Network Working Group
Request for Comments: 3536
Category: Informational

P. Hoffman
IMC & VPNC

May 2003

Terminology Used in Internationalization in the IETF

Status of this Memo

Copyright Notice

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.

Table of Contents

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 for anyone to enter or read these text strings, which means that Internet users must be able to be enter text in typical input methods and displayed in any human language. Further, text containing any character should be able to be passed between Internet applications easily. This is the challenge of internationalization.

1.1 Purpose of this document

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.

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 lightly covering 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 are not normative for IETF standards; however, they are useful and standards may make informative reference to this document after it becomes an RFC. Some of 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 in a way that will be most useful to the IETF audience.

This document uses definitions from many documents that have been developed outside the IETF. The primary documents used are:

- ISO/IEC 10646 [ISOIEC10646]

- The Unicode Standard [UNICODE]

- W3C Character Model [CHARMOD]

- IETF RFCs, including [RFC2277]

1.2 Format of the definitions in this document

In the body of this document, the source for the definition is shown in angle brackets, such as "<ISOIEC10646>". Many definitions are shown as "<NONE>", which means that the definitions were crafted originally for this document. The angle bracket notation for the source of definitions is different than the square bracket notation used for references to documents, such as in the paragraph above; these references are given in Section 9.

For some terms, there are commentary and examples after the definitions. In those cases, the part before the angle brackets is the definition that comes from the original source, and the part after the angle brackets is commentary that is not a definition (such as examples or further exposition).

Examples in this document use the notation for code points and names from the Unicode Standard [UNICODE] and ISO/IEC 10646 [ISOIEC10646]. For example, the letter "a" may be represented as either "U+0061" or "LATIN SMALL LETTER A".

2. Fundamental Terms

This section covers basic topics that are needed for almost anyone who is involved with making IETF protocols more friendly to non-ASCII text and with other aspects of internationalization.

language

A language is a way that humans interact. The use of language occurs in many forms, the most common of which are speech, writing, and signing. <NONE>

Some languages have a close relationship between the written and spoken forms, while others have a looser relationship. [RFC3066] discusses languages in more detail and provides identifiers for languages for use in Internet protocols. Note that computer languages are explicitly excluded from this definition.

script

A set of graphic characters used for the written form of one or more languages. <ISOIEC10646>

Examples of scripts are Latin, Cyrillic, Greek, Arabic, and Han (the ideographs used in writing Chinese, Japanese, and Korean). [RFC2277] discusses scripts in 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. Almost all protocols that are designed (or were re-designed) to handle non-ASCII text deal with scripts (the written systems) or characters, while fewer actually deal with languages.

A single name can mean either a language or a script; for example, "Arabic" is both the name of a language and the name of a script. In fact, many scripts borrow their names from the names of languages. Further, many scripts are used for many languages; for example, the Russian and Bulgarian languages are written in the Cyrillic script. Some languages can be expressed using different scripts; the Mongolian language can be written in either the Mongolian and Cyrillic scripts, and the Serbo-Croatian language is written using both the Latin and Cyrillic scripts. Further, some languages are normally expressed with more than one script at the same time; for example, the Japanese language is normally expressed in the Kanji (Han), Katakana, and Hiragana scripts in a single string of text.

character

A member of a set of elements used for the organization, control, or representation of data. <ISOIEC10646>

There are at least three common definitions of the word "character":

- a general description of a text entity
- a unit of a writing system, often synonymous with "letter" or similar terms
- the encoded entity itself

When people talk about characters, they are mostly using one of the first two definitions.

A particular 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 only that shape is commonly used in print, nor that the particular shape is associated only with that name.

coded character

A character together with its coded representation. <ISOIEC10646>

coded character set

A coded character set (CCS) is a set of unambiguous rules that establishes a character set and the relationship between the characters of the set and their coded representation. <ISOIEC10646>

character encoding form

A character encoding form is a mapping from a character set definition to the actual code units used to represent the data. <UNICODE>

repertoire

The collection of characters included in a character set. Also called a character repertoire. <UNICODE>

glyph

A glyph is an abstract form that represents one or more glyph images. The term "glyph" is often a synonym for glyph image, which is the actual, concrete image of a glyph representation having been rasterized or otherwise imaged onto some display surface. In displaying character data, one or more glyphs may be selected to depict a particular character. These glyphs are selected by a rendering engine during composition and layout processing. <UNICODE>

glyph code

A glyph code is a numeric code that refers to a glyph. Usually, the glyphs contained in a font are referenced by their glyph code. Glyph codes are local to a particular font; that is, a different font containing the same glyphs may use different codes. <UNICODE>

transcoding

Transcoding is the process of converting text data from one character encoding form to another. Transcoders work only at the level of character encoding and do not parse the text. Note: Transcoding may involve one-to-one, many-to-one, one-to-many or many-to-many mappings. Because some legacy mappings are glyphic, they may not only be many-to-many, but also discontinuous: thus XYZ may map to yxz. <CHARMOD>

In this definition, "many-to-one" means a sequence of characters mapped to a single character. The "many" does not mean alternative characters that map to the single character.

character encoding scheme

A character encoding scheme (CES) is a character encoding form plus byte serialization. There are many character encoding schemes in Unicode, such as UTF-8 and UTF-16. <UNICODE>

Some CESs are associated with a single CCS; for example, UTF-8 [RFC2279] applies only to ISO/IEC 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 CCSs with a CES. Charset names are registered by the IANA according to procedures documented in [RFC2278]. <NONE>

Many protocol definitions use the term "character set" in their descriptions. The terms "charset" or "character encoding scheme" are strongly preferred over the term "character set" because "character set" has other definitions in other contexts and this can be confusing.

internationalization

In the IETF, "internationalization" means to add or improve the handling of non-ASCII text in a protocol. <NONE>

Many protocols that handle text only handle one script (often, the one that contains the letters used in English text), or leave the question of what character set is used up to local guesswork

(which leads, of course, to interoperability problems). Adding non-ASCII text to such a protocol allows the protocol to handle more scripts, hopefully all of the ones useful in the world.

localization

The process of adapting an internationalized application platform or application to a specific cultural environment. In localization, the same semantics are preserved while the syntax may be changed. [FRAMEWORK]

Localization is the act of tailoring an application for a different language or script or culture. Some 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.

The major work of localization is translating the user interface and documentation. Localization involves not only changing the language interaction, but also other relevant changes such as display of numbers, dates, currency, and so on. The better internationalized an application is, the easier it is to localize it for a particular language and character encoding scheme.

Localization is rarely an IETF matter, and protocols that are merely localized, even if they are serially localized for several locations, are generally considered unsatisfactory for the global Internet.

Do not confuse "localization" with "locale", which is described in Section 7 of this document.

i18n, l10n

These are abbreviations for "internationalization" and "localization". <NONE>

"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".

multilingual

The term "multilingual" has many widely-varying definitions and thus is not recommended for use in standards. Some of the definitions relate to the ability to handle international

characters; other definitions relate to the ability to handle multiple charsets; and still others relate to the ability to handle multiple languages. <NONE>

displaying and rendering text

To display text, a system puts characters on a visual display device such as a screen or a printer. To render text, a system analyzes the character input to determine how to display the text. The terms "display" and "render" are sometimes used interchangeably. Note, however, that text might be rendered as audio and/or tactile output, such as in systems that have been designed for people with visual disabilities. <NONE>

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 glyph in the font that represents 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 the same character. Formatting characters can also change the way that a renderer would display text. Rendering can also be difficult for some scripts that have complex display rules for base characters, such as Arabic and Indic scripts.

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, few if any of them appear commonly in IETF standards work.

3.1 Standards bodies

ISO

The International Organization for Standardization has been involved with standards for characters 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 characters area; the one that is most used in the IETF is commonly referred to as "ISO/IEC 10646", although its official name has

more qualifications. (The IEC is International Electrotechnical Commission). ISO/IEC 10646 describes a CCS that covers almost all known written characters in use today.

ISO/IEC 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/IEC 10646. Information on WG2, and its work products, can be found at <http://www.dkuug.dk/JTC1/SC2/WG2/>.

The standard, which comes in multiple parts, can be purchased in both print and CD-ROM versions. One example of how to cite the standard is given in [RFC2279]. Any standard that cites ISO/IEC 10646 needs to evaluate how to handle the versioning problem that is relevant to the protocol's needs.

ISO is responsible for other standards that might be of interest to protocol developers. [ISO 639] specifies the names of languages, and [ISO 3166] specifies the abbreviations of countries. Character work is done in the group known as ISO/IEC JTC1/SC22 and ISO TC46, as well as other ISO groups.

Another relevant ISO group is JTC 1/SC22/WG20, which is responsible for internationalization in JTC1, such as for international string ordering. Information on WG20, and its work products, can be found at <http://www.dkuug.dk/jtc1/sc22/wg20/>

Unicode Consortium

The second important group for international character standards is the Unicode Consortium. The Unicode Consortium is a trade association of companies, governments, and other groups interested in promoting the Unicode Standard [UNICODE]. The Unicode Standard is a CCS whose repertoire and code points are identical to ISO/IEC 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. Examples of these attributes include case conversion and numeric properties.

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/>.

World Wide Web Consortium (W3C)

This group created and maintains the standard for XML, the markup language for text that has become very popular. XML has always been fully internationalized so that there is no need for a new version to handle international text.

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), and CEN/ISSS (Europe).

3.2 Encodings and transformation formats of ISO/IEC 10646

Characters in the ISO/IEC 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.

Basic Multilingual Plane (BMP)

The BMP is composed of the first 2^16 code points in ISO/IEC 10646. The BMP is also called "plane 0".

UCS-2 and UCS-4

UCS-2 and UCS-4 are the two encoding forms defined for ISO/IEC 10646. UCS-2 addresses only the BMP. Because many useful characters (such as many Han characters) have been defined outside of the BMP, many people would consider UCS-2 to be dead. Theoretically, UCS-4 addresses the entire range of 2^31 code points from ISO/IEC 10646 as 32-bit values. However, for interoperability with UTF-16, ISO 10646 restricts the range of characters that will actually be allocated to the values 0..0x10FFFF.

UTF-8

UTF-8, a transformation format specified in [RFC2279], is the preferred encoding for IETF protocols. Characters in the BMP are encoded as one, two, or three octets. Characters outside the BMP are encoded as four octets. Characters from the US-ASCII repertoire have the same on-the-wire representation in UTF-8 as they do in US-ASCII.

UTF-16, UTF-16BE, and UTF-16LE

UTF-16, UTF-16BE, and UTF-16LE, three transformation formats defined in [RFC2781], are not required by any IETF standards, and are thus used much less often than UTF-8. Characters in the BMP are always encoded as two octets, and characters outside the BMP are encoded as four octets. The three formats differ based on the order of the octets and the presence of a special lead-in mark called the "byte order mark" or "BOM".

UTF-32

The Unicode Consortium has defined UTF-32 as a transformation format for UCS-4 in [UTR19].

SCSU and BOCU-1

The Unicode Consortium has defined an encoding, SCSU, which is designed to offer good compression for typical text. SCSU is described in [UTR6]. A different encoding that is meant to be MIME-friendly, BOCU-1, is described in [UTN6]. Although compression is attractive, as opposed to UTF-8 , neither of these (at the time of this writing) has attracted much interest in the IETF.

3.3 Native CCSs and charsets

Before ISO/IEC 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 for Cyrillic and Shift-JIS for Japanese scripts.

The official list of the registered charset names for use with IETF protocols is maintained by IANA and can be found at <http://www.iana.org/assignments/character-sets>. The list contains preferred names and aliases. Note that this list has historically contained many errors, such as names that are in fact not charsets or references that do not give enough detail to reliably map names to charsets.

Probably the most well-known native CCS is ASCII [US-ASCII]. This CCS is used as the basis for keywords and parameter names in many IETF protocols, and as the sole CCS in numerous IETF protocols that have not yet been internationalized.

[UTR22] describes issues involved in mapping character data between charsets, and an XML format for mapping table data.

4. Character Issues

This section contains terms and topics that are commonly used in character handling and therefore are of concern to people adding non-ASCII text handling to protocols. These topics are standardized outside the IETF.

combining character

A member of an identified subset of the coded character set of ISO/IEC 10646 intended for combination with the preceding non-combining graphic character, or with a sequence of combining characters preceded by a non-combining character. <ISOIEC10646>

composite sequence

A sequence of graphic characters consisting of a non-combining character followed by one or more combining characters. A graphic symbol for a composite sequence generally consists of the combination of the graphic symbols of each character in the sequence. A composite sequence is not a character and therefore is not a member of the repertoire of ISO/IEC 10646. <ISOIEC10646>

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", or it might be a combination of the three characters "a", a non-destructive backspace, and an acute. The rules for combining two or more characters are called "composition rules", and the rules for taking apart a character into other characters is called "decomposition rules". The results of composition is called a "precomposed character"; the results of decomposition is called a "decomposed character".

normalization

Normalization is the transformation of data to a normal form, for example, to unify spelling. <UNICODE>

Note that the phrase "unify spelling" in the definition above does not mean unifying different words with the same meaning (such as "color" and "colour"). Instead, it means unifying different character sequences that are intended to form the same composite characters (such as "<a><n><combining tilde><o>" and "<a><n with tilde><o>").

The purpose of normalization is to allow two strings to be compared for equivalence. The strings "<a><n><combining tilde><o>" and "<a><n with tilde><o>" would be shown identically on a text display device. If a protocol designer wants those two strings to be considered equivalent during comparison, the protocol must define where normalization occurs.

The terms "normalization" and "canonicalization" are often used interchangeably. Generally, they both mean to convert a string of one or more characters into another string based on standardized rules. Some CCSs allow multiple equivalent representations for a written string; normalization selects one among multiple equivalent representations as a base for reference purposes in comparing strings. In strings of 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.

case

Case is the feature of certain alphabets where the letters have two distinct forms. These variants, which may differ markedly in shape and size, are called the uppercase letter (also known as capital or majuscule) and the lowercase letter (also known as small or minuscule). Case mapping is the association of the uppercase and lowercase forms of a letter. <UNICODE>

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 or for which there is a special mapping rule, such as the Turkish dotless "i" and some Greek characters with modifiers. Case mapping can even be dependent on locale. Converting text to have only one case is called "case folding".

sorting and collation

Collating is the process of ordering units of textual information. Collation is usually specific to a particular language. It is sometimes known as alphabetizing, although alphabetization is just a special case of sorting and collation. <UNICODE>

Collation is concerned with the determination of the relative order of any particular pair of strings, and algorithms concerned with collation focus on the problem of providing appropriate weighted keys for string values, to enable binary comparison of the key values to determine the relative ordering of the strings.

Sorting is the process of actually putting data records into specified orders, according to criteria for comparison between the records. Sorting can apply to any kind of data (including textual data) for which an ordering criterion can be defined. Algorithms concerned with sorting focus on the problem of performance (in terms of time, memory, or other resources) in actually putting the data records into a specified order.

A sorting algorithm for string data can be internationalized by providing it with the appropriate collation-weighted keys corresponding to the strings to be ordered.

Many processes have a need to order strings in a consistent sequence (sorted). For only a few 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 sorting. That is, the codepoint order is also the order that a person would use in sorting the characters. For many CCS/CES combinations, the codepoint order would make no sense to a person and therefore is not useful for sorting if the results will be displayed to a person.

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 language-specific 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 language-specific sort.

code table

A code table is a table showing the characters allocated to the octets in a code. <ISOIEC10646>

Code tables are also commonly called "code charts".

4.1 Types of characters

The following definitions of types of characters do not clearly delineate each character into one type, nor do they allow someone to accurately predict what types would apply to a particular character. The definitions are intended for application designers to help them think about the many (sometimes confusing) properties of text.

alphabetic

An informative Unicode property. Characters that are the primary units of alphabets and/or syllabaries, whether combining or noncombining. This includes composite characters that are canonical equivalents to a combining character sequence of an alphabetic base character plus one or more combining characters: letter digraphs; contextual variant of alphabetic characters; ligatures of alphabetic characters; contextual variants of ligatures; modifier letters; letterlike symbols that are compatibility equivalents of single alphabetic letters; and miscellaneous letter elements. <UNICODE>

ideographic

Any symbol that primarily denotes an idea (or meaning) in contrast to a sound (or pronunciation), for example, a symbol showing a telephone or the Han characters used in Chinese, Japanese, and Korean. <UNICODE>

punctuation

Characters that separate units of text, such as sentences and phrases, thus clarifying the meaning of the text. The use of punctuation marks is not limited to prose; they are also used in mathematical and scientific formulae, for example. <UNICODE>

symbol

One of a set of characters other than those used for letters, digits, or punctuation, and representing various concepts generally not connected to written language use per se. Examples include symbols for mathematical operators, symbols for OCR, symbols for box-drawing or graphics, and symbols for dingbats. <NONE>

Examples of symbols include characters for arrows, faces, and geometric shapes. [UNICODE] has a property that defines characters as symbols.

nonspacing character

A combining character whose positioning in presentation is dependent on its base character. It generally does not consume space along the visual baseline in and of itself. <UNICODE>

A combining acute accent (U+0301) is an example of a nonspacing character.

diacritic

A mark applied or attached to a symbol to create a new symbol that represents a modified or new value. They can also be marks applied to a symbol irrespective of whether it changes the value of that symbol. In the latter case, the diacritic usually represents an independent value (for example, an accent, tone, or some other linguistic information). Also called diacritical mark or diacritical. <UNICODE>

control character

The 65 characters in the ranges U+0000..U+001F and U+007F..U+009F. They are also known as control codes. <UNICODE>

formatting character

Characters that are inherently invisible but that have an effect on the surrounding characters. <UNICODE>

Examples of formatting characters include characters for specifying the direction of text and characters that specify how to join multiple characters.

compatibility character

A graphic character included as a coded character of ISO/IEC 10646 primarily for compatibility with existing coded character sets. <ISOIEC10646>

For example, U+FF01 (FULLWIDTH EXCLAMATION MARK) was included for compatibility with Asian character sets that include full-width and half-width ASCII characters.

5. User interface for 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. Internationalization challenges assumptions about the type and limitations of the input and output devices that may be used with applications that use various protocols. It is therefore useful to consider how users typically interact with text that might contain one or more non-ASCII characters.

input methods

An input method is a mechanism for a person to enter text into an application. <NONE>

Text can be entered into a computer in many ways. Keyboards are by far the most common device used, but many characters cannot be entered on typical computer keyboards in a single stroke. 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 methods 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.

rendering rules

A rendering rule is an algorithm that a system uses to decide how to display a string of text. <NONE>

Some scripts can be directly displayed with fonts, where each character from an input stream can simply be copied from a glyph system and put on the screen or printed page. Other scripts need rules that are based on the context of the characters in order to render text for display.

Some examples of these rendering rules include:

- Scripts such as Arabic (and many others), where the form of the letter changes depending on the adjacent letters, 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. The rendering rules must choose between two or more glyphs.

- Scripts such as the Indic scripts, where consonants may change their form if they are adjacent to certain other consonants or may be displayed in an order different from the way they are stored and pronounced. The rendering rules must choose between two or more glyphs.

- Arabic and Hebrew scripts, where the order of the characters displayed are changed by the bidirectional properties of the alphabetic characters and with right-to-left and left-to-right ordering marks. The rendering rules must choose the order that characters are displayed.

graphic symbol

A graphic symbol is the visual representation of a graphic character or of a composite sequence. <ISOIEC10646>

font

A font is a collection of glyphs used for the visual depiction of character data. A font is often associated with a set of parameters (for example, size, posture, weight, and serifness), which, when set to particular values, generate a collection of imagable glyphs. <UNICODE>

bidirectional display

The process or result of mixing left-to-right oriented text and right-to-left oriented text in a single line is called bidirectional display. <UNICODE>

Most of the world’s written languages are displayed left-to-right. However, many widely-used written languages such as ones based on the Hebrew or Arabic scripts are displayed right-to-left. Right-to-left text often confuses 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. (Note that some languages are written both horizontally and vertically.)

Further, bidirectional text can cause confusion because there are formatting characters in ISO/IEC 10646 which cause the order of display of text to change. These explicit formatting characters change the display regardless of the implicit left-to-right or right-to-left properties of characters.

It is common to see strings with text in both directions, such as strings that include both text and numbers, or strings that contain a mixture of scripts.

[UNICODE] has a long and incredibly detailed algorithm for displaying bidirectional text.

undisplayable character

A character that has no displayable form. <NONE>

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. Note, however, that every character in [UNICODE] has a glyph associated with it, and that the glyphs for undisplayable characters are enclosed in a dashed square as an indication that the actual character is 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

Protocol elements are uniquely-named parts of a protocol. <NONE>

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

name spaces

A name space is the set of valid names for a particular item, or the syntactic rules for generating these valid names. <NONE>

Many items in Internet protocols use names to identify specific instances or values. The names may be generated (by some prescribed rules), registered centrally (e.g., such as with IANA), or have a distributed registration and control mechanism, such as the names in the DNS.

on-the-wire encoding

The encoding and decoding used before and after transmission over the network is often called the "on-the-wire" (or sometimes just "wire") format. <NONE>

Characters are identified by codepoints. Before being transmitted in a protocol, they must first be encoded as bits and octets. 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.

parsed text

Text strings that is analyzed for subparts. <NONE>

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 what appears after the "Re:" prefix.

charset identification

Specification of the charset used for a string of text. <NONE>

Protocols that allow more than one charset to be used in the same place should require that the text be identified with the appropriate charset. Without this identification, a program looking at the text cannot definitively discern the charset of the text. Charset identification is also called "charset tagging".

language identification

Specification of the human language used for a string of text. <NONE>

Some protocols (such as MIME and HTTP) allow text that is meant for machine processing to be identified with the language used in the text. Such identification is important for machine-processing of the text, such as by systems that render the text by speaking it. Language identification is also called "language tagging".

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, [RFC2822], the standard for Internet mail headers). MIME is described in RFCs 2045 through 2049, as well as more recent RFCs. <NONE>

transfer encoding syntax

A transfer encoding syntax (TES) (sometimes called a transfer encoding scheme) is a reversible transform of already-encoded data that is represented in one or more character encoding schemes. <NONE>

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. The main examples of TESs used in the IETF include Base64 and quoted-printable.

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 defined in [RFC2045]. <NONE>

quoted printable

Quoted printable is a transfer encoding syntax that allows strings that have non-ASCII characters mixed in with mostly ASCII printable characters to be somewhat human readable. It is described in [RFC2047]. <NONE>

The quoted printable syntax is generally considered to be a failure at being readable. It is jokingly referred to as "quoted unreadable".

XML

XML (which is an approximate abbreviation for Extensible Markup Language) is a popular method for structuring text. XML text is explicitly tagged with charsets. The specification for XML can be found at <http://www.w3.org/XML/>. <NONE>

ASN.1 text formats

The ASN.1 data description language has many formats for text data. 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/IEC 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 (ACE)

Starting in 1996, 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/IEC 10646 characters as legal DNS host names (as described in STD 13). At the time of this writing, no 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.

locale

Locale is the user-specific location and cultural information managed by a computer. <NONE>

Because languages differ from country to country (and even region to region within a country), the locale of the user 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. Some locales (especially the popular "C" and "POSIX" locales) do not include language information.

It should be noted that there are many thorny, unsolved issues with locale. For example, should text be viewed using the locale information of the person who wrote the text or the person viewing it? What if the person viewing it is travelling to different locations? Should only some of the locale information affect creation and editing of text?

Latin characters

"Latin characters" is a not-precise term for characters historically related to ancient Greek script and currently used throughout the world. <NONE>

The base Latin characters make up the ASCII repertoire and have been augmented by many single and multiple diacritics and quite a few other characters. ISO/IEC 10646 encodes the Latin characters in the ranges U+0020..U+024F, U+1E00..U+1EFF, and other ranges.

romanization

The transliteration of a non-Latin script into Latin characters. <NONE>

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. For example, there are two popular romanizations of Chinese: Wade-Giles and Pinyin, the latter of which is by far more common today. Many romanization systems are inexact and do not give perfect round trip mappings between the native script and the Latin characters.

CJK characters and Han characters

The ideographic characters used in Chinese, Japanese, Korean, and traditional Vietnamese writing systems are often called 'CJK characters' after the initial letters of the language names in English. They are also called "Han characters", after the term in Chinese that is often used for these characters. <NONE>

Note that CJK and Han characters do not include the phonetic characters used in the Japanese and Korean languages.

In ISO/IEC 10646, the Han characters were "unified", meaning that each set of Han characters from Japanese, Chinese, and/or Korean that had the same origin was assigned a single code point. The positive result of this was that many fewer code points were needed to represent Han; the negative result of this was that characters that people who write the three languages think are different have the same code point. There is a great deal of disagreement on the nature, the origin, and the severity of the problems caused by Han unification.

translation

The process of conveying the meaning of some passage of text in one language, so that it can be expressed equivalently in another language. <NONE>

Many language translation systems are inexact and cannot be applied repeatedly to go from one language to another to another.

transliteration

The process of representing the characters of an alphabetical or syllabic system of writing by the characters of a conversion alphabet. <NONE>

Many script transliterations are exact, and many have perfect round-trip mappings. The notable exception to this is romanization, described above. Transliteration involves converting text expressed in one script into another script, generally on a letter-by-letter basis.

transcription

The process of systematically writing the sounds of some passage of spoken language, generally with the use of a technical phonetic alphabet (usually Latin-based) or other systematic transcriptional orthography. Transcription also sometimes refers to the conversion of written text into a transcribed (usually Latin-based) form, based on the sound of the text as if it had been spoken. <NONE>

Unlike transliterations, which are generally designed to be round-trip convertible, transcriptions of written material are almost never round-trip convertible to their original form.

regular expressions

Regular expressions provide a mechanism to select specific strings from a set of character strings. Regular expressions are a language used to search for text within strings, and possibly modify the text found with other text. <NONE>

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. There is no single regular expression language, but there are numerous very similar dialects.

The Unicode Consortium has a good discussion about how to adapt regular expression engines to use Unicode. [UTR18]

private use

ISO/IEC 10646 code points from U+E000 to U+F8FF, U+F0000 to U+FFFFD, and U+100000 to U+10FFFD are available for private use. This refers to code points of the standard whose interpretation is not specified by the standard and whose use may be determined by private agreement among cooperating users. <UNICODE>

The use of these "private use" characters is defined by the parties who transmit and receive them, and is thus not appropriate for standardization. (The IETF has a long history of private use names for things such as "x-" names in MIME types, charsets, and languages. The experience with these has been quite negative, with many implementors assuming that private use names are in fact public and long-lived.)

8. Security Considerations

Security is not discussed in this document.

9. References

9.1 Normative References

[ISOIEC10646] ISO/IEC 10646-1:2000. International Standard -- Information technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane, 2000.

[UNICODE] 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/), The Unicode Consortium, 2002.

9.2 Informative References

[CHARMOD] Character Model for the World Wide Web 1.0, W3C, <http://www.w3.org/TR/charmod/>.

[FRAMEWORK] ISO/IEC TR 11017:1997(E). Information technology - Framework for internationalization, prepared by ISO/IEC JTC 1/SC 22/WG 20, 1997.

[ISO 639] ISO 639:2000 (E/F) - Code for the representation of names of languages, 2000.

[ISO 3166] ISO 3166:1988 (E/F) - Codes for the representation of names of countries, 2000.

[RFC2045] Freed, N. and N. Borenstein, "MIME Part One: Format of Internet Message Bodies", November 1996.

[RFC2047] Moore, K., "MIME Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.

[RFC2277] Alvestrand, H., "IETF Policy on Character Sets and Languages", BCP 18, RFC 2277, January 1998.

[RFC2279] Yergeau, F., "UTF-8, a transformation format of ISO 10646", RFC 2279, January 1998.

[RFC2781] Hoffman, P. and F. Yergeau, "UTF-16, an encoding of ISO 10646", RFC 2781, February 2000.

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

[RFC3066] Alvestrand, H., "Tags for the Identification of Languages", BCP 47, RFC 3066, January 2001.

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

[UTN6] "BOCU-1: MIME-Compatible Unicode Compression", M. Scherer & M. Davis, Unicode Technical Note #6.

[UTR6] "A Standard Compression Scheme for Unicode", M. Wolf, et. al., Unicode Technical Report #6.

[UTR15] "Unicode Normalization Forms", M. Davis & M. Duerst, Unicode Technical Report #15.

[UTR18] "Unicode Regular Expression Guidelines", M. Davis, Unicode Technical Report #18.

[UTR19] "UTF-32", M. Davis, Unicode Technical Report #19.

[UTR22] "Character Mapping Markup Language", M. Davis, Unicode Technical Report #22.

10. Additional Interesting Reading

ALA-LC Romanization Tables, Randall Barry (ed.), U.S. Library of Congress, 1997, ISBN 0844409405

Blackwell Encyclopedia of Writing Systems, Florian Coulmas, Blackwell Publishers, 1999, ISBN 063121481X

The World’s Writing Systems, Peter Daniels and William Bright, Oxford University Press, 1996, ISBN 0195079930

Writing Systems of the World, Akira Nakanishi, Charles E. Tuttle Company, 1980, ISBN 0804816549

11. Index

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. Harald Alvestrand and Martin Duerst made extensive useful comments on early versions. Others who contributed to the development include:

Dan Kohn
Jacob Palme
Johan van Wingen
Peter Constable
Yuri Demchenko
Susan Harris
Zita Wenzel
John Klensin
Henning Schulzrinne
Leslie Daigle
Markus Scherer
Ken Whistler

B. Author’s Address

Paul Hoffman
Internet Mail Consortium and VPN Consortium
127 Segre Place
Santa Cruz, CA 95060 USA

EMail: paul.hoffman@imc.org and paul.hoffman@vpnc.org

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.