Internet Draft
draft-hoffman-i18n-terms-01.txt
January 17, 2001
Expires in six months

Paul Hoffman
IMC & VPNC

Terminology Used in Internationalization in the IETF

Status of this memo

Abstract

This document provides a glossary of terms used in the IETF when discussing internationalization. The purpose is to help frame discussions of internationalization in the various areas of the IETF and to help introduce the main concepts to IETF participants.

1. Introduction

As [RFC2277] summarizes: "Internationalization is for humans. This means that protocols are not subject to internationalization; text strings are." Many protocols throughout the IETF use text strings that are entered by, or are visible to, humans. It should be possible to make these text strings readable to anyone, which means that the text must be able to be displayed in any human language. This is the challenge of internationalization.

1.1 About this document

Internationalization is discussed in many working groups of the IETF. However, few working groups have internationalization experts. When designing or updating protocols, the question often comes up "should we internationalize this" (or, more likely, "do we have to internationalize this").

This document gives an overview of internationalization as it applies to IETF standards work by covering lightly the many aspects of internationalization and the vocabulary associated with those topics. It is not meant to be a complete description of internationalization. The definitions in this document come from many earlier IETF documents and books.

As in many fields, there is disagreement in the internationalization community on definitions for many words. The topic of language brings up particularly passionate opinions for experts and non-experts alike. This document attempts to define terms that will be most useful to the IETF audience.

Note that this is a very early draft of the document. Many definitions here will likely change, and some topics may be added. Discussion of this document is encouraged. Information on the mailing list for this document can be found at <http://www.imc.org/ietf-i18n-terms/>.

1.2 Foundations for internationalization

language

A language is a way that humans interact. The use of language occurs in many forms, the most common of which are writing, vocal, and visual. Each language form is independent: some languages have a close relationship between the written and vocal forms, while others have a looser relationship. [RFC1766] and [RFC1766bis] discuss languages in more detail.

script

A script is a collection of symbols used to represent textual information in one or more writing systems. A script can have many attributes, such as the number of written characters and the rules for combining written characters. Most IETF protocols that deal with languages deal only with scripts, not spoken or visual forms. [RFC2277] discusses scripts in more detail.

It is common for internationalization novices to mix up the terms "language" and "script". This can be a problem in protocols that differentiate the two, such as mail content protocols. Almost all internationalized protocols deal with scripts (the written systems), while fewer deal with languages. Many languages can be expressed using different scripts.

grapheme, phoneme, alphabet

A grapheme is an abstract, atomic written entity of a script. A phoneme is an abstract, atomic spoken entity of a spoken language. An alphabet is a script that maps between graphemes and phonemes.

internationalization

In the IETF, the verb "internationalize" means to add or improve the handling of international information in a protocol. Many protocols that handle text only handle one script, the one that contains the letters used in English text. Internationalizing such a protocol allows the protocol to handle more scripts, hopefully all of the ones useful to anyone in the world.

localization

Internationalized applications can handle a wide variety of languages. Typical users only understand a small number of languages, so the program must be tailored to interact with users in just the languages they know. Localization involves not only changing the language interaction, but also other relevant changes such as display of numbers, dates, currency, and so on.

i18n, l10n

These are abbreviations for "internationalization" and "localization". "18" is the number of characters between the "i" and the "n" in "internationalization", and "10" is the number of characters between the "l" and the "n" in "localization".

2. Fundamental Terms

This section covers basic topics that are needed for almost anyone who is involved with internationalization of IETF protocols. Many terms in this section are based on [IDN-REQ].

characters

A character is a member of a set of elements used for organization, control, or representation of data. In written form, a language is expressed in characters. The same set of characters can often be used to express many languages.

In ISO 10646 [ISO10646], a character is identified by its name, not by its shape. A name may suggest a meaning, but the character may be used for representing other meanings as well. A name may suggest a shape, but that does not imply that just that is commonly used in print.

coded character

A coded character is a character with its coded representation. A coded character set (CCS) is a set of unambiguous rules that establish a character set and the relationship between the characters of the set and their coded representation. The specified set of characters in a CCS is often called the "repertoire".

character encoding scheme

A character encoding scheme (CES) is a mapping from one or more coded character sets to a set of octets. Some CESs are associated with a single CCS; for example, UTF-8 [RFC2279] applies only to ISO 10646. Other CESs, such as ISO 2022, are associated with many CCSs.

charset

A charset is a method of mapping a sequence of octets to a sequence of abstract characters. A charset is, in effect, a combination of one or more CCS with a CES. Charset names are registered by the IANA according to procedures documented in [RFC2278]. A particular charset may have different glyphs depending on the language being used.

transfer encoding syntax

A transfer encoding syntax (TES) (sometimes called a transfer encoding scheme) is a reversible transform of already-encoded data represented in one or more character encoding schemes. TESs are useful for encoding types of character data into an another format, usually for allowing new types of data to be transmitted over legacy protocols. Examples of TESs used in the IETF include Base64 and quoted-printable (described in more detail in Chapter 6).

3. Standards Bodies and Standards

This section describes some of the standards bodies and standards that appear in discussions of internationalization in the IETF. This is an incomplete and possibly over-full list; listing too few bodies or standards can be just as politically dangerous as listing too many. Note that there are many other bodies that deal with internationalization; however, none of them appear commonly in IETF standards work.

ISO

The International Organization for Standardization has been involved with standards for scripts since before the IETF was started. ISO is a non-governmental group made up of national bodies. ISO has many diverse standards in the international scripts area; the one that is most used in the IETF is commonly referred to as "ISO 10646", although its official name has more qualifications. ISO 10646 describes a CCS that covers almost all known written characters in use today.

ISO 10646 is controlled by the group known as "ISO/IEC JTC 1/SC 2 WG2", often called "WG2" for short. ISO standards go through many steps before being finished, and years often go by between changes to ISO 10646. Information on WG2, and its work products, can be found at <http://anubis.dkuug.dk/JTC1/SC2/WG2/>.

Unicode Consortium

The second important group for international character standards is the Unicode Consortium. The Unicode Consortium is a trade association of companies and governments interested in promoting the Unicode Standard [Unicode3]. The Unicode Standard is a CCS whose repertoire is identical to ISO 10646. The Unicode Consortium has added features to the base CCS which make it more useful in protocols, such as defining attributes for each character.

The Unicode Consortium publishes addenda to the Unicode Standard as Unicode Technical Reports. There are many types of technical reports at various stages of maturity. The Unicode Standard and affiliated technical reports can be found at <http://www.unicode.org/>

encodings and transformations of ISO 10646

Characters in the ISO 10646 CCS can be expressed in many ways. Encoding forms are direct addressing methods, while transformation formats are methods for expressing encoding forms as bits on the wire. Two encoding forms are defined for ISO 10646:

UCS-2 addresses the first 2^15 characters as 16-bit values. This range is also called the "Basic Multilingual Plane" (BMP), and is also called "plane 0".

UCS-4 addresses the entire range of 2^31 characters as 32-bit values.

Many transformation formats of the CCS are used in IETF standards. The two most common are:

UTF-8, defined in [RFC2279], is the preferred encoding for IETF protocols. Characters in the BMP are encoded as one, two, or three octets.

UTF-16 (BE & LE), defined in [RFC2781], is used much less often than UTF-8. Characters in the BMP are always encoded as two octets, and many characters outside the BMP as four octets.

native CCSs and charsets

Before ISO 10646 was developed, many countries developed their own CCSs and charsets. Many dozen of these are in common use on the Internet today. Examples include ISO 8859-5 (Russia) and Shift-JIS (Japan). The official list of the registered charset tags is maintained by IANA.

ASCII

Probably the most well-known native CCS is ASCII [US-ASCII]. This CCS is used in numerous IETF protocols that have not yet been internationalized.

local and regional standards organizations

Just as there are many native CCSs and charsets, there are many local and regional standards organizations to create and support them. Common examples of these are ANSI (United States), JIS (Japan), GB (China), and CEN/ISSS (Europe).

4. Linguistic Issues

This section contains terms and topics that are commonly used in linguistics and therefore are of concern to people internationalizing protocols. These topics are standardized outside the IETF.

composition and decomposition

In some CCSs, some characters consist of combinations of other characters. For example, the letter "a with acute" might be a combination of the two characters "a" and "combining acute". The rules for combining two or more characters are called "composition", and the rules for taking apart a character into other characters is called "decomposition".

normalization and canonicalization

These two terms are often used interchangeably in internationalization. Generally, they both mean to convert a string of one or more characters into another string based on standardized rules. In internationalized text, these rules are usually based on decomposing combined characters or composing characters with combining characters. [UTR15] describes the process and many forms of normalization in detail. Normalization is important when comparing strings to see if they are the same.

locale and region

Because languages differ from country to country (and even region to region within a country), the locale of the user of internationalized text can often be an important factor. Typically, the locale information for a user includes the language(s) used. Locale issues go beyond character use, and can include things such as the display format for currency, dates, and times.

case

In many scripts, particularly those from or based on European alphabets, there are two forms for letters: uppercase and lowercase. There is usually (but not always) a one-to-one mapping between the same letter in the two cases. However, there are many examples of characters which exist in one case but for which there is no corresponding character in the other case. Case conversion can even be dependant on locale. Converting between the two cases is sometimes called "folding".

sorting

Many processes have a need to order strings in a consistent sequence (sorted). For some CCS/CES combinations, there is an obvious sort order that can be done without reference to the linguistic meaning of the characters: the codepoint order is sufficient. For other CCS/CES (such as the ISO 2022 family) there is no such order that works well.

Codepoint order is usually not how any human educated by a local school system expects to see strings ordered; if one orders to the expectations of a human, one has a localized sort.

Sorting to codepoint order will seem inconsistent if the strings are not normalized before sorting because different representations of the same character will sort differently. This problem may be smaller with a localized sort.

glyph

A graphic character or glyph is a character, other than a control function, that has a visual representation (normally handwritten, printed, or displayed). It is a recognizable abstract graphic symbol which is independent of any specific design. There are many types of glyphs, including alphabetic characters, digits, punctuation, diacritics, and symbols.

font

A collection of glyph images having the same basic design.

code table or code page

A tabular representation of a coded character set, also showing the coded representations.

code space

The numeric domain occupied by all bit combinations used for the coding of a coded character set.

4.1 Types of characters

alphabetic

Characters from the spoken part of phonetic or syllabic scripts. Examples include Latin letters a through z, Arabic letters, and Katakana characters from Japanese.

ideographic

Characters that, by themselves, represent words or concepts (as compared to phonetic sounds). Examples include Han characters used in Chinese, Japanese, and Korean.

punctuation

Non-alphabetic characters that are used to delimit sounds, sentences, and phrases. Examples include the period, comma, and hyphen.

symbols

Characters that represent pictures or icons (as compared to ones that represent letters or punctuation). Examples include characters for arrows, faces, and geometric shapes.

spacing characters

Characters that represent horizontal or vertical spaces in written text. Examples include the space character, the tab character, and the paragraph mark.

diacritics

Characters that combine with alphabetic characters, usually to change the spoken pronunciation of the base alphabetic character. Examples include the combining acute accent, combining tilde, and combining ring above.

combining character

Characters that visually change the characters that precede them. Examples include combining diacritics, many Arabic alphabetic letters, and many letters from the Indic scripts.

control character

Non-displaying characters that cause changes in the systems in which they are entered. Examples include the null character, the delete character, and the bell character.

formatting character

Non-displaying character that has an effect on the surrounding characters. Examples include characters for specifying the direction of text, letter-spacing characters, and characters for specifying joining.

5. User interface for internationalized text

Although the IETF does not standardize user interfaces, many protocols make assumptions about how a user will enter or see text that is used in the protocol. Many protocols make inherent assumptions such as that text will be typed on a standard (that is, US-centric) keyboard, or that text will be displayed on a character-based monitor. Internationalization challenges assumptions like these, and it is therefore useful to consider how users typically interact with internationalized text.

input methods

Text can be entered into a computer in many ways. Keyboards are by far the most common method used, but many characters cannot be entered on typical computer keyboards. Many operating systems come with system software that lets users input characters outside the range of what is allowed by keyboards.

For example, there are dozens of different input method for Han characters in Chinese, Japanese, and Korean. Some start with phonetic input through the keyboard, while others use the number of strokes in the character. Input methods are also needed for scripts that have many diacritics, such as European characters that have two or three diacritics on a single alphabetic character.

display methods

Many scripts can be directly displayed with fonts, where each character from an input stream can simply be copied from a font system and put on the screen. Other scripts need rules that are based on the input stream in order to display text. Some examples of these display rules include:

- Scripts such as Arabic (and many others), where the form of the letter changes depending on whether the letter is standing alone, at the beginning of a word, in the middle of a word, or at the end of a word

- Scripts such as the Indic scripts, where consonants may change their form if they are adjacent to certain other consonants

- Arabic and Hebrew script, where the order of the characters displayed can be changed with right-to-left and left-to-right ordering marks

rendering characters

Combining characters modify the display of the character (or, in some cases, characters) that precede them. When rendering such text, the display engine must either find the character in the font that contains the base character and all of the combining characters, or it must render the combination itself. Such rendering can be straight-forward, but it is sometimes complicated when the combining marks interact with each other, such as when there are two combining marks that would appear above one character.

bidirectional scripts

Most of the world's written languages are displayed left-to-right. However, many widely-used written languages such as Hebrew and ones based on the Arabic script are displayed right-to-left. Right-to-left text often confounds protocol writers because they have to keep thinking in terms of the order of characters in a string in memory, and that order might be different than what they see on the screen.

Bidirectional text can cause even more confusion because there are formatting characters in ISO 10646 which cause the order of display of text to change. These formatting characters are needed so that one can properly display right-to-left characters with left-to-right characters at the same time without forcing the one of the strings to be stored in reverse order. [Unicode3] has a long and incredibly detailed discussion of bidirectional text.

undisplayable characters

Some characters have no displayable form. For instance, the zero-width space (U+200B) cannot be displayed because it takes up no horizontal space. Formatting characters such as those for setting the direction of text are also undisplayable.

6. Text in current IETF protocols

Many IETF protocols started off being fully internationalized, while others have been internationalized as they were revised. In this process, IETF members have seen patterns in the way that many protocols use text. This section describes some specific protocol interactions with text.

protocol elements

Almost every protocol has named elements, such as "source port" in TCP. In some protocols, the names of the elements (or text aliases for the names) are transmitted within the protocol. For example, in SMTP, the names of the verbs are part of the command stream. The names of protocol elements are not normally seen by end users.

on-the-wire encoding

Characters exist as codepoints in a charset. Before being transmitted in a protocol, they must first be encoded. Similarly, when characters are received in a transmission, they have been encoded, and a protocol that needs to process the individual characters needs to decode them before processing. The encoding and decoding used before and after transmission is often called the "on-the-wire" (or sometimes just "wire") format.

name spaces

Many items in Internet protocols use names to identify content. The field of names for particular item is called its name space. The names in a name space may be controlled centrally (such as by IANA) or may have distributed control, such as the names in the DNS.

parsed text

In some protocols, free text in text fields might be parsed. For example, many mail user agents will parse the words in the text of the Subject: field to attempt to thread based on the "Re:" prefix.

charset tagging

Protocols that allow more than one charset to be used on text that is meant for human consumption usually require that the text be tagged with the appropriate charset. Without this tagging, a program looking at the text cannot definitively discern the charset of the text.

language tagging

Some protocols allow text that is meant for machine processing to be tagged with the language used in the text. Such tagging is important for machine-processing of the text, such as by systems that "display" the text by speaking it.

MIME

MIME (Multipurpose Internet Mail Extensions) is a message format that allows for textual message bodies and headers in character sets other than US-ASCII in formats that require ASCII (most notably, RFC 822). MIME is described in RFCs 2045 through 2049.

Base64

Base64 is a transfer encoding syntax that allows binary data to be represented by the ASCII characters A through Z, a through z, 0 through 9, +, /, and =. It is described in RFC 2045.

quoted printable

The original design of the quoted printable transfer encoding syntax was to allow strings that had some non-ASCII printable characters mixed in with mostly ASCII printable characters to be human readable. It is described in RFC 2047. It is generally considered to be somewhat of a failure at being readable.

ASN.1 text formats

The ASN.1 data description language has many formats for text items. The formats allow for different repertoires and different encodings. Some of the formats that appear in IETF standards based on ASN.1 include IA5String (all ASCII characters), PrintableString (most ASCII characters, but missing many punctuation characters), BMPString (characters from ISO 10646 plane 0 in UTF-16BE format), UTF8String (just as the name implies), and TeletexString (also called T61String; the repertoire changes over time).

ASCII-compatible encoding

Starting in 2000, many ASCII-compatible encoding schemes (which are actually transfer encoding syntaxes) have been proposed as possible solutions for internationalizing host names. Their goal is to be able to encode any string of ISO 10646 characters as legal DNS host names (as described in STD 13). At the time of this writing, non ACE has become an IETF standard.

7. Other Common Terms In Internationalization

This is a hodge-podge of other terms that have appeared in internationalization discussions in the IETF. It is likely that additional terms will be added as this document matures.

Latin

"Latin characters" is a not-precise term for characters derived from Greek script and used throughout the world. The base Latin characters make up the ASCII repertoire and have been augmented by many single and multiple diacritics.

romanization

Because of the widespread use of Latin characters, people have tried to represent many languages that are not based on a Latin repertoire in Latin. This process is called "romanization". For example, there are two popular romanizations of Chinese: Wade-Giles and Pinyin, the latter of which is by far more common today. Most romanization systems are inexact and do not give perfect round trip mappings between the native script and the Latin characters.

CJK and Han

The ideographic characters used in Chinese, Japanese, Korean, and traditional Vietnamese scripts are often called "CJK" characters after the initial letters of the script names in English. They are also called "Han" characters, after the romanized translation of the term in Chinese that is often used for these characters. Note that CJK and Han characters do not include the phonetic characters of the Japanese or Korean alphabets.

translation

The process of converting one language to another, or one script to another, is called translation. Most language translation systems are inexact and do not give one-to-one round trip mappings between the languages. Most script translations are exact, and many have perfect round-trip mappings.

regular expressions

Pattern matching for text involves being able to represent one or more code points in an abstract notation, such as searching for all capital Latin letters or all punctuation. The most common mechanism in IETF protocols for naming such patterns is the use of regular expressions.

private use characters

Many CCSs have code points that are defined as existing characters that have not defined semantics. The use of these "private use" characters is defined by the parties who transmit and receive them, and is thus not appropriate for standardization.

8. Security Considerations

Security is not discussed in this document.

9. References

[IDN-REQ] "Requirements of Internationalized Domain Names", work in progress (draft-ietf-idn-requirements), Z. Wenzel and J. Seng.

[ISO10646] ISO/IEC 10646-1:2000. International Standard -- Information technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane.

[RFC1766] "Tags for the Identification of Languages", RFC 1766, H. Alvestrand.

[RFC1766bis] "Tags for the Identification of Languages", work in progress (draft-alvestrand-lang-tag-v2), H. Alvestrand.

[RFC2277] "IETF Policy on Character Sets and Languages", RFC 2277, H. Alvestrand.

[RFC2279] "UTF-8, a transformation format of ISO 10646", RFC 2279, F. Yergeau.

[RFC2781] "UTF-16, an encoding of ISO 10646", RFC 2781,
P. Hoffman and F. Yergeau.

[Unicode3] The Unicode Consortium, "The Unicode Standard -- Version 3.0", ISBN 0-201-61633-5. Described at <http://www.unicode.org/unicode/standard/versions/Unicode3.0.html>.

[US-ASCII] Coded Character Set -- 7-bit American Standard Code for Information Interchange, ANSI X3.4-1986.

[UTR15] "Unicode Normalization Forms", Unicode Technical Report

#15, M. Davis & M. Duerst.

10. Additional Interesting Reading

ALA-LC Romanization Tables, Randall Barry (ed.), ISBN 0844409405

Blackwell Encyclopedia of Writing Systems, Florian Coulmas, ISBN 063121481X

Unicode Standard version 3.0, Unicode Consortium, ISBN 0201616335

Writing Systems of the World, Akira Nakanishi, ISBN 0804816549

A. Acknowledgements

The definitions in this document come from many sources, including a wide variety of IETF documents.

James Seng contributed to the initial outline of this document. Others who contributed to the development include:

Jacob Palme
Susan Harris
Harald Alvestrand
Johan van Wingen
Yuri Demchenko

B. Author Contact Information

Paul Hoffman
Internet Mail Consortium and VPN Consortium
127 Segre Place
Santa Cruz, CA 95060 USA
paul.hoffman@imc.org and paul.hoffman@vpnc.org