normalization: NFC.
Casemapping: None.
Additional Mappings: Map fullwidth and halfwidth characters to their decomposition mappings (recommended).
Directionality: The "Bidi Rule" defined in RFC 5893 applies.
Specification: RFC XXXX. [Note to RFC Editor: please change XXXX to the number issued for this specification.]

## 7. 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 [RFC2119]

The feature set specified here provides a basis for interoperability testing and follows the spirit of a proposal made by Larry Masinter within the IETF's NEWTRK Working Group in 2005 [INTEROP].

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 that it conforms to the underlying length limits of the DNS.
Section: Section 2.2
Roles: Server MUST, client SHOULD.

Feature: address-domain-prep
Description: Ensure that the domainpart of an XMPP address conforms to IDNA2008, that it contains only NR-LDH labels and U-labels (not A-labels), and that all uppercase and titlecase code points are mapped to their lowercase equivalents.
Section: Section 2.2
Roles: Server MUST, client SHOULD.

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: Server MUST, client SHOULD.

Feature: address-localpart-prep

Description: Ensure that the localpart of an XMPP address conforms to the "LocalpartNameClass" subclass, that all code points are normalized using NFC, and that all uppercase and titlecase code points are mapped to their lowercase equivalents.
Section: Section 2.3
Roles: Server MUST, client SHOULD.

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: Server MUST, client SHOULD.

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: Server MUST, client SHOULD.

## 8. References

### 8.1. Normative References

[FRAMEWORK] Saint-Andre, P. and M. Blanchet, "Precis Framework: Handling Internationalized Strings in Protocols", draft-ietf-precis-framework-05 (work in progress), August 2012.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

[RFC5891] Klensin, J., "Internationalized Domain Names in

Applications (IDNA): Protocol", RFC 5891, August 2010.

[RFC5892] Faltstrom, P., "The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)", RFC 5892, August 2010.

[RFC5893] Alvestrand, H. and C. Karp, "Right-to-Left Scripts for Internationalized Domain Names for Applications (IDNA)", RFC 5893, August 2010.

[RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 2011.

[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/).

[UTR36] The Unicode Consortium, "Unicode Technical Report #36: Unicode Security Considerations", 2008, <http://www.unicode.org/reports/tr36/>.

## 8.2. Informative References

[INTEROP] Masinter, L., "Formalizing IETF Interoperability Reporting", Work in Progress, October 2005.

[I-D.yoneya-precis-mappings]
YONEYA, Y. and T. NEMOTO, "Mapping characters for PRECIS classes", draft-yoneya-precis-mappings-02 (work in progress), July 2012.

[I-D.ietf-precis-nickname]
Saint-Andre, P., "Preparation and Comparison of Nicknames", draft-ietf-precis-nickname-01 (work in progress), September 2012.

[RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, October 1989.

[RFC1535] Gavron, E., "A Security Problem and Proposed Correction With Widely Deployed DNS Software", RFC 1535,

October 1993.

[RFC3454] Hoffman, P. and M. Blanchet, "Preparation of Internationalized Strings ("stringprep")", RFC 3454, December 2002.

[RFC3490] 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.

[RFC3920] Saint-Andre, P., Ed., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 3920, October 2004.

[RFC3921] Saint-Andre, P., Ed., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", RFC 3921, October 2004.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC3987] Duerst, M. and M. Suignard, "Internationalized Resource Identifiers (IRIs)", RFC 3987, January 2005.

[RFC4422] Melnikov, A. and K. Zeilenga, "Simple Authentication and Security Layer (SASL)", RFC 4422, June 2006.

[RFC5122] Saint-Andre, P., "Internationalized Resource Identifiers (IRIs) and Uniform Resource Identifiers (URIs) for the Extensible Messaging and Presence Protocol (XMPP)", RFC 5122, February 2008.

[RFC5894] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Background, Explanation, and Rationale", RFC 5894, August 2010.

[RFC5895] Resnick, P. and P. Hoffman, "Mapping Characters for Internationalized Domain Names in Applications (IDNA) 2008", RFC 5895, September 2010.

[RFC6121] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", RFC 6121, March 2011.

[RFC6122] Saint-Andre, P., "Extensible Messaging and Presence

Protocol (XMPP): Address Format", RFC 6122, March 2011.

[RFC6365] Hoffman, P. and J. Klensin, "Terminology Used in Internationalization in the IETF", BCP 166, RFC 6365, September 2011.

[UTR39] The Unicode Consortium, "Unicode Technical Report #39: Unicode Security Mechanisms", August 2010, <http://unicode.org/reports/tr39/>.

[XEP-0004] Eatmon, R., Hildebrand, J., Miller, J., Muldowney, T., and P. Saint-Andre, "Data Forms", XSF XEP 0004, August 2007.

[XEP-0016] Millard, P. and P. Saint-Andre, "Privacy Lists", XSF XEP 0016, February 2007.

[XEP-0029] Kaes, C., "Definition of Jabber Identifiers (JIDs)", XSF XEP 0029, October 2003.

[XEP-0030] Hildebrand, J., Millard, P., Eatmon, R., and P. Saint-Andre, "Service Discovery", XSF XEP 0030, June 2008.

[XEP-0045] Saint-Andre, P., "Multi-User Chat", XSF XEP 0045, February 2012.

[XEP-0048] Blackman, R., Millard, P., and P. Saint-Andre, "Bookmarks", XSF XEP 0048, November 2007.

[XEP-0054] Saint-Andre, P., "vcard-temp", XSF XEP 0054, July 2008.

[XEP-0060] Millard, P., Saint-Andre, P., and R. Meijer, "Publish-Subscribe", XSF XEP 0060, July 2010.

[XEP-0065] Smith, D., Miller, M., Saint-Andre, P., and J. Karneges, "SOCKS5 Bytestreams", XSF XEP 0065, April 2011.

[XEP-0077] Saint-Andre, P., "In-Band Registration", XSF XEP 0077, January 2012.

[XEP-0079]
Miller, M. and P. Saint-Andre, "Advanced Message Processing", XSF XEP 0079, November 2005.

[XEP-0114]
Saint-Andre, P., "Jabber Component Protocol", XSF XEP 0114, March 2005.

[XEP-0144]
Saint-Andre, P., "Roster Item Exchange", XSF XEP 0144, August 2005.

[XEP-0165]
Saint-Andre, P., "Best Practices to Discourage JID Mimicking", XSF XEP 0165, December 2007.

[XEP-0166]
Ludwig, S., Beda, J., Saint-Andre, P., McQueen, R., Egan, S., and J. Hildebrand, "Jingle", XSF XEP 0166, December 2009.

[XEP-0191]
Saint-Andre, P., "Blocking Command", XSF XEP 0191, July 2012.

[XEP-0203]
Saint-Andre, P., "Delayed Delivery", XSF XEP 0203, September 2009.

[XEP-0220]
Miller, J., Saint-Andre, P., and P. Hancke, "Server Dialback", XSF XEP 0220, August 2012.

[XEP-0292]
Saint-Andre, P. and S. Mizzi, "vCard4 Over XMPP", XSF XEP 0292, October 2011.

[XML]
Maler, E., Yergeau, F., Sperberg-McQueen, C., Paoli, J., and T. Bray, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", World Wide Web Consortium Recommendation REC-xml-20081126, November 2008, <http://www.w3.org/TR/2008/REC-xml-20081126>.

## Appendix A. Differences from RFC 6122

Based on consensus derived from working group discussion, implementation and deployment experience, and formal interoperability

testing, the following substantive modifications were made from RFC 6122.

- Changed domainpart preparation to use IDNA2008 (instead of IDNA2003).
- Changed localpart preparation to use the LocalpartNameClass subclass of the PRECIS NameClass (instead of the Nodeprep profile of Stringprep).
- Changed resourcepart preparation to use the PRECIS FreeClass (instead of the Resourceprep profile of Stringprep).
- Specified that internationalized labels within domainparts must be U-labels (instead of should be U-labels).
- Specified the use of Unicode normalization form C (instead of KC as specified in the Nodeprep and Resourceprep profiles of Stringprep).
- Specified that fullwidth and halfwidth should be mapped to their decomposition mappings (previously handled through the use of NFKC).
- Specified that servers must enforce the address formatting rules.

## Appendix B. Acknowledgements

Thanks to Joe Hildebrand and Florian Zeitz for their feedback.

Some text in this document was borrowed or adapted from [RFC5890], [RFC5891], [RFC5894], and [XEP-0165].

## Author's Address

Peter Saint-Andre
Cisco Systems, Inc.
1899 Wynkoop Street, Suite 600
Denver, CO 80202
USA

Phone: +1-303-308-3282
Email: psaintan@cisco.com