

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: September 15, 2011

P. Saint-Andre  
Cisco  
March 14, 2011

Internationalized Addresses in XMPP  
draft-saintandre-xmpp-il8n-03

Abstract

The Extensible Messaging and Presence Protocol (XMPP) as defined in RFC 3920 used stringprep in the preparation and comparison of non-ASCII characters within XMPP addresses. This document explores a post-stringprep approach to XMPP addresses.

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 September 15, 2011.

Copyright Notice

Copyright (c) 2011 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
2. Proposed PRECIS String Classes . . . . .	4
2.1. Domaineypthings . . . . .	5
2.2. Nameypthings . . . . .	5
2.3. Wordypthings . . . . .	6
2.4. Stringypthings . . . . .	6
3. Normalization . . . . .	7
4. Subclassing . . . . .	8
5. XMPP Use of PRECIS String Classes . . . . .	8
5.1. Localpart . . . . .	8
5.2. Resourcepart . . . . .	9
6. XMPP Migration Issues . . . . .	9
7. XMPP Protocol Slots . . . . .	9
7.1. JID Slot . . . . .	9
7.2. Localpart Slot . . . . .	10
7.3. Resourcepart Slot . . . . .	11
7.4. Wordything Slot . . . . .	11
7.5. Stringything Slot . . . . .	11
8. XMPP Error Handling . . . . .	11
9. XMPP User Interface Issues . . . . .	12
10. Security Considerations . . . . .	12
11. IANA Considerations . . . . .	12
12. Acknowledgements . . . . .	12
13. Informative References . . . . .	12
Author's Address . . . . .	16

## 1. Introduction

The Extensible Messaging and Presence Protocol [RFC6120] is a widely-deployed technology for real-time communication, commonly used for instant messaging (IM) among human users but also for communication among automated systems. XMPP addresses (also called "JabberIDs" or JIDs) are of the form <localpart@domainpart/resourcepart>, where the localpart and resourcepart are formally optional but quite common because they are used to identify clients and other entities on the network. In some sense, XMPP addresses have always been internationalized, because the developers of the original Jabber open-source project intended that all data sent over the wire would consist of UTF-8 encoded Unicode code points. However, at that time (1999) the Jabber developers were quite unsophisticated about internationalization, nor could they simply re-use a reliable internationalization technology that had been developed by the wider Internet community (as they could, for example, by re-using Secure Sockets Layer and Transport Layer Security for channel encryption); this lack of sophistication is evident in the community's first attempt at formally defining the format for JabberIDs in early 2002 [XEP-0029].

When the first instantiation of the IETF's XMPP WG was formed in late 2002, IDNA2003 [RFC3490] had not yet been published and stringprep [RFC3454] was a new technology. During its work on [RFC3920], the XMPP WG absorbed as best it could the advice of internationalization experts regarding appropriate methods for preparing and comparing XMPP addresses; however, the participants in the XMPP WG were ignorant of internationalization and therefore did not necessarily make fully-informed decisions. As a result of this early work, in [RFC3920] the XMPP WG decided to re-use IDNA2003 [RFC3490] and Nameprep [RFC3491] for the domainpart of a JID and to define two additional stringprep profiles: Nodeprep for the localpart and Resourceprep for the resourcepart.

Since the publication of [RFC3920] in 2004, the Internet community has gained more experience with internationalization. In particular, IDNA2003, which is based on stringprep, has been superseded by IDNA2008 ([RFC5890], [RFC5891], [RFC5892], [RFC5893], [RFC5894]), which does not use stringprep. This migration away from stringprep for internationalized domain names has prompted other "customers" of stringprep to consider new approaches to the preparation and comparison of internationalized addresses. As a result, the IETF has formed the PRECIS WG as a common forum for seeking solutions to the problem statement outlined in [PROBLEM].

This document has two purposes: (1) provide input to the PRECIS WG and (2) help inform the decisions of the XMPP WG regarding

internationalization of XMPP addresses, eventually leading to replacement of [RFC6122]. Note well that so far this document present only the author's opinions, and that it does not reflect the consensus of the XMPP WG or the PRECIS WG.

## 2. Proposed PRECIS String Classes

Both [PROBLEM] and [FRAMEWORK] propose that it might be valuable to think of internationalized addresses in terms of broad "string classes". Application technologies like XMPP could either borrow such a string class unchanged or "profile" such a string class with modifications.

This document does not yet make recommendations about borrowing or adapting more general string classes, in part because those classes are not yet clearly defined. However, as input to further discussion, this document explores four string classes that are used in XMPP:

- o Domain names. These are defined in IDNA specification and re-used in XMPP and other applications. However, additional guidelines might be helpful for applications (or at least for XMPP) to fill the gap between what was provided in IDNA2003 (such as normalization and various mapping steps) and what is now provided in IDNA2008. For consistency with the next three string classes we call these "domaineythings".
- o Username-like things. Such a "nameything" is a word or set of words that is used to identify or address a network entity such as a user, an account, a venue (e.g., a chatroom), an information source (e.g., a feed), or a collection of data (e.g., a file). An XMPP localpart is a kind of nameything, but might profile a base definition of nameythings developed by the PRECIS WG.
- o Password-like things. Such a "wordything" is a sequence of letters, numbers, and symbols that is used as a secret for access to some resource on a network (e.g., an account or a venue). In XMPP, wordythings are often used by clients to authenticate with servers, as provided in various SASL mechanisms.
- o Free-form things. Such a "stringything" is a sequence of letters, numbers, symbols, spaces, and other code points that is used for more expressive purposes in an application protocol. An XMPP resourcepart is a kind of stringything, but might profile a base definition of stringythings developed by the PRECIS WG.

The following subsections discuss these string classes in more

detail, with reference to the properties described in Section 3 of [PROBLEM] (input restrictions, normalization, case mapping, and bidirectionality).

## 2.1. Domaineypthings

The IDNA2008 protocol is defined in [RFC5890], [RFC5891], [RFC5892], [RFC5893], and [RFC5894]. However, IDNA2008 covers a smaller range of topics than IDNA2003 [RFC3490]. In particular, normalization and mappings are out of scope for IDNA2008 (although one possible approach is described informationally in [RFC5895]). The XMPP WG, or even the PRECIS WG, might want to choose a normalization form and a set of mappings that would be recommended or required for use on the wire, despite the fact that these matters were not specified in a normative way for IDNA2008. This is especially important in modern application protocols that communicate using UTF-8-encoded Unicode code points instead of 8-bit or 7-bit ASCII (as in older application protocols such as [RFC5322]).

## 2.2. Nameythings

Most application technologies need a special class of strings that can be used to include or communicate things like usernames, chatroom names, file names, and data feed names. We group such things into a bucket called "nameythings". Ideally, the PRECIS WG would define a "nameything" class that could be profiled by various application technologies. We suggest that the base class would have the following features:

- o Control characters (e.g., U+0000 through U+001F) would be disallowed.
- o Space characters (U+0020, along with any code point having a GeneralCategory of Zs) would be disallowed.
- o All other 7-bit ASCII characters (i.e., U+0021 through U+007E) would be protocol-valid, even if their Unicode GeneralCategory is disallowed by the rules specified below.
- o As with IDNA2008, any character that has a compatibility equivalent would be disallowed.
- o Uppercase and titlecase code points would be mapped to their lowercase equivalents.
- o The normalization form would be NFD (see below).
- o Profiles of the base class would be able to exclude specific code points that are included in the base.
- o Profiles of the base class would be able to exclude character classes with other properties (e.g., math symbols) that are included in the base.

OPEN ISSUE: Should symbol characters outside the 7-bit ASCII range be

disallowed?

OPEN ISSUE: How to handle right-to-left code points? It might be reasonable to simply use the "Bidi Rule" from [RFC5893], however "." is allowed in nameythings and the Bidi Rule is probably too complex for our purposes because domaineythings have internal structure (based around the "." character) whereas nameythings do not.

### 2.3. Wordythings

Many application technologies need a special class of strings that can be used to communicate secrets that are typically used as passwords or passphrases. We group such things into a bucket called "wordythings". Ideally, the PRECIS WG would define a "wordything" class that could be profiled by various application technologies. We suggest that the base class would have the following features:

- o Control characters (e.g., U+0000 through U+001F) would be disallowed.
- o Space characters (U+0020, along with any code point having a GeneralCategory of Zs) would be disallowed.
- o All other 7-bit ASCII characters (i.e., U+0021 through U+007E) would be protocol-valid, even if their Unicode GeneralCategory is disallowed by the rules specified below.
- o Any character that has a compatibility equivalent would be disallowed.
- o In order to maximize the entropy of passwords and passphrases, uppercase and titlecase code points would be protocol-valid and would not be mapped to their lowercase equivalents.
- o The normalization form would be NFD (see below).
- o Profiles of the base class would be able to exclude specific code points that are included in the base.
- o Profiles of the base class would be able to exclude character classes with other properties (e.g., math symbols) that are included in the base.

Although some application protocols use passwords and passphrases directly, others re-use technologies that themselves use passwords in some deployments (e.g., this is true of XMPP, which re-uses Simple Authentication and Security Layer or SASL [RFC4422]).

### 2.4. Stringythings

Some application technologies need a special class of strings that can be used in a free-form way. We group such things into a bucket called "stringythings". Ideally, the PRECIS WG would define a "stringything" class that could be profiled by various application technologies. We suggest that the base class would have the

following features:

- o Control characters (e.g., U+0000 through U+001F) would be disallowed.
- o Space characters (U+0020, along with any code point having a GeneralCategory of Zs) would be protocol-valid.
- o All other 7-bit ASCII characters (i.e., U+0021 through U+007E) would be protocol-valid, even if their Unicode GeneralCategory is disallowed by the rules specified below.
- o Characters with compatibility equivalents would be protocol-valid.
- o Uppercase and titlecase code points would protocol-valid and would not be mapped to their lowercase equivalents.
- o The normalization form would be NFD (see below).
- o Profiles of the base class would be able to exclude specific code points that are included in the base.
- o Profiles of the base class would be able to exclude character classes with other properties (e.g., math symbols) that are included in the base.

OPEN ISSUE: How to handle right-to-left codepoints? It might be reasonable to simply use the "Bidi Rule" from [RFC5893], however "." is allowed in stringythings and the Bidi Rule is probably too complex for our purposes because domaineythings have internal structure (based around the "." character) whereas stringythings do not.

### 3. Normalization

Following IDNA2003, existing stringprep profiles all use Unicode Normalization Form KC (NFKC), which performs canonical decomposition and compatibility decomposition, followed by canonical and compatibility recomposition (regarding normalization forms, see [UAX15]). This choice made sense in IDNA2003 because the DNS packet format has fixed-length labels, and NFKC in effect compresses a sequence of characters into the smallest number of bytes possible by performing recomposition. However, experience with some of the application protocols that are currently using NFKC has shown that recomposition is an expensive operation to perform in application servers. In addition, the application protocols that use stringprep all use TCP with security-layer or application-layer compression, so fixing the length of strings is much less important.

What matters most in application protocols is ensuring that network entities (such as clients and servers) all communicate a consistent string representation over the wire. For this purpose, Normalization Form D (NFD), which simply performs canonical decomposition, provides the most efficient approach. As noted above, we can disallow any characters that would require compatibility decomposition, thus

removing the need for compatibility decomposition and recomposition. This is what happened in IDNA2008, enabling IDNA technologies to move from NFKC to NFC. If the same basic approach is taken in the PRECIS WG, while at the same time removing the need for recomposition entirely (by making code points with compatibility equivalents), NFKC (the most complex and therefore most computationally intensive normalization form) can be replaced with NFD (the least complex and therefore least computationally intensive normalization form). Another relevant factor is that  $NFD(x) = NFD(NFD(x))$ , which means that application servers can be optimized for the case where the normalization has already occurred. In general, using NFD will likely result in significant performance improvements within application servers.

#### 4. Subclassing

The opportunity for subclassing PRECIS string classes opens the possibility that different applications technologies will subclass a given class in different ways. For example, imagine that the XMPP community defines a detailed subclass of "nameything" that is optimized for the comparison of JabberIDs. However, the email community might do the same for email addresses. At that point, the XMPP comparison methods might diverge significantly from the mail comparison methods, leading to interoperability problems if a given deployment makes use of the same usernames for both JabberIDs and email addresses. The PRECIS WG needs to consider these matters and find a productive balance between compatibility within an application technology and interoperability across application technologies.

#### 5. XMPP Use of PRECIS String Classes

##### 5.1. Localpart

The localpart of an XMPP address would be redefined as a profile or subclass of the PRECIS "nameything" class. The following additional restrictions would apply:

- o Space characters (U+0020, along with any code point having a GeneralCategory of Zs) would be disallowed.
- o The following Unicode code points would be disallowed: U+0022 ("), U+0026 (&), U+0027 ('), U+002F (/), U+003A (:), U+003C (<), U+003E (>), U+0040 (@).

OPEN ISSUE: Should symbol characters outside the 7-bit ASCII range be disallowed?



## 5.2. Resourcepart

The resourcepart of an XMPP address would be redefined as a profile or subclass of the PRECIS "stringything" class, or might even simply use the identity subclass of "stringything".

## 6. XMPP Migration Issues

Any move away from Nameprep, Nodeprep, and Resourceprep as they are defined today will inevitably introduce the potential for migration issues, such as JIDs that were not ambiguous before the migration but that become ambiguous after the migration. These issues need to be clearly defined and well understood so that the costs and benefits of any change can be properly assessed -- especially if the change might have an impact on authentication (e.g., as described in [RFC3920]), authorization (e.g., presence subscriptions as described in [RFC6121]), access (e.g., joining a chatroom as described in [XEP-0045]), identification (e.g., in XMPP URIs or IRIs as described in [RFC5122]), and other security-related functions.

## 7. XMPP Protocol Slots

IDNA2008 defined the concept of a "domain name slot", i.e., "a protocol element or a function argument or a return value (and so on) explicitly designated for carrying a domain name" (Section 2.3.2.6 of [RFC5890]). Similarly, the XMPP community can define the concepts of a "JID slot", a "localpart slot", and a "resourcepart slot" (and might re-use the concepts of a "nameything slot", "wordything slot", and "stringything slot" from PRECIS specifications). The community has yet to determine the full inventory of such slots. However, the following subsections provide a start at such an inventory.

### 7.1. JID Slot

In XMPP systems, JabberIDs can appear in at least the following slots:

- o Core [RFC6120]: the 'from' and 'to' stream attributes; the 'from' and 'to' stanza attributes.
- o IM [RFC6121]: the 'jid' attribute of the roster <item/> element.
- o Privacy Lists [RFC3921], [XEP-0016]: the 'value' attribute of the <item/> element when the value of the 'type' attribute is "jid".
- o Data Forms [XEP-0004]: the <value/> element when the 'type' attribute is "jid-single" or "jid-multi".

- o Flexible Offline Message Retrieval [XEP-0013]: the 'jid' attribute of the <x/> element.
- o Service Discovery [XEP-0030]: the 'jid' attribute of the <item/> element.
- o Extended Stanza Addressing [XEP-0033]: the 'jid' attribute of the <address/> element.
- o Multi-User Chat [XEP-0045]: the 'actor' child of the <item/> element; the 'jid' attribute of the <item/> element; the 'from' and 'to' attributes of the <invite/> and <decline/> elements; the 'jid' attribute of the <destroy/> element.
- o Bookmarks [XEP-0048]: the 'jid' attribute of the <conference/> element.
- o vCards [XEP-0054]: the <JABBERID/> of the <vCard/> element.
- o Jabber Search [XEP-0055]: the 'jid' attribute of the <item/> element.
- o Publish-Subscribe [XEP-0060]: the 'jid' attribute of the <affiliation/>, <options/>, <subscribe>, <subscription/>, and <unsubscribe/> elements; the 'publisher' attribute of the <item/> element.
- o SOCKS5 Bytestreams [XEP-0065]: the 'jid' attribute of the <streamhost/> and <streamhost-used/> elements.
- o Advanced Message Processing [XEP-0079]: the 'from' and 'to' attributes of the <amp/> element.
- o Jabber Component Protocol [XEP-0114]: the 'from' and 'to' attributes of the <iq/>, <message/>, and <presence/> elements.
- o Message Archiving [XEP-0136]: the 'with' attribute of the <chat/>, <from/>, and <item/> elements.
- o Roster Item Exchange [XEP-0144]: the 'jid' attribute of the <item/> element.
- o Jingle [XEP-0166]: the 'initiator' and 'responder' attributes of the <jingle/> element.
- o Delayed Delivery [XEP-0203]: the 'from' attribute of the <delay/> element.
- o Simple Communications Blocking [XEP-0191]: the 'jid' attribute of the <item/> element.
- o Server Dialback [RFC3921], [XEP-0220]: the 'from' and 'to' attributes of the <result/> and <verify/> elements.
- o Direct MUC Invitations [XEP-0249]: the 'jid' attribute of the <x/> element.

## 7.2. Localpart Slot

In XMPP systems, localparts can appear in at least the following slots:

- o Multi-User Chat [XEP-0045]: the <unique/> element.

- o In-Band Registration [XEP-0077]: the <username/> element.

### 7.3. Resourcepart Slot

In XMPP systems, resourceparts can appear in at least the following slots:

- o Core [RFC6120]: the <resource/> child of the <bind/> element.
- o Multi-User Chat [XEP-0045]: the 'nick' attribute of the <item/> element.
- o Bookmarks [XEP-0048]: the 'nick' attribute of the <conference/> element.
- o Jabber Search [XEP-0055]: the 'nick' attribute of the <item/> and <query/> elements.
- o Publish-Subscribe [XEP-0060]: the 'node' attribute of the <address/> element (this might actually be a "stringything slot" but typically it is handled as a resourcepart).

### 7.4. Wordything Slot

In XMPP systems, generic "wordythings" can appear in at least the following slots:

- o Multi-User Chat [XEP-0045]: the <password/> child of the <destroy/> and <x/> elements.
- o Bookmarks [XEP-0048]: the 'password' attribute of the <conference/> element.
- o Direct MUC Invitations [XEP-0249]: the 'password' attribute of the <x/> element.

### 7.5. Stringything Slot

In XMPP systems, generic "stringythings" can appear in at least the following slots:

- o Flexible Offline Message Retrieval [XEP-0013]: the 'node' attribute of the <x/> element.
- o Extended Stanza Addressing [XEP-0033]: the 'node' attribute of the <address/> element.
- o Publish-Subscribe [XEP-0060]: the 'node' attribute of various XML elements.

## 8. XMPP Error Handling

Both the core XMPP specifications and various XMPP extensions might need to define more robust error handling. Although this topic has yet to be explored in detail, it is likely that specifications can

more widely use the existing <jid-malformed/> error condition defined in [RFC6120].

#### 9. XMPP User Interface Issues

[RFC5895] introduces the helpful concept of "the dividing line between user interface and protocol" and applies that concept to the complex process of translating the user's (presumed) intentions into bits on the wire. IDNA2003 conflated user interface processing and machine-readable protocols, and in many ways XMPP inherited that same error. It would be desirable for XMPP technologies to define a clear dividing line between user interface and protocol. This might mean that the XMPP community will need to define recommended mappings that are applied to a string before it is considered a JID (or the localpart of resourcepart of a JID).

#### 10. Security Considerations

The inclusion of non-ASCII characters in XMPP addresses has important security implications, such as the ability to mimic characters or entire addresses through the inclusion of "confusable characters" (see [RFC4690] and [RFC5890]). These issues are explored at some length in [RFC6122]. Other security considerations might apply and will be described in a future version of this specification.

#### 11. IANA Considerations

This document defines no actions for the IANA.

#### 12. Acknowledgements

Special thanks to Joe Hildebrand for extensive discussions about internationalization and XMPP. Many participants in the XMPP WG Interim Meeting in February 2011 provided valuable feedback. Thanks also to Jack Erwin, Matt Miller, and Tory Patnoe for additional discussions.

#### 13. Informative References

[FRAMEWORK]

Blanchet, M., "Precis Framework: Handling Internationalized Strings in Protocols", draft-blanchet-precis-framework-00 (work in progress),

July 2010.

- [PROBLEM] Blanchet, M. and A. Sullivan, "Stringprep Revision Problem Statement", draft-ietf-precis-problem-statement-01 (work in progress), December 2010.
- [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.
- [RFC3491] Hoffman, P. and M. Blanchet, "Nameprep: A Stringprep Profile for Internationalized Domain Names (IDN)", RFC 3491, March 2003.
- [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.
- [RFC4422] Melnikov, A. and K. Zeilenga, "Simple Authentication and Security Layer (SASL)", RFC 4422, June 2006.
- [RFC4690] Klensin, J., Faltstrom, P., Karp, C., and IAB, "Review and Recommendations for Internationalized Domain Names (IDNs)", RFC 4690, September 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.
- [RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 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.
- [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.
- [RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", draft-ietf-xmpp-3920bis-22 (work in progress), December 2010.
- [RFC6121] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", draft-ietf-xmpp-3921bis-20 (work in progress), January 2011.
- [RFC6122] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Address Format", draft-ietf-xmpp-address-09 (work in progress), January 2011.
- [UAX15] The Unicode Consortium, "Unicode Standard Annex #15: Unicode Normalization Forms", September 2010.
- [XEP-0004] Eatmon, R., Hildebrand, J., Miller, J., Muldowney, T., and P. Saint-Andre, "Data Forms", XSF XEP 0004, August 2007.
- [XEP-0013] Saint-Andre, P. and C. Kaes, "Flexible Offline Message Retrieval", XSF XEP 0013, July 2005.
- [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-0033] Hildebrand, J. and P. Saint-Andre, "Extended Stanza Addressing", XSF XEP 0033, September 2004.
- [XEP-0045] Saint-Andre, P., "Multi-User Chat", XSF XEP 0045, July 2008.
- [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-0055] Saint-Andre, P., "Jabber Search", XSF XEP 0055, September 2009.
- [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 in progress, last updated 2010.
- [XEP-0077] Saint-Andre, P., "In-Band Registration", XSF XEP 0077, September 2009.
- [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-0136] Paterson, I., Perlow, J., Saint-Andre, P., Karneges, J., Tsvyashchenko, A., and Y. Leboulanger, "Message Archiving", XSF XEP 0136, June 2010.

- [XEP-0144] Saint-Andre, P., "Roster Item Exchange", XSF XEP 0144, August 2005.
- [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., "Simple Communications Blocking", XSF XEP 0191, February 2007.
- [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, March 2010.
- [XEP-0249] Saint-Andre, P., "Direct MUC Invitations", XSF XEP 0249, December 2009.

Author's Address

Peter Saint-Andre  
Cisco  
1899 Wyknoop Street, Suite 600  
Denver, CO 80202  
USA

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



