Internet Engineering Task Force (IETF)
Request for Comments: 8266
Obsoletes: 7700
Category: Standards Track
ISSN: 2070-1721

P. Saint-Andre
Jabber.org

October 2017

Preparation, Enforcement, and Comparison of Internationalized Strings Representing Nicknames

Abstract

This document describes methods for handling Unicode strings representing memorable, human-friendly names (called "nicknames", "display names", or "petnames") for people, devices, accounts, websites, and other entities. This document obsoletes RFC 7700.

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 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8266.

Table of Contents

1. Introduction

1.1. Overview

A number of technologies and applications provide the ability for a person to choose a memorable, human-friendly name in a communications context or to set such a name for another entity such as a device, account, contact, or website. Such names are variously called "nicknames" (e.g., in chat room applications), "display names" (e.g., in Internet mail), or "petnames" (see [PETNAME-SYSTEMS]); for consistency, these are all called "nicknames" in this document.

Nicknames are commonly supported in technologies for textual chat rooms, such as:

- Internet Relay Chat (IRC) [RFC2811]
- The Message Session Relay Protocol (MSRP) [RFC4975] [RFC7701]
- Centralized Conferencing (XCON) [RFC5239] [XCON-SYSTEM]
- The Extensible Messaging and Presence Protocol (XMPP) [RFC6120] [XEP-0045]

Recent chat room technologies also allow internationalized nicknames because they support code points from outside the ASCII range [RFC20], typically by means of the Unicode coded character set [Unicode]. Although such nicknames tend to be used primarily for display purposes, they are sometimes used for programmatic purposes as well (e.g., kicking users out of a chat room or avoiding nickname conflicts).

A similar usage enables a person to set their own preferred display name or to set a preferred display name for another user (e.g., the "display-name" construct in the Internet message format [RFC5322] and the <nick/> element in XMPP [XEP-0172]).

Memorable, human-friendly names are also used in contexts other than personal messaging, such as names for devices (e.g., in a network visualization application), websites (e.g., for bookmarks in a web browser), accounts (e.g., in a web interface for a list of payees in a bank account), people (e.g., in a contact list application), and the like.

The rules specified in this document can be applied in all of the foregoing contexts.

It is important to understand that a nickname is a personally memorable name or handle for something that has a more stable, underlying identity, such as a URI or a file path. To ensure secure operation of applications that use nicknames, authentication and authorization decisions MUST be made on the basis of the thing's identity, not its nickname.

To increase the likelihood that memorable, human-friendly names will work in ways that make sense for typical users throughout the world, this document defines rules for handling nicknames in terms of the preparation, enforcement, and comparison of internationalized strings (PRECIS) framework specification [RFC8264].

1.2. Terminology

Many important terms used in this document are defined in [RFC8264], [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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Nickname Profile

2.1. Rules

The following rules apply within the Nickname profile of the PRECIS FreeformClass defined in the PRECIS framework specification [RFC8264].

1. Width Mapping Rule: There is no width mapping rule (such a rule is not necessary because width mapping is performed as part of normalization using Normalization Form KC (NFKC) as specified below).

2. Additional Mapping Rule: The additional mapping rule consists of the following sub-rules.

   a. Map any instances of non-ASCII space to SPACE (U+0020); a non-ASCII space is any Unicode code point having a general category of "Zs", naturally with the exception of SPACE (U+0020). (The inclusion of only ASCII space prevents confusion with various non-ASCII space code points, many of which are difficult to reproduce across different input methods.)

   b. Remove any instances of the ASCII space character at the beginning or end of a nickname (e.g., "stpeter " is mapped to "stpeter").

   c. Map interior sequences of more than one ASCII space character to a single ASCII space character (e.g., "St Peter" is mapped to "St Peter").

3. Case Mapping Rule: Apply the Unicode toLowerCase() operation, as defined in the Unicode Standard [Unicode]. In applications that prohibit conflicting nicknames, this rule helps to reduce the possibility of confusion by ensuring that nicknames differing only by case (e.g., "stpeter" vs. "StPeter") would not be presented to a human user at the same time. (As explained below, this is typically appropriate only for comparison, not for enforcement.)

4. Normalization Rule: Apply Unicode Normalization Form KC. Because NFKC is more "aggressive" in finding matches than other normalization forms (in the terminology of Unicode, it performs both canonical and compatibility decomposition before recomposing code points), this rule helps to reduce the possibility of confusion by increasing the number of code points that would

match; for example, the character "" (ROMAN NUMERAL FOUR, U+2163) would match the combination of "I" (LATIN CAPITAL LETTER I, U+0049) and "V" (LATIN CAPITAL LETTER V, U+0056).

5. Directionality Rule: There is no directionality rule. The "Bidi Rule" (defined in [RFC5893]) and similar rules are unnecessary and inapplicable to nicknames, because it is perfectly acceptable for a given nickname to be presented differently in different layout systems (e.g., a user interface that is configured to handle primarily a right-to-left script versus an interface that is configured to handle primarily a left-to-right script), as long as the presentation is consistent in any given layout system.

Implementation experience has shown that applying the rules for the Nickname profile is not an idempotent procedure for all code points. Therefore, an implementation SHOULD apply the rules repeatedly until the output string is stable; if the output string does not stabilize after reapplying the rules three (3) additional times after the first application, the implementation SHOULD terminate application of the rules and reject the input string as invalid.

2.2. Preparation

An entity that prepares an input string for subsequent enforcement according to this profile MUST ensure that the string consists only of Unicode code points that conform to the FreeformClass string class defined in [RFC8264].

2.3. Enforcement

An entity that performs enforcement according to this profile MUST prepare an input string as described in Section 2.2 and MUST also apply the following rules specified in Section 2.1 in the order shown:

1. Additional Mapping Rule
2. Normalization Rule

Note: An entity SHOULD apply the Case Mapping Rule only during comparison.

After all of the foregoing rules have been enforced, the entity MUST ensure that the nickname is not zero bytes in length (this is done after enforcing the rules to prevent applications from mistakenly omitting a nickname entirely, because when internationalized strings are accepted a non-empty sequence of characters can result in a zero-length nickname after canonicalization).

The result of the foregoing operations is an output string that conforms to the Nickname profile. Until an implementation produces such an output string, it MUST NOT treat the string as conforming (in particular, it MUST NOT assume that an input string is conforming before the enforcement operation has been completed).

2.4. Comparison

An entity that performs comparison of two strings according to this profile MUST prepare each input string as specified in Section 2.2 and MUST apply the following rules specified in Section 2.1 in the order shown:

1. Additional Mapping Rule
2. Case Mapping Rule
3. Normalization Rule

The two strings are to be considered equivalent if and only if they are an exact octet-for-octet match (sometimes called "bit-string identity").

Until an implementation determines whether two strings are to be considered equivalent, it MUST NOT treat them as equivalent (in particular, it MUST NOT assume that two input strings are equivalent before the comparison operation has been completed).

3. Examples

The following examples illustrate a small number of nicknames that are consistent with the format defined above, along with the output string resulting from application of the PRECIS rules for comparison purposes (note that the characters "<" and ">" are used to delineate the actual nickname and are not part of the nickname strings).

#	Nickname	Output for Comparison
1	<Foo>	<foo>
2	<foo>	<foo>
3	<Foo Bar>	<foo bar>
4	<foo bar>	<foo bar>
5	<Σ>	σ (GREEK SMALL LETTER SIGMA, U+03C3)
6	<σ>	σ (GREEK SMALL LETTER SIGMA, U+03C3)
7	<ς>	ς (GREEK SMALL LETTER FINAL SIGMA, U+03C2)
8	<>	ϋ (GREEK SMALL LETTER UPSILON WITH DIALYTIKA, U+03CB)
9	<∞>	∞ (INFINITY, U+221E)
10	<Richard >	<richard iv>

Table 1: A Sample of Legal Nicknames

Regarding examples 5, 6, and 7: applying the Unicode toLowerCase() operation to the character "Σ" (GREEK CAPITAL LETTER SIGMA, U+03A3) results in the character "σ" (GREEK SMALL LETTER SIGMA, U+03C3); however, the toLowerCase() operation does not modify the character "ς" (GREEK SMALL LETTER FINAL SIGMA, U+03C2). Therefore, the comparison operation defined in Section 2.4 would result in matching of the nicknames in examples 5 and 6 but not the nicknames in examples 5 and 7 or 6 and 7.

Regarding example 8: this is an instance where applying the rules for the Nickname profile is not an idempotent procedure (see Section 2.1). In particular:

1. Applying toLowerCase() to the character "" (GREEK UPSILON WITH DIARESIS AND HOOK SYMBOL, U+03D4) results in no changes, and applying NFKC to that character results in the character "Ϋ" (GREEK CAPITAL LETTER UPSILON WITH DIALYTIKA, U+03AB).

2. Applying toLowerCase() to "Ϋ" (GREEK CAPITAL LETTER UPSILON WITH DIALYTIKA, U+03AB) results in the character "ϋ" (GREEK SMALL LETTER UPSILON WITH DIALYTIKA, U+03CB), and applying NFKC to that character results in no changes.

Regarding example 9: symbol characters such as "∞" (INFINITY, U+221E) are allowed by the PRECIS FreeformClass and thus can be used in nicknames.

Regarding example 10: applying the Unicode toLowerCase() operation to the character "" (ROMAN NUMERAL FOUR, U+2163) results in the character "" (SMALL ROMAN NUMERAL FOUR, U+2173), and applying NFKC to the character "" (SMALL ROMAN NUMERAL FOUR, U+2173) results in the characters "i" (LATIN SMALL LETTER I, U+0069) and "v" (LATIN SMALL LETTER V, U+0076).

4. Use in Application Protocols

This specification defines only the PRECIS-based rules for handling of nickname strings. It is the responsibility of an application protocol (e.g., MSRP, XCON, or XMPP) or application definition to specify the protocol slots in which nickname strings can appear, the entities that are expected to enforce the rules governing nickname strings, and the point during protocol processing or interface handling when the rules need to be enforced. See Section 6 of [RFC8264] for guidelines about using PRECIS profiles in applications.

Above and beyond the PRECIS-based rules specified here, application protocols can also define application-specific rules governing nickname strings (rules regarding the minimum or maximum length of nicknames, further restrictions on allowable code points or character ranges, safeguards to mitigate the effects of visually similar characters, etc.).

Naturally, application protocols can also specify rules governing the actual use of nicknames in applications (reserved nicknames, authorization requirements for using nicknames, whether certain nicknames can be prohibited, handling of duplicates, the relationship between nicknames and underlying identifiers such as SIP URIs or Jabber IDs, etc.).

Entities that enforce the rules specified in this document are encouraged to be liberal in what they accept by following this procedure:

1. Where possible, map characters (e.g., through width mapping, additional mapping, case mapping, or normalization) and accept the mapped string.

2. If mapping is not possible (e.g., because a character is disallowed in the FreeformClass), reject the string.

5. IANA Considerations

IANA has added the following entry to the "PRECIS Profiles" registry:

Name: Nickname.

Base Class: FreeformClass.

Applicability: Nicknames or display names in messaging and text conferencing technologies; petnames for devices, accounts, and people; and other uses of nicknames, display names, or petnames.

Replaces: None.

Width Mapping Rule: None (handled via NFKC).

Additional Mapping Rule: Map non-ASCII space characters to SPACE (U+0020), strip leading and trailing space characters, and map interior sequences of multiple space characters to a single instance of SPACE (U+0020).

Case Mapping Rule: Map uppercase and titlecase code points to lowercase using the Unicode toLowerCase() operation.

Normalization Rule: NFKC.

Directionality Rule: None.

Enforcement: To be specified by applications.

Specification: RFC 8266.

6. Security Considerations

6.1. Authentication and Authorization

It is important to understand that a nickname is a personally memorable name or handle for something that has a more stable, underlying identity, such as a URI or a file path. To ensure secure operation of applications that use nicknames, authentication and authorization decisions MUST be made on the basis of the thing's identity, not its nickname.

6.2. Reuse of PRECIS

The security considerations described in [RFC8264] apply to the FreeformClass string class used in this document for nicknames.

6.3. Reuse of Unicode

The security considerations described in [UTS39] apply to the use of Unicode code points in nicknames.

6.4. Visually Similar Characters

[RFC8264] describes some of the security considerations related to visually similar characters, also called "confusable characters" or "confusables", and provides some examples of such characters.

Although the mapping rules defined in Section 2 of this document are designed, in part, to reduce the possibility of confusion about nicknames, this document does not provide more-detailed recommendations regarding the handling of visually similar characters, such as those provided in [UTS39].

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5893] Alvestrand, H., Ed. and C. Karp, "Right-to-Left Scripts for Internationalized Domain Names for Applications (IDNA)", RFC 5893, DOI 10.17487/RFC5893, August 2010, <https://www.rfc-editor.org/info/rfc5893>.

[RFC6365] Hoffman, P. and J. Klensin, "Terminology Used in Internationalization in the IETF", BCP 166, RFC 6365, DOI 10.17487/RFC6365, September 2011, <https://www.rfc-editor.org/info/rfc6365>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8264] Saint-Andre, P. and M. Blanchet, "PRECIS Framework: Preparation, Enforcement, and Comparison of Internationalized Strings in Application Protocols", RFC 8264, DOI 10.17487/RFC8264, October 2017, <https://www.rfc-editor.org/info/rfc8264>.

[Unicode] The Unicode Consortium, "The Unicode Standard", <http://www.unicode.org/versions/latest/>.

[UTS39] Unicode Technical Standard #39, "Unicode Security Mechanisms", edited by Mark Davis and Michel Suignard, <http://unicode.org/reports/tr39/>.

7.2. Informative References

[Err4570] RFC Errata, Erratum ID 4570, RFC 7700, <https://www.rfc-editor.org/errata/eid4570>.

[PETNAME-SYSTEMS] Stiegler, M., "An Introduction to Petname Systems", updated June 2010, February 2005, <http://www.skyhunter.com/marcs/petnames/IntroPetNames.html>.

[RFC20] Cerf, V., "ASCII format for network interchange", STD 80, RFC 20, DOI 10.17487/RFC0020, October 1969, <https://www.rfc-editor.org/info/rfc20>.

[RFC2811] Kalt, C., "Internet Relay Chat: Channel Management", RFC 2811, DOI 10.17487/RFC2811, April 2000, <https://www.rfc-editor.org/info/rfc2811>.

[RFC4975] Campbell, B., Ed., Mahy, R., Ed., and C. Jennings, Ed., "The Message Session Relay Protocol (MSRP)", RFC 4975, DOI 10.17487/RFC4975, September 2007, <https://www.rfc-editor.org/info/rfc4975>.

[RFC5239] Barnes, M., Boulton, C., and O. Levin, "A Framework for Centralized Conferencing", RFC 5239, DOI 10.17487/RFC5239, June 2008, <https://www.rfc-editor.org/info/rfc5239>.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <https://www.rfc-editor.org/info/rfc5322>.

[RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, DOI 10.17487/RFC6120, March 2011, <https://www.rfc-editor.org/info/rfc6120>.

[RFC7700] Saint-Andre, P., "Preparation, Enforcement, and Comparison of Internationalized Strings Representing Nicknames", RFC 7700, DOI 10.17487/RFC7700, December 2015, <https://www.rfc-editor.org/info/rfc7700>.

[RFC7701] Niemi, A., Garcia-Martin, M., and G. Sandbakken, "Multi-party Chat Using the Message Session Relay Protocol (MSRP)", RFC 7701, DOI 10.17487/RFC7701, December 2015, <https://www.rfc-editor.org/info/rfc7701>.

[XCON-SYSTEM] Barnes, M., Boulton, C., and S. Loreto, "Chatrooms within a Centralized Conferencing (XCON) System", Work in Progress, draft-boulton-xcon-session-chat-08, July 2012.

[XEP-0045] Saint-Andre, P., "Multi-User Chat", XSF XEP 0045, September 2017, <https://xmpp.org/extensions/xep-0045.html>.

[XEP-0172] Saint-Andre, P. and V. Mercier, "User Nickname", XSF XEP 0172, March 2012, <https://xmpp.org/extensions/xep-0172.html>.

Appendix A. Changes from RFC 7700

The following changes were made from [RFC7700].

- Addressed [Err4570] by removing the directionality rule from Sections 2.3 and 2.4.
- In accordance with working group discussions and updates to [RFC8264], removed the use of the Unicode toCaseFold() operation in favor of the Unicode toLowerCase() operation.
- Clarified several editorial matters.
- Updated references.

Acknowledgements

Thanks to William Fisher for his implementation feedback, especially regarding idempotence.

Thanks to Sam Whited for his feedback and for submitting [Err4570].

See [RFC7700] for acknowledgements related to the specification that this document supersedes.

Author's Address

Peter Saint-Andre
Jabber.org
P.O. Box 787
Parker, CO 80134
United States of America

Phone: +1 720 256 6756
Email: stpeter@jabber.org
URI: https://www.jabber.org/