Network Working Group
Internet-Draft
Intended status: Informational
Expires: October 2, 2014

P. Saint-Andre
&yet

March 31, 2014

An Interoperable Subset of Characters for Internationalized Usernames
draft-saintandre-username-interop-03

Abstract

Various Internet protocols define constructs for usernames, i.e., the localpart of an address such as "localpart@example.com". This document describes a subset of Unicode characters to allow in internationalized usernames for the sake of maximal interoperability across Internet protocols.

Table of Contents

1. Introduction

Various Internet protocols define constructs for usernames, i.e., the localpart of an address such as "localpart@example.com". As further described under Appendix A), examples include the localparts of email addresses, Kerberos Principal Names, Network Access Identifiers, SIP URIs, instant messaging URIs and presence URIs, XMPP addresses, and account URIs, as well as certain forms of SASL simple user names (see [I-D.ietf-precis-saslprepbis]). This document describes a subset of Unicode characters [UNICODE] to allow in internationalized usernames for the sake of maximal interoperability across Internet protocols. This subset might prove useful in cases where a provider offers multiple services (say, email and instant messaging) using the same underlying identifier, or where the same identifier (e.g., an account URI) is used when interacting with multiple providers.

2. Terminology

Many important terms used in this document are defined in [I-D.ietf-precis-framework], [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].

3. Subset

The interoperable subset of characters provided here is defined as a profile of the PRECIS IdentifierClass specified in [I-D.ietf-precis-framework]. In essence, the IdentifierClass restricts the allowable characters to letters and digits from all the scripts of Unicode [UNICODE] while grandfathering all the characters from the ASCII range [RFC20]. The profile defined here,

"LocalpartIdentifierClass", further restricts the characters from the ASCII range to those known to work across existing application protocols (as described under Appendix A).

The syntax is defined as follows using the Augmented Backus-Naur Form (ABNF) as specified in [RFC5234].

```
localpart = 1*1023(localpoint)
          ;
          ; a "localpoint" is a UTF-8 encoded Unicode code point
          ; that conforms to the "LocalpartIdentifierClass"
          ; profile of the PRECIS IdentifierClass
```

A "localpart" MUST consist only of Unicode code points that conform to the "LocalpartIdentifierClass" profile of the "IdentifierClass" base string class defined in [I-D.ietf-precis-framework]. The LocalpartIdentifierClass profile includes all code points allowed by the IdentifierClass base class, with the exception of the following characters, which are disallowed (again, see Appendix A for the reasoning behind these restrictions):

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

U+0023 (NUMBER SIGN), i.e., '#'

U+0025 (PERCENT SIGN), i.e., '%'

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

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

U+0028 (LEFT PARENTHESIS), i.e., '('

U+0029 (RIGHT PARENTHESIS), i.e., ')'

U+002C (COMMA), i.e., ','

U+002E (FULL STOP), i.e., '.'

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

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

U+003B (SEMICOLON), i.e., ';'

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

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

U+003F (QUESTION MARK), i.e., '?'

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

U+005B (LEFT SQUARE BRACKET), i.e., '['

U+005C (REVERSE SOLIDUS), i.e., '\'

U+005D (RIGHT SQUARE BRACKET), i.e., ']'

U+005E (CIRCUMFLEX ACCENT), i.e., '^'

U+0060 (GRAVE ACCENT), i.e., '`'

U+007B (LEFT CURLY BRACKET), i.e., '{'

U+007C (VERTICAL), i.e., '|'

U+007D (RIGHT CURLY BRACKET), i.e., '}'

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

1. Fullwidth and halfwidth characters MUST be mapped to their decomposition mappings.

2. So-called additional mappings MAY be applied, such as mapping of characters that are similar to common delimiters (such as '@', ':', '/', '+', '-', and '.', e.g., mapping of IDEOGRAPHIC FULL STOP (U+3002) to FULL STOP (U+002E)) and special handling of certain characters or classes of characters (e.g., mapping of non-ASCII spaces to ASCII space); the PRECIS mappings document [I-D.ietf-precis-mappings] describes such mappings in more detail.

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

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

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).

A localpart MUST NOT be zero octets in length and MUST NOT be more than 1023 octets in length. This rule is to be enforced after any normalization and mapping of code points.

4. IANA Considerations

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

Name: LocalpartIdentifierClass.

Applicability: Usernames that are intended to be interoperable across multiple application protocols.

Base Class: IdentifierClass.

Replaces: None.

Width Mapping: Map fullwidth and halfwidth characters to their decomposition mappings.

Additional Mappings: None required or recommended.

Case Mapping: Map uppercase and titlecase characters to lowercase.

Normalization: NFC.

Directionality: The "Bidi Rule" defined in RFC 5893 applies.

Exclusions: 24 non-alphanumeric characters in the ASCII range.

Enforcement: Up to the application protocol or deployment.

Specification: this document. [Note to RFC Editor: please change "this document" to the RFC number issued for this specification.]

5. Security Considerations

Deploying usernames that are interoperable across multiple protocols could potentially give malicious entities multiple ways to attack an account or user.

The security considerations described in [I-D.ietf-precis-framework] apply to the "IdentifierClass" base string class used in this document.

The security considerations described in [UTS39] apply to the use of Unicode characters.

6. References

6.1. Normative References

[I-D.ietf-precis-framework]
Saint-Andre, P. and M. Blanchet, "Precis Framework: Handling Internationalized Strings in Protocols", draft-ietf-precis-framework-15 (work in progress), March 2014.

[RFC20] Cerf, V., "ASCII format for network interchange", RFC 20, October 1969.

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

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

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

[UNICODE] The Unicode Consortium, "The Unicode Standard, Version 6.3", 2013, <http://www.unicode.org/versions/Unicode6.3.0/>.

6.2. Informative References

[I-D.ietf-appsawg-acct-uri]
Saint-Andre, P., "The 'acct' URI Scheme", draft-ietf-appsawg-acct-uri-07 (work in progress), January 2014.

[I-D.ietf-precis-mappings]
Yoneya, Y. and T. NEMOTO, "Mapping characters for PRECIS classes", draft-ietf-precis-mappings-07 (work in progress), February 2014.

[I-D.ietf-precis-saslprepbis]
Saint-Andre, P. and A. Melnikov, "Preparation and Comparison of Internationalized Strings Representing Usernames and Passwords", draft-ietf-precis-saslprepbis-07 (work in progress), March 2014.

[RFC821] Postel, J., "Simple Mail Transfer Protocol", STD 10, RFC 821, August 1982.

[RFC2822] Resnick, P., "Internet Message Format", RFC 2822, April 2001.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3856] Rosenberg, J., "A Presence Event Package for the Session Initiation Protocol (SIP)", RFC 3856, August 2004.

[RFC3860] Peterson, J., "Common Profile for Instant Messaging (CPIM)", RFC 3860, August 2004.

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

[RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005.

[RFC4282] Aboba, B., Beadles, M., Arkko, J., and P. Eronen, "The Network Access Identifier", RFC 4282, December 2005.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.

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

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

[UTS39] The Unicode Consortium, "Unicode Technical Standard #39: Unicode Security Mechanisms", July 2012, <http://unicode.org/reports/tr39/>.

Appendix A. Analysis

This document takes the following username constructs into consideration:

- Email addresses [RFC5322]
- Kerberos Principal Names [RFC4120]
- Network Access Identifiers [RFC4282]
- SIP URIs [RFC3261]

- Instant messaging URIs [RFC3860] and presence URIs [RFC3856]

- XMPP addresses (a.k.a. Jabber Identifiers) [RFC6120]

- Account URIs [I-D.ietf-appsawg-acct-uri]

Each of those address formats defines something that can be used as the "localpart" of an address.

The localpart of an email address uses either the "local-part" or the "dot-atom-text" rule in [RFC5322]. Here we make the simplifying assumption that the "dot-atom-text" rule applies:

```
dot-atom-text =  1*atext *("." 1*atext)
atext         =  ALPHA / DIGIT /    ; Any character except
                 "!" / "#" / "$" /  ; controls, SP, and
                 "%" / "&" / "'" /  ; specials. Used for
                 "*" / "+" / "-" /  ; atoms.
                 "/" / "=" / "?" /
                 "^" / "_" / "`" /
                 "{" / "|" / "}" /
                 "~"
```

We make the same simplifying assumption for im: and pres: URIs (although their specifications reference [RFC2822]).

A Kerberos Principal Name is a sequence of strings of type KerberosString, where each KerberosString is a GeneralString that is constrained to contain only characters in IA5String.

```
PrincipalName   ::= SEQUENCE {
        name-type       [0] Int32,
        name-string     [1] SEQUENCE OF KerberosString
}
KerberosString  ::= GeneralString (IA5String)
```

A Network Address Identifier inherits from [RFC821]. Here we care only about the "username" rule:

```
username    =  dot-string
dot-string  =  string
dot-string  =/ dot-string "." string
string      =  char
string      =/ string char
char        =  c
char        =/ "\" x
c           =  %x21    ; '!'              allowed
                       ; '"'              not allowed
c           =/ %x23    ; '#'              allowed
c           =/ %x24    ; '$'              allowed
c           =/ %x25    ; '%'              allowed
c           =/ %x26    ; '&'              allowed
c           =/ %x27    ; '''              allowed
                       ; '(', ')'         not allowed
c           =/ %x2A    ; '*'              allowed
c           =/ %x2B    ; '+'              allowed
                       ; ','              not allowed
c           =/ %x2D    ; '-'              allowed
                       ; '.'              not allowed
c           =/ %x2F    ; '/'              allowed
c           =/ %x30-39 ; '0'-'9'          allowed
                       ; ';', ':', '<'    not allowed
c           =/ %x3D    ; '='              allowed
                       ; '>'              not allowed
c           =/ %x3F    ; '?'              allowed
                       ; '@'              not allowed
c           =/ %x41-5a ; 'A'-'Z'          allowed
                       ; '[', '\', ']'    not allowed
c           =/ %x5E    ; '^'              allowed
c           =/ %x5F    ; '_'              allowed
c           =/ %x60    ; '`'              allowed
c           =/ %x61-7A ; 'a'-'z'          allowed
c           =/ %x7B    ; '{'              allowed
c           =/ %x7C    ; '|'              allowed
c           =/ %x7D    ; '}'              allowed
c           =/ %x7E    ; '~'              allowed
                       ; DEL              not allowed
c           =/ %x80-FF ; UTF-8-Octet      allowed
x           =  %x00-FF ; all 128 ASCII characters
```

The localpart of a sip:/sips: URI inherits from the "userinfo" rule in [RFC3986] with several changes; here we discuss the SIP "user" rule only:

```
user             =  1*( unreserved / escaped / user-unreserved )
user-unreserved  =  "&" / "=" / "+" / "$" / "," / ";" / "?" / "/"
unreserved       =  alphanum / mark
mark             =  "-" / "_" / "." / "!" / "~" / "*" / "'"
                    / "(" / ")"
```

The localpart of an XMPP address allows any ASCII character except space, controls, and the " & ' / : < > @ characters.

The 'acct' URI syntax borrows the 'host', 'pct-encoded', 'sub-delims', 'unreserved' rules from [RFC3986]:

```
acctURI      =  "acct" ":" userpart "@" host
userpart     =  unreserved / sub-delims
                0*( unreserved / pct-encoded / sub-delims )
```

To summarize the foregoing information, the following table lists the allowed and disallowed characters in the localpart of identifiers for each protocol (aside from the alphanumeric, space, and control characters), in order by hexadecimal character number (where each "A" row shows the allowed characters and each "D" row shows the disallowed characters).

Table 1: Allowed and Disallowed Characters (Non-Alphanumeric)

EMAIL ADDRESSES, IM/PRES URIs	
A	! #$%&' *+ - / = ? ^_`{|}~
D	" () , . :;< > @[\]
KERBEROS PRINCIPAL NAMES	
A	!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
D	
NETWORK ADDRESS IDENTIFIERS	
A	! #$%&' *+ - / = ? ^_`{|}~
D	" () , . :;< > @[\]
SIP/SIPS URIs	
A	! $ &'()*+,-./ ; = ? _ ~
D	"# % : < > @[\]^ `{|}
XMPP ADDRESSES	
A	! #$% ()*+,-. ; = ? [\]^_`{|}~
D	" &' /: < > @
ACCT URIs	
A	! $%&'()*+,-. ; = \ ^_`{|}~
D	"# /: < >?@[ ]

The interoperable subset allows only characters that are allowed in all of the foregoing formats, as shown in the following table.

Table 2: Subset Characters (Non-Alphanumeric)

INTEROPERABLE SUBSET	
A	! $ *+ - = _ ~
D	"# %&'() , ./:;< >?@[\]^ `{|}

Appendix B. Acknowledgements

Thanks to Sean Turner for inspiring the work on this document. Thanks also to Paul Hoffman, John Klensin, and Glen Zorn for their comments.

Author's Address

Peter Saint-Andre
&yet
P.O. Box 787
Parker, CO 80134
USA

Email: ietf@stpeter.im