Internet Engineering Task Force (IETF)
Request for Comments: 6122
Updates: 3920
Category: Standards Track
ISSN: 2070-1721

P. Saint-Andre
Cisco

March 2011

Extensible Messaging and Presence Protocol (XMPP): Address Format

Abstract

This document defines the format for addresses used in the Extensible Messaging and Presence Protocol (XMPP), including support for non-ASCII characters. This document updates RFC 3920.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6122.

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 defined canonically by [RFC3920] in 2004.

As specified in RFC 3920, the XMPP address format reuses the "stringprep" technology for preparation of non-ASCII characters [STRINGPREP], including the Nameprep profile for internationalized domain names as specified in [NAMEPREP] and [IDNA2003] along with two XMPP-specific profiles for the localpart and resourcepart.

Since the publication of RFC 3920, IDNA2003 has been superseded by IDNA2008 (see [IDNA-PROTO] and related documents), which is not based on stringprep. Following the lead of the IDNA community, other technology communities that use stringprep have begun discussions about migrating away from stringprep toward more "modern" approaches. The XMPP community is participating in those discussions (mostly within the PRECIS Working Group) in order to find a replacement for the Nodeprep and Resourceprep profiles of stringprep defined in RFC 3920. Because all other aspects of revised documentation for XMPP have been incorporated into [XMPP], the XMPP Working Group decided to temporarily split the XMPP address format into a separate document so as not to significantly delay publication of improved documentation for XMPP. It is expected that this document will be obsoleted as soon as work on a new approach to preparation and comparison of internationalized addresses has been completed.

Therefore, this specification provides corrected documentation of the XMPP address format using the internationalization technologies available in 2004 (when RFC 3920 was published). Although this document normatively references [IDNA2003] and [NAMEPREP], XMPP software implementations are encouraged to begin migrating to IDNA2008 (see [IDNA-PROTO] and related documents) because the specification that obsoletes this one will use IDNA2008 rather than IDNA2003.

This document updates RFC 3920.

1.2. Terminology

Many important terms used in this document are defined in [IDNA2003], [STRINGPREP], [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 or 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*(nodepoint)
                ;
                ; a "nodepoint" is a UTF-8 encoded Unicode code
                ; point that satisfies the Nodeprep profile of
                ; stringprep
                ;
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 ']'), which was not
                ; the case in RFC 3920
                ;
ifqdn         = 1*(namepoint)
                ;
                ; a "namepoint" is a UTF-8 encoded Unicode
                ; code point that satisfies the Nameprep
                ; profile of stringprep
                ;
resourcepart  = 1*(resourcepoint)
                ;
                ; a "resourcepoint" is a UTF-8 encoded Unicode
                ; code point that satisfies the Resourceprep
                ; profile of stringprep
                ;
```

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 purpose 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 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 a description of the process for 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 into 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, a publish-subscribe service, 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 [IDNA2003] or [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, such as application of the [NAMEPREP] profile of [STRINGPREP] or completion of the ToASCII operation as described in [IDNA2003].

A domainpart consisting of a fully qualified domain name MUST be an "internationalized domain name" as defined in [IDNA2003]; that is, it MUST be "a domain name in which every label is an internationalized label" and MUST follow the rules for construction of

internationalized domain names specified in [IDNA2003]. When preparing a text label (consisting of a sequence of UTF-8 encoded Unicode code points) for representation as an internationalized label in the process of constructing an XMPP domainpart or comparing two XMPP domainparts, an application MUST ensure that for each text label it is possible to apply without failing the ToASCII operation specified in [IDNA2003] with the UseSTD3ASCIIRules flag set (thus forbidding ASCII code points other than letters, digits, and hyphens). If the ToASCII operation can be applied without failing, then the label is an internationalized label. (Note: The ToASCII operation includes application of the [NAMEPREP] profile of [STRINGPREP] and encoding using the algorithm specified in [PUNYCODE]; for details, see [IDNA2003].) Although XMPP applications do not communicate the output of the ToASCII operation (called an "ACE label") over the wire, it MUST be possible to apply that operation without failing to each internationalized label. If an XMPP application receives as input an ACE label, it SHOULD convert that ACE label to an internationalized label using the ToUnicode operation (see [IDNA2003]) before including the label in an XMPP domainpart that will be communicated over the wire on an XMPP network (however, instead of converting the label, there are legitimate reasons why an application might instead refuse the input altogether and return an error to the entity that provided the offending data).

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 resulting from application of the Nameprep profile of stringprep (e.g., in Nameprep some characters can be mapped to nothing, which might result in a string of zero length). 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".

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). The entity represented by an XMPP localpart is addressed within the context of a specific domain (i.e., <localpart@domainpart>).

A localpart MUST be formatted such that the Nodeprep profile of [STRINGPREP] can be applied without failing (see Appendix A). Before comparing two localparts, an application MUST first ensure that the Nodeprep profile has been applied to each identifier (the profile need not be applied each time a comparison is made, as long as it has been applied before comparison).

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 resulting from application of the Nodeprep profile of stringprep (e.g., in Nodeprep some characters can be mapped to nothing, which might result in a string of zero length).

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) belonging to the entity associated with an XMPP localpart at a domain (i.e., <localpart@domainpart/resourcepart>).

A resourcepart MUST be formatted such that the Resourceprep profile of [STRINGPREP] can be applied without failing (see Appendix B). Before comparing two resourceparts, an application MUST first ensure that the Resourceprep profile has been applied to each identifier (the profile need not be applied each time a comparison is made, as long as it has been applied before comparison).

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 resulting from application of the Resourceprep profile of stringprep (e.g., in Resourceprep some characters can be mapped to nothing, which might result in a string of zero length).

> Informational Note: For historical reasons, the term "resource identifier" is often used in XMPP to refer to the optional portion of an XMPP address that follows the domainpart and the "/" separator character; to help prevent confusion between an XMPP "resource identifier" and the meanings of "resource" and "identifier" provided in Section 1.1 of [URI], this specification uses the term "resourcepart" instead of "resource identifier" (as in RFC 3920).

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@domain".

- The '@' character is allowed in the resourcepart and is often used in the "nick" shown in XMPP chatrooms. 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 servers MUST, and XMPP clients SHOULD, support [IDNA2003] for domainparts (including the [NAMEPREP] profile of [STRINGPREP]), the Nodeprep (Appendix A) profile of [STRINGPREP] for localparts, and the Resourceprep (Appendix B) profile of [STRINGPREP] 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].

4. Security Considerations

4.1. Reuse of Stringprep

The security considerations described in [STRINGPREP] apply to the Nodeprep (Appendix A) and Resourceprep (Appendix B) profiles defined in this document for XMPP localparts and resourceparts. The security considerations described in [STRINGPREP] and [NAMEPREP] apply to the Nameprep profile that is reused here for XMPP domainparts.

4.2. Reuse of Unicode

The security considerations described in [UNICODE-SEC] 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.

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 such poorly implemented or rogue servers. 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 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. For example, in some XMPP clients the address "ju1iet@example.org" (spelled with the number one as the third character of the localpart) might appear to be the same as "juliet@example.org (spelled with the lower-case version of the letter "L"), especially on casual visual

inspection; this phenomenon is sometimes called "typejacking". A more sophisticated example of address mimicking might involve the use of characters from outside the familiar Latin extended-A block of Unicode code points, such as the characters U+13DA U+13A2 U+13B5 U+13AC U+13A2 U+13AC U+13D2 from the Cherokee block instead of the similar-looking US-ASCII characters "STPETER".

In some examples of address mimicking, it is unlikely that the average user could tell the difference between the real JID and the fake JID. (Indeed, there is no programmatic way to distinguish with full certainty which is the fake JID and which is the real JID; in some communication contexts, the JID formed of Cherokee characters might be the real JID and the JID formed of US-ASCII characters might thus appear to be the fake JID.) Because JIDs can contain almost any properly encoded Unicode code point, it can be relatively easy to mimic some JIDs in XMPP systems. The possibility of address mimicking introduces security vulnerabilities of the kind that have also plagued the World Wide Web, specifically the phenomenon known as phishing.

These problems arise because Unicode and ISO/IEC 10646 repertoires have many characters that look similar (so-called "confusable characters" or "confusables"). In many cases, XMPP users might perform visual matching, such as when comparing the JIDs of communication partners. Because it is impossible to map similar-looking characters without a great deal of context (such as knowing the fonts used), stringprep and stringprep-based technologies such as Nameprep, Nodeprep, and Resourceprep do nothing to map similar-looking characters together, nor do they prohibit some characters because they look like others. As a result, XMPP localparts and resourceparts could contain confusable characters, producing JIDs that appear to mimic other JIDs and thus leading to security vulnerabilities such as the following:

- 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.

Despite the fact that some specific suggestions about identification and handling of confusable characters appear in the Unicode Security Considerations [UNICODE-SEC], it is also true (as noted in [IDNA-DEFS]) that "there are no comprehensive technical solutions to the problems of confusable characters". Mimicked JIDs that involve characters from only one script, or from the script typically employed by a particular user or community of language users, are not easy to combat (e.g., the simple typejacking attack previously described, which relies on a surface similarity between the characters "1" and "l" in some presentations). However, mimicked addresses that involve characters from more than one script, or from a script not typically employed by a particular user or community of language users, can be mitigated somewhat through the application of appropriate registration policies at XMPP services and presentation policies in XMPP client software. Therefore, the following policies are encouraged:

1. Because an XMPP service that allows registration of XMPP user accounts (localparts) plays a role similar to that of a registry for DNS domain names, such a service SHOULD establish a policy about the scripts or blocks of characters it will allow in localparts at the service. Such a policy is likely to be informed by the languages and scripts that are used to write registered account names; in particular, to reduce confusion, the service MAY forbid registration of XMPP localparts that contain characters from more than one script and to restrict registrations to characters drawn from a very small number of scripts (e.g., scripts that are well-understood by the administrators of the service). Such policies are also appropriate for XMPP services that allow temporary or permanent registration of XMPP resourceparts, e.g., during resource binding [XMPP] or upon joining an XMPP-based chat room [XEP-0045]. For related considerations in the context of domain name registration, refer to Section 4.3 of [IDNA-PROTO] and Section 3.2 of [IDNA-RATIONALE]. Note well that methods for enforcing such restrictions are out of scope for this document.

2. Because every human user of an XMPP client presumably has a preferred language (or, in some cases, a small set of preferred languages), an XMPP client SHOULD gather that information either explicitly from the user or implicitly via the operating system of the user's device. Furthermore, because most languages are typically represented by a single script (or a small set of scripts) and most scripts are typically contained in one or more blocks of characters, an XMPP client SHOULD warn the user when presenting a JID that mixes characters from more than one script or block, or that uses characters outside the normal range of the user's preferred language(s). This recommendation is not intended to discourage communication across different communities of language users; instead, it recognizes the existence of such communities and encourages due caution when presenting unfamiliar scripts or characters to human users.

5. IANA Considerations

The following sections update the registrations provided in [RFC3920].

5.1. Nodeprep Profile of Stringprep

The Nodeprep profile of stringprep is defined under Nodeprep (Appendix A). The IANA has registered Nodeprep in the "Stringprep Profiles" registry.

Name of this profile:

Nodeprep

RFC in which the profile is defined:

RFC 6122

Indicator whether or not this is the newest version of the profile:

This is the first version of Nodeprep

5.2. Resourceprep Profile of Stringprep

The Resourceprep profile of stringprep is defined under Resourceprep (Appendix B). The IANA has registered Resourceprep in the "Stringprep Profiles" registry.

Name of this profile:

Resourceprep

RFC in which the profile is defined:

RFC 6122

Indicator whether or not this is the newest version of the profile:

This is the first version of Resourceprep

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 the Nameprep profile of stringprep.
Section:  Section 2.2
Roles:  Client SHOULD, Server 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 Nodeprep profile of stringprep.
Section:  Section 2.3
Roles:  Client SHOULD, Server 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 Resourceprep profile of stringprep.
Section:  Section 2.4
Roles:  Client SHOULD, Server MUST.
```

7. References

7.1. Normative References

[ABNF] Crocker, D., Ed. 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.

[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.

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

[NAMEPREP] Hoffman, P. and M. Blanchet, "Nameprep: A Stringprep Profile for Internationalized Domain Names (IDN)", RFC 3491, March 2003.

See Section 1 for an explanation of why the normative reference to an obsoleted specification is needed.

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

[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/).

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

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

[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.

[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.

[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., Ed., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 3920, October 2004.

[RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, August 2010.

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

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

[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, 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 0045, 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, <http://www.w3.org/TR/2006/REC-xml-20060816>.

[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. Nodeprep

A.1. Introduction

This appendix defines the "Nodeprep" profile of stringprep. As such, it specifies processing rules that will enable users to enter internationalized localparts in the Extensible Messaging and Presence Protocol (XMPP) and have the highest chance of getting the content of the strings correct. (An XMPP localpart is the optional portion of an XMPP address that precedes an XMPP domainpart and the '@' separator; it is often but not exclusively associated with an instant messaging username.) These processing rules are intended only for XMPP localparts and are not intended for arbitrary text or any other aspect of an XMPP address.

This profile defines the following, as required by [STRINGPREP]:

- The intended applicability of the profile: internationalized localparts within XMPP
- The character repertoire that is the input and output to stringprep: Unicode 3.2, specified in A.2
- The mappings used: specified in A.3
- The Unicode normalization used: specified in A.4
- The characters that are prohibited as output: specified in A.5
- Bidirectional character handling: specified in A.6

A.2. Character Repertoire

This profile uses Unicode 3.2 with the list of unassigned code points in Table A.1, both as defined in Appendix A of [STRINGPREP].

A.3. Mapping

This profile specifies mapping using the following tables from [STRINGPREP]:

Table B.1
Table B.2

A.4. Normalization

This profile specifies the use of Unicode Normalization Form KC, as described in [STRINGPREP].

A.5. Prohibited Output

This profile specifies the prohibition of using the following tables from [STRINGPREP].

Table C.1.1
Table C.1.2
Table C.2.1
Table C.2.2
Table C.3
Table C.4
Table C.5
Table C.6
Table C.7
Table C.8
Table C.9

In addition, the following additional Unicode characters are also prohibited:

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., @

A.6. Bidirectional Characters

This profile specifies checking bidirectional strings, as described in Section 6 of [STRINGPREP].

A.7. Notes

Because the additional characters prohibited by Nodeprep are prohibited after normalization, an implementation MUST NOT enable a human user to input any Unicode code point whose decomposition includes those characters; such code points include but are not necessarily limited to the following (refer to [UNICODE] for complete information):

- U+2100 (ACCOUNT OF)
- U+2101 (ADDRESSED TO THE SUBJECT)
- U+2105 (CARE OF)
- U+2106 (CADA UNA)
- U+226E (NOT LESS-THAN)
- U+226F (NOT GREATER-THAN)
- U+2A74 (DOUBLE COLON EQUAL)
- U+FE13 (PRESENTATION FORM FOR VERTICAL COLON)
- U+FE60 (SMALL AMPERSAND)
- U+FE64 (SMALL LESS-THAN SIGN)
- U+FE65 (SMALL GREATER-THAN SIGN)
- U+FE6B (SMALL COMMERCIAL AT)
- U+FF02 (FULLWIDTH QUOTATION MARK)
- U+FF06 (FULLWIDTH AMPERSAND)
- U+FF07 (FULLWIDTH APOSTROPHE)
- U+FF0F (FULLWIDTH SOLIDUS)
- U+FF1A (FULLWIDTH COLON)
- U+FF1C (FULLWIDTH LESS-THAN SIGN)
- U+FF1E (FULLWIDTH GREATER-THAN SIGN)
- U+FF20 (FULLWIDTH COMMERCIAL AT)

Appendix B. Resourceprep

B.1. Introduction

This appendix defines the "Resourceprep" profile of stringprep. As such, it specifies processing rules that will enable users to enter internationalized resourceparts in the Extensible Messaging and Presence Protocol (XMPP) and have the highest chance of getting the content of the strings correct. (An XMPP resourcepart is the optional portion of an XMPP address that follows an XMPP domainpart and the '/' separator.) These processing rules are intended only for XMPP resourceparts and are not intended for arbitrary text or any other aspect of an XMPP address.

This profile defines the following, as required by [STRINGPREP]:

- The intended applicability of the profile: internationalized resourceparts within XMPP

- The character repertoire that is the input and output to stringprep: Unicode 3.2, specified in B.2

- The mappings used: specified in B.3

- The Unicode normalization used: specified in B.4

- The characters that are prohibited as output: specified in B.5

- Bidirectional character handling: specified in B.6

B.2. Character Repertoire

This profile uses Unicode 3.2 with the list of unassigned code points in Table A.1, both as defined in Appendix A of [STRINGPREP].

B.3. Mapping

This profile specifies mapping using the following tables from [STRINGPREP]:

Table B.1

B.4. Normalization

This profile specifies the use of Unicode Normalization Form KC, as described in [STRINGPREP].

B.5. Prohibited Output

This profile specifies the prohibition of using the following tables from [STRINGPREP].

Table C.1.2
Table C.2.1
Table C.2.2
Table C.3
Table C.4
Table C.5
Table C.6
Table C.7
Table C.8
Table C.9

B.6. Bidirectional Characters

This profile specifies checking bidirectional strings, as described in Section 6 of [STRINGPREP].

Appendix C. Differences from RFC 3920

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

- Corrected the ABNF syntax to ensure consistency with [URI] and [IRI], including consistency with RFC 3986 and [RFC5952] with regard to IPv6 addresses (e.g., enclosing the IPv6 address in square brackets '[' and ']' -- see also Section 4.9.3.19 of [XMPP]).

- Corrected the ABNF syntax to prevent zero-length localparts, domainparts, and resourceparts (and also noted that the underlying length limits from the DNS apply to domainparts).

- To avoid confusion with the term "node" as used in [XEP-0030] and [XEP-0060], changed the term "node identifier" to "localpart" (but retained the name "Nodeprep" for backward compatibility).

- To avoid confusion with the terms "resource" and "identifier" as used in [URI], changed the term "resource identifier" to "resourcepart".

- Corrected the Nameprep processing rules to require use of the UseSTD3ASCIIRules flag.

Appendix D. Acknowledgements

Thanks to Ben Campbell, Waqas Hussain, Jehan Pages, and Florian Zeitz for their feedback. Thanks also to Richard Barnes and Elwyn Davies for their reviews on behalf of the Security Directorate and the General Area Review Team, respectively.

The Working Group chairs were Ben Campbell and Joe Hildebrand. The responsible Area Director was Gonzalo Camarillo.

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
Cisco
1899 Wyknoop Street, Suite 600
Denver, CO 80202
USA

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