Internet Engineering Task Force (IETF)
Request for Comments: 6365
BCP: 166
Obsoletes: 3536
Category: Best Current Practice
ISSN: 2070-1721

P. Hoffman
VPN Consortium
J. Klensin

September 2011

Terminology Used in Internationalization in the IETF

Abstract

This document provides a list 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.

Status of This Memo

This memo documents an Internet Best Current Practice.

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 BCPs is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6365.

Table of Contents

1. Introduction

As the IETF Character Set Policy specification [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. Subject only to the limitations of their own knowledge and facilities, it should be possible for anyone to enter or read these text strings, which means that Internet users must be able to enter text using typical input methods and have it be 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 terminology as it applies to IETF standards work by lightly covering the many aspects of internationalization and the vocabulary associated with those topics. Some of the overview is somewhat tutorial in nature. It is not meant to be a complete description of internationalization. The definitions here SHOULD be used by IETF standards. IETF standards that explicitly want to create different definitions for the terms defined here can do so, but unless an alternate definition is provided the definitions of the terms in this document apply. IETF standards that have a requirement for different definitions are encouraged, for clarity’s sake, to find terms different than the ones defined here. Some of the definitions in this document come from 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 inside and outside the IETF. The primary documents used are:

- ISO/IEC 10646 [ISOIEC10646]
- The Unicode Standard [UNICODE]
- W3C Character Model [CHARMOD]
- IETF RFCs, including the Character Set Policy specification [RFC2277] and the domain name internationalization standard [RFC5890]

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 "<RFC6365>", 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 the reference sections of this document.

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 an example 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". See RFC 5137 [RFC5137] for a description of this notation.

1.3. Normative Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 (see Section 4.2) and with other aspects of internationalization.

language

A language is a way that humans communicate. The use of language occurs in many forms, the most common of which are speech, writing, and signing. <RFC6365>

Some languages have a close relationship between the written and spoken forms, while others have a looser relationship. The so-called LTRU (Language Tag Registry Update) standards [RFC5646] [RFC4647] discuss languages in more detail and provide 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 characters, often called ideographs after a subset of them, used in writing Chinese, Japanese, and Korean). RFC 2277 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 to write more than one language; for example, the Russian and Bulgarian languages are written in the Cyrillic script. Some languages can be expressed using different scripts or were used with different scripts at different times; the Mongolian language can be written in either the Mongolian or Cyrillic scripts; Malay is primarily written in Latin script today, but the earlier, Arabic-script-based, Jawa form is still in use; and a number of languages were converted

from other scripts to Cyrillic in the first half of the last century, some of which have switched again more recently. 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.

writing system

A set of rules for using one or more scripts to write a particular language. Examples include the American English writing system, the British English writing system, the French writing system, and the Japanese writing system. <UNICODE>

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, but generalized to include digits and symbols of various sorts

* the encoded entity itself

When people talk about characters, they usually intend one of the first two definitions. The term "character" is often abbreviated as "char".

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 coded character set (CCS) 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 image of a character that can be displayed after being imaged onto a display surface. <RFC6365>

The Unicode Standard has a different definition that refers to an abstract form that may represent different images when the same character is rendered under different circumstances.

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 unordered: 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-16BE. <UNICODE>

Some CESs are associated with a single CCS; for example, UTF-8 [RFC3629] applies only to the identical CCSs of ISO/IEC 10646 and Unicode. 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 [RFC2978]. <RFC6365>

Many protocol definitions use the term "character set" in their descriptions. The terms "charset", or "character encoding scheme" and "coded character set", are strongly preferred over the term "character set" because "character set" has other definitions in other contexts, particularly outside the IETF. When reading IETF standards that use "character set" without defining the term, they usually mean "a specific combination of one CCS with a CES", particularly when they are talking about the "US-ASCII character set".

internationalization

In the IETF, "internationalization" means to add or improve the handling of non-ASCII text in a protocol. <RFC6365> A different perspective, more appropriate to protocols that are designed for global use from the beginning, is the definition used by W3C:

> "Internationalization is the design and development of a product, application or document content that enables easy localization for target audiences that vary in culture, region, or language." [W3C-i18n-Def]

Many protocols that handle text only handle one charset (US-ASCII), or leave the question of what CCS and encoding are used up to local guesswork (which leads, of course, to interoperability problems). If multiple charsets are permitted, they must be explicitly identified [RFC2277]. Adding non-ASCII text to a protocol allows the protocol to handle more scripts, hopefully all of the ones useful in the world. In today’s world,

that is normally best accomplished by allowing Unicode encoded in UTF-8 only, thereby shifting conversion issues away from individual choices.

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 8 of this document.

i18n, l10n

These are abbreviations for "internationalization" and "localization". <RFC6365>

"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. <RFC6365>

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. <RFC6365>

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 straightforward, 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 and ISO/IEC JTC 1

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. Most of ISO's work in information technology is performed jointly with a similar body, the International Electrotechnical Commission (IEC) through a joint committee known as "JTC 1". ISO and ISO/IEC JTC 1 have 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", sometimes with a specific date. 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 "SC2/WG2" or "WG2" for short. ISO standards go through many steps before being finished, and years often go by between changes to the base ISO/IEC 10646 standard although amendments are now issued to track Unicode changes. Information on WG2, and its work products, can be found at <http://www.dkuug.dk/JTC1/SC2/WG2/>. Information on SC2, and its work products, can be found at <http://www.iso.org/iso/standards_development/technical_committees/list_of_iso_technical_committees/iso_technical_committee.htm?commid=45050>

The standard comes as a base part and a series of attachments or amendments. It is available in PDF form for downloading or in a CD-ROM version. One example of how to cite the standard is given in [RFC3629]. 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 concerned about internationalization. ISO 639 [ISO639] specifies the names of languages and forms part of the basis for the IETF's Language Tag work [RFC5646]. ISO 3166 [ISO3166] specifies the names and code abbreviations for countries and territories and is used in several protocols and databases including names for country-code top level domain names. The responsibilities of ISO TC 46 on Information and Documentation <http://www.iso.org/iso/standards_development/technical_committees/list_of_iso_technical_committees/iso_technical_committee.htm?commid=48750> include a series of standards for transliteration of various languages into Latin characters.

Another relevant ISO group was JTC 1/SC22/WG20, which was responsible for internationalization in JTC 1, such as for international string ordering. Information on WG20, and its work products, can be found at <http://www.dkuug.dk/jtc1/sc22/wg20/>. The specific tasks of SC22/WG20 were moved from SC22 into SC2, and there has been little significant activity since that occurred.

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 that make it more useful in protocols, such as defining attributes for each character. Examples of these attributes include case conversion and numeric properties.

The actual technical and definitional work of the Unicode Consortium is done in the Unicode Technical Committee (UTC). The terms "UTC" and "Unicode Consortium" are often treated, imprecisely, as synonymous in the IETF.

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/>.

A reciprocal agreement between the Unicode Consortium and ISO/IEC JTC 1/SC 2 provides for ISO/IEC 10646 and The Unicode Standard to track each other for definitions of characters and assignments of code points. Updates, often in the form of amendments, to the former sometimes lag updates to the latter for a short period, but the gap has rarely been significant in recent years.

At the time that the IETF character set policy [RFC2277] was established and the first version of this terminology specification was published, there was a strong preference in the IETF community for references to ISO/IEC 10646 (rather than Unicode) when possible. That preference largely reflected a more general IETF preference for referencing established open international standards over specifications from consortia. However, the Unicode definitions of character properties and classes are not part of ISO/IEC 10646. Because IETF specifications are increasingly dependent on those definitions

(for example, see the explanation in Section 4.2) and the Unicode specifications are freely available online in convenient machine-readable form, the IETF's preference has shifted to referencing the Unicode Standard. The latter is especially important when version consistency between code points (either standard) and Unicode properties (Unicode only) is required.

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. However, in some circumstances, XML files may be sensitive to differences among Unicode versions.

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), CEN/ISSS (Europe), JIS (Japan), and SAC (China).

3.2. Encodings and Transformation Formats of ISO/IEC 10646

Characters in the ISO/IEC 10646 CCS can be expressed in many ways. Historically, "encoding forms" are both direct addressing methods, while "transformation formats" are methods for expressing encoding forms as bits on the wire. That distinction has mostly disappeared in recent years.

Documents that discuss characters in the ISO/IEC 10646 CCS often need to list specific characters. RFC 5137 describes the common methods for doing so in IETF documents, and these practices have been adopted by many other communities as well.

Basic Multilingual Plane (BMP)

The BMP is composed of the first 2^16 code points in ISO/IEC 10646 and contains almost all characters in contemporary use. The BMP is also called "Plane 0".

UCS-2 and UCS-4

UCS-2 and UCS-4 are the two encoding forms historically 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 consider UCS-2 to be obsolete.

UCS-4 addresses the entire range of code points from ISO/IEC 10646 (by agreement between ISO/IEC JTC 1 SC2 and the Unicode Consortium, a range from 0..0x10FFFF) as 32-bit values with zero padding to the left. UCS-4 is identical to UTF-32BE (without use of a BOM (see below)); UTF-32BE is now the preferred term.

UTF-8

UTF-8 [RFC3629] 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. The IETF-specific definition of UTF-8 in RFC 3629 is identical to that in recent versions of the Unicode Standard (e.g., in Section 3.9 of Version 6.0 [UNICODE]).

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

UTF-16, UTF-16BE, and UTF-16LE, three transformation formats described in [RFC2781] and defined in The Unicode Standard (Sections 3.9 and 16.8 of Version 6.0), are not required by any IETF standards, and are thus used much less often in protocols than UTF-8. Characters in the BMP are always encoded as two octets, and characters outside the BMP are encoded as four octets using a "surrogate pair" arrangement. The latter is not part of UCS-2, marking the difference between UTF-16 and UCS-2. The three UTF-16 formats differ based on the order of the octets and the presence or absence of a special lead-in ordering identifier called the "byte order mark" or "BOM".

UTF-32

The Unicode Consortium and ISO/IEC JTC 1 have defined UTF-32 as a transformation format that incorporates the integer code point value right-justified in a 32-bit field. As with UTF-16, the byte order mark (BOM) can be used and UTF-32BE and UTF-32LE are defined. UTF-32 and UCS-4 are essentially equivalent and the terms are often used interchangeably.

SCSU and BOCU-1

The Unicode Consortium has defined an encoding, SCSU [UTR6], which is designed to offer good compression for typical text. 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.

The compression provided as a side effect of the Punycode algorithm [RFC3492] is heavily used in some contexts, especially IDNA [RFC5890], but imposes some restrictions. (See also Section 7.)

3.3. Native CCSs and Charsets

Before ISO/IEC 10646 was developed, many countries developed their own CCSs and charsets. Some of these were adopted into international standards for the relevant scripts or writing systems. 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. ASCII became the basis for ISO/IEC 646 which, in turn, formed the basis for many national and international standards, such as the ISO 8859 series, that mix Basic Latin characters with characters from another script.

It is important to note that, strictly speaking, "ASCII" is a CCS and repertoire, not an encoding. The encoding used for ASCII in IETF protocols involves the 7-bit integer ASCII code point right-justified in an 8-bit field and is sometimes described as the "Network Virtual Terminal" or "NVT" encoding [RFC5198]. Less formally, "ASCII" and "NVT" are often used interchangeably. However, "non-ASCII" refers only to characters outside the ASCII repertoire and is not linked to a specific encoding. See Section 4.2.

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

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.

code point

A value in the codespace of a repertoire. For all common repertoires developed in recent years, code point values are integers (code points for ASCII and its immediate descendants were defined in terms of column and row positions of a table).

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. Combining characters are inherently non-spacing. <ISOIEC10646>

composite sequence or combining character 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. The Unicode Standard often uses the term "combining character sequence" to refer to composite sequences. A composite sequence is not a character and therefore is not a member of the repertoire of ISO/IEC 10646. <ISOIEC10646> However, Unicode now assigns names to some such sequences especially when the names are required to match terminology in other standards [UAX34].

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. In the same or other CCSs, it might be available as a single code point. The rules for combining two or more characters are called "composition rules", and the rules for taking apart a character into other characters are called "decomposition rules". The result of decomposition is called a "decomposed character"; the result of composition is usually a "precomposed 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 strings with the same meaning as words (such as "color" and "colour"). Instead, it means unifying different character sequences that are intended to form the same composite characters, such as "<n><combining tilde>" and "<n with tilde>" (where "<n>" is U+006E, "<combining tilde>" is U+0303, and "<n with tilde>" is U+00F1).

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. However, in Unicode, "canonicalization" or similar terms are used to refer to a particular type of normalization equivalence ("canonical equivalence" in contrast to "compatibility equivalence"), so the term should be used with some care. 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. Unicode Standard Annex #15 [UTR15] describes the process and many forms of normalization in detail. Normalization is important when comparing strings to see if they are the same.

The Unicode NFC and NFD normalizations support canonical equivalence; NFKC and NFKD support canonical and compatibility equivalence.

case

Case is the feature of certain alphabets where the letters have two (or occasionally more) distinct forms. These forms, 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 that 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", some Greek characters with modifiers, and characters like the German Sharp S (Eszett) and Greek Final Sigma that traditionally do not have uppercase forms. Case mapping can even be dependent on locale or language. Converting text to have only a single case, primarily for comparison purposes, is called "case folding". Because of the various unusual cases, case mapping can be quite controversial and some case folding algorithms even more so. For example, some programming languages such as Java have case-folding algorithms that are locale-sensitive; this makes those algorithms incredibly resource-intensive and makes them act differently depending on the location of the system at the time the algorithm is used.

sorting and collation

Collating is the process of ordering units of textual information. Collation is usually specific to a particular language or even to a particular application or locale. 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.

The relative orders of letters in collation sequences can differ widely based on the needs of the system or protocol defining the collation order. For example, even within ASCII characters, there are two common and very different collation orders: "A, a, B, b,..." and "A, B, C, ..., Z, a, b,...", with additional variations for lowercase first and digits before and after letters.

In practice, it is rarely necessary to define a collation sequence for characters drawn from different scripts, but arranging such sequences so as to not surprise users is usually particularly problematic.

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 the desired 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 (sorted) sequence. For only a few CCS/CES combinations, there is an obvious sort order that can be applied without reference to the linguistic meaning of the characters: the code point order is sufficient for sorting. That is, the code point order is also the order that a person would use in sorting the characters. For many CCS/CES combinations, the code point order would make no sense to a person and therefore is not useful for sorting if the results will be displayed to a person.

Code point 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" or "human language" sort. Sorting to code point 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 non-combining. 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 variants 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>

While Unicode and many other systems use this term to refer to all Han characters, strictly speaking not all of those characters are actually ideographic. Some are pictographic (such as the telephone example above), some are used phonetically, and so on. However, the convention is to describe the script as ideographic as contrasted to alphabetic.

digit or number

All modern writing systems use decimal digits in some form; some older ones use non-positional or other systems. Different scripts may have their own digits. Unicode distinguishes between numbers and other kinds of characters by assigning a special General Category value to them and subdividing that value to distinguish between decimal digits, letter digits, and other digits. <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. <RFC6365>

Examples of symbols include characters for mathematical operators, symbols for optical character recognition (OCR), symbols for box-drawing or graphics, as well as symbols for dingbats, arrows, faces, and geometric shapes. Unicode has a property that identifies symbol characters.

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 they change 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. The basic space character, U+0020, is often considered as a control character as well, making the total number 66. They are also known as control codes. In terminology adopted by Unicode from ASCII and the ISO 8859 standards, these codes are treated as belonging to three ranges: "C0" (for U+0000..U+001F), "C1" (for U+0080...U+009F), and the single control character "DEL" (U+007F). <UNICODE>

Occasionally, in other vocabularies, the term "control character" is used to describe any character that does not normally have an associated glyph; it is also sometimes used for device control sequences [ISO6429]. Neither of those usages is appropriate to internationalization terminology in the IETF.

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 or compatibility variant

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

The Unicode definition of compatibility charter also includes characters that have been incorporated for other reasons. Their list includes several separate groups of characters included for compatibility purposes: halfwidth and fullwidth characters used with East Asian scripts, Arabic contextual forms (e.g., initial or final forms), some ligatures, deprecated formatting characters, variant forms of characters (or even copies of them) for particular uses (e.g., phonetic or mathematical applications), font variations, CJK compatibility ideographs, and so on. For additional information and the separate term "compatibility decomposable character", see the Unicode standard.

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

Some efforts in the IETF have concluded that it would be useful to support mapping of some groups of compatibility equivalents and not others (e.g., supporting or mapping width variations while preserving or rejecting mathematical variations). See the IDNA Mapping document [RFC5895] for one example.

4.2. Differentiation of Subsets

Especially as existing IETF standards are internationalized, it is necessary to describe collections of characters including especially various subsets of Unicode. Because Unicode includes ways to code substantially all characters in contemporary use, subsets of the Unicode repertoire can be a useful tool for defining these collections as repertoires independent of specific Unicode coding.

However specific collections are defined, it is important to remember that, while older CCSs such as ASCII and the ISO 8859 family are close-ended and fixed, Unicode is open-ended, with new character definitions, and often new scripts, being added every year or so. So, while, e.g., an ASCII subset, such as "uppercase letters", can be specified as a range of code points (4/1 to 5/10 for that example), similar definitions for Unicode either have to be specified in terms of Unicode properties or are very dependent on Unicode versions (and the relevant version must be identified in any specification). See the IDNA code point specification [RFC5892] for an example of specification by combinations of properties.

Some terms are commonly used in the IETF to define character ranges and subsets. Some of these are imprecise and can cause confusion if not used carefully.

non-ASCII

> The term "non-ASCII" strictly refers to characters other than those that appear in the ASCII repertoire, independent of the CCS or encoding used for them. In practice, if a repertoire such as that of Unicode is established as context, "non-ASCII" refers to characters in that repertoire that do not appear in the ASCII repertoire. "Outside the ASCII repertoire" and "outside the ASCII range" are practical, and more precise, synonyms for "non-ASCII".

letters

> The term "letters" does not have an exact equivalent in the Unicode standard. Letters are generally characters that are used to write words, but that means very different things in different languages and cultures.

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. <RFC6365>

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 or Vietnamese characters that have two or three diacritics on a single alphabetic character.

The term "input method editor" (IME) is often used generically to describe the tools and software used to deal with input of characters on a particular system.

rendering rules

A rendering rule is an algorithm that a system uses to decide how to display a string of text. <RFC6365>

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 and other characters and with right-to-left and left-to-right ordering marks. The rendering rules must choose the order that characters are displayed.

* Some writing systems cannot have their rendering rules suitably defined using mechanisms that are now defined in the Unicode Standard. None of those languages are in active non-scholarly use today.

* Many systems use a special rendering rule when they lack a font or other mechanism for rendering a particular character correctly. That rule typically involves substitution of a small open box or a question mark for the missing character. See "undisplayable character" below.

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, generates a collection of imagable glyphs. <UNICODE>

The term "font" is often used interchangeably with "typeface". As historically used in typography, a typeface is a family of one or more fonts that share a common general design. For example, "Times Roman" is actually a typeface, with a collection of fonts

such as "Times Roman Bold", "Times Roman Medium", "Times Roman Italic", and so on. Some sources even consider different type sizes within a typeface to be different fonts. While those distinctions are rarely important for internationalization purposes, there are exceptions. Those writing specifications should be very careful about definitions in cases in which the exceptions might lead to ambiguity.

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, often abbreviated as "bidi". <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 primarily right-to-left (numerals are a common exception in the modern scripts). 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, an order that might be different from what they see on the screen. (Note that some languages are written both horizontally and vertically and that some historical ones use other display orderings.)

Further, bidirectional text can cause confusion because there are formatting characters in ISO/IEC 10646 that 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. Text that might contain those characters typically requires careful processing before being sorted or compared for equality.

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 [UAX9].

undisplayable character

A character that has no displayable form. <RFC6365>

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.

The property of a character that causes it to be undisplayable is intrinsic to its definition. Undisplayable characters can never be displayed in normal text (the dashed square notation is used only in special circumstances). Printable characters whose Unicode definitions are associated with glyphs that cannot be rendered on a particular system are not, in this sense, undisplayable.

writing style

Conventions of writing the same script in different styles. <RFC6365>

Different communities using the script may find text in different writing styles difficult to read and possibly unintelligible. For example, the Perso-Arabic Nastalique writing style and the Arabic Naskh writing style both use the Arabic script but have very different renderings and are not mutually comprehensible. Writing styles may have significant impact on internationalization; for example, the Nastalique writing style requires significantly more line height than Naskh writing style.

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. <RFC6365>

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, and it is rarely appropriate to internationalize protocol element names (even while the elements themselves can be internationalized).

name spaces

A name space is the set of valid names for a particular item, or the syntactic rules for generating these valid names. <RFC6365>

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. <RFC6365>

Characters are identified by code points. 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 have been analyzed for subparts. <RFC6365>

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

Such conventions are very sensitive to localization. If, for example, a form like "Re:" is altered by an MUA to reflect the language of the sender or recipient, a system that subsequently does threading may not recognize the replacement term as a delimiter string.

charset identification

Specification of the charset used for a string of text. <RFC6365>

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. <RFC6365>

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". The IETF "LTRU" standards [RFC5646] and [RFC4647] provide a comprehensive model for language identification.

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 5322, the standard for Internet mail headers [RFC5322]). MIME is described in RFCs 2045 through 2049, as well as more recent RFCs. <RFC6365>

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. <RFC6365>

TESs are useful for encoding types of character data into 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. MIME identifies the transfer encoding syntax for body parts as a Content-transfer-encoding, occasionally abbreviated C-T-E.

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]. <RFC6365>

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]. <RFC6365>

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 that is not encoded as UTF-8 is explicitly tagged with charsets, and all text in XML consists only of Unicode characters. The specification for XML can be found at <http://www.w3.org/XML/>. <RFC6365>

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).

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 and some other purposes. Their goal is to be able to encode any string of ISO/IEC 10646 characters using the preferred syntax for domain names (as described in STD 13). At the time of this writing, only the ACE produced by Punycode [RFC3492] has become an IETF standard.

The choice of ACE forms to internationalize legacy protocols must be made with care as it can cause some difficult side effects [RFC6055].

LDH label

The classical label form used in the DNS and most applications that call on it, albeit with some additional restrictions, reflects the early syntax of "hostnames" [RFC0952] and limits those names to ASCII letters, digits, and embedded hyphens. The hostname syntax is identical to that described as the "preferred name syntax" in Section 3.5 of RFC 1034 [RFC1034] as modified by

RFC 1123 [RFC1123]. LDH labels are defined in a more restrictive and precise way for internationalization contexts as part of the IDNA2008 specification [RFC5890].

7. Terms Associated with Internationalized Domain Names

7.1. IDNA Terminology

The current specification for Internationalized Domain Names (IDNs), known formally as Internationalized Domain Names for Applications or IDNA, is referred to in the IETF and parts of the broader community as "IDNA2008" and consists of several documents. Section 2.3 of the first of those documents, commonly known as "IDNA2008 Definitions" [RFC5890] provides definitions and introduces some specialized terms for differentiating among types of DNS labels in an IDN context. Those terms are listed in the table below; see RFC 5890 for the specific definitions if needed.

ACE Prefix
A-label
Domain Name Slot
IDNA-valid string
Internationalized Domain Name (IDN)
Internationalized Label
LDH Label
Non-Reserved LDH label (NR-LDH label)
U-label

Two additional terms entered the IETF's vocabulary as part of the earlier IDN effort [RFC3490] (IDNA2003):

Stringprep

Stringprep [RFC3454] provides a model and character tables for preparing and handling internationalized strings. It was used in the original IDN specification (IDNA2003) via a profile called "Nameprep" [RFC3491]. It is no longer in use in IDNA, but continues to be used in profiles by a number of other protocols. <RFC6365>

Punycode

This is the name of the algorithm [RFC3492] used to convert otherwise-valid IDN labels from native-character strings expressed in Unicode to an ASCII-compatible encoding (ACE). Strictly speaking, the term applies to the algorithm only. In practice, it is widely, if erroneously, used to refer to strings that the algorithm encodes.

7.2. Character Relationships and Variants

The term "variant" was introduced into the IETF i18n vocabulary with the JET recommendations [RFC3743]. As used there, it referred strictly to the relationship between Traditional Chinese characters and their Simplified equivalents. The JET recommendations provided a model for identifying these pairs of characters and labels that used them. Specific recommendations for variant handling for the Chinese language were provided in a follow-up document [RFC4713].

In more recent years, the term has also been used to describe other collections of characters or strings that might be perceived as equivalent. Those collections have involved one or more of several categories of characters and labels containing them including:

- "visually similar" or "visually confusable" characters. These may be limited to characters in different scripts, characters in a single script, or both, and may be those that can appear to be alike even when high-distinguishability reference fonts are used or under various circumstances that may involve malicious choices of typefaces or other ways to trick user perception. Trivial examples include ASCII "l" and "1" and Latin and Cyrillic "a".

- Characters assigned more than one Unicode code point because of some special property. These characters may be considered "the same" for some purposes and different for others (or by other users). One of the most commonly cited examples is the Arabic YEH, which is encoded more than once because some of its shapes are different across different languages. Another example are the Greek lowercase sigma and final sigma: if the latter were viewed purely as a positional presentation variation on the former, it should not have been assigned a separate code point.

- Numerals and labels including them. Unlike letters, the "meaning" of decimal digits is clear and unambiguous regardless of the script with which they are associated. Some scripts are routinely used almost interchangeably with European digits and digits native to that script. The Arabic script has two sets of digits (U+0660..U+0669 and U+06F0..U=06F9), written identically for zero through three and seven through nine but differently for four through six; European digits predominate in other areas. Substitution of digits with the same numeric value in labels may give rise to another type of variant.

- Orthographic differences within a language. Many languages have alternate choices of spellings or spellings that differ by locale. Users of those languages generally recognize the spellings as equivalent, at least as much so as the variations described above.

Examples include "color" and "colour" in English, German words spelled with o-umlaut or "oe", and so on. Some of these relationships may also create other types of language-specific perceived differences that do not exist for other languages using the same script. For example, in Arabic language usage at the end of words, ARABIC LETTER TEH MARBUTA (U+0629) and ARABIC LETTER HEH (U+0647) are differently shaped (one has 2 dots in top of it), but they are used interchangeably in writing: they "sound" similar when pronounced at the end of phrase, and hence the LETTER TEH MARBUTA sometimes is written as LETTER HEH and the two are considered "confusable" in that context.

The term "variant" as used in this section should also not be confused with other uses of the term in this document or in Unicode terminology (e.g., those in Section 4.1 above). If the term is to be used at all, context should clearly distinguish among these different uses and, in particular, between variant characters and variant labels. Local text should identify which meaning, or combination of meanings, are intended.

8. Other Common Terms in Internationalization

This is a hodge-podge of other terms that have appeared in internationalization discussions in the IETF.

locale

Locale is the user-specific location and cultural information managed by a computer. <RFC6365>

Because languages and orthographic conventions 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, information that would apply to the location of the system storing or providing the text, or the person viewing it? What if the person viewing it is traveling 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 as modified in the Roman Republic and Empire and currently used throughout the world. <RFC6365>

The base Latin characters are a subset of 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 including ranges U+0020..U+024F and U+1E00..U+1EFF.

Because "Latin characters" is used in different contexts to refer to the letters from the ASCII repertoire, the subset of those characters used late in the Roman Republic period, or the different subset used to write Latin in medieval times, the entire ASCII repertoire, all of the code points in the extended Latin script as defined by Unicode, and other collections, the term should be avoided in IETF specifications when possible. Similarly, "Basic Latin" should not be used as a synonym for "ASCII".

romanization

The transliteration of a non-Latin script into Latin characters. <RFC6365>

Because of their widespread use, Latin characters (or graphemes constructed from them) are often used to try to write text in languages that didn't previously have writing systems or whose writing systems were originally based on different scripts. 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. <RFC6365>

Note that Han characters do not include the phonetic characters used in the Japanese and Korean languages. Users of the term "CJK characters" may or may not assume those additional characters are included.

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. <RFC6365>

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. <RFC6365>

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. There are many official and unofficial transliteration standards, most notably those from ISO TC 46 and the U.S. Library of Congress.

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 form, based on the sound of the text as if it had been spoken. <RFC6365>

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, at least without some supplemental information.

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. <RFC6365>

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 that are not quite consistent with each other.

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

private use character

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. Most of the experience with these has been quite negative, with many implementors assuming that private use names are in fact public and long-lived.)

9. Security Considerations

Security is not discussed directly in this document. While the definitions here have no direct effect on security, they are used in many security contexts. For example, authentication usually involves comparing two tokens, and one or both of those tokens might be text; thus, some methods of comparison might involve using some of the internationalization concepts for which terms are defined in this document.

Having said that, other RFCs dealing with internationalization have security consideration descriptions that may be useful to the reader of this document. In particular, the security considerations in RFC 3454, RFC 3629, RFC 4013 [RFC4013], and RFC 5890 go into a fair amount of detail.

10. References

10.1. Normative References

[ISOIEC10646] ISO/IEC, "ISO/IEC 10646:2011. International Standard -- Information technology - Universal Multiple-Octet Coded Character Set (UCS)", 2011.

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

[UNICODE] The Unicode Consortium, "The Unicode Standard, Version 6.0", (Mountain View, CA: The Unicode Consortium, 2011. ISBN 978-1-936213-01-6). <http://www.unicode.org/versions/Unicode6.0.0/>.

10.2. Informative References

[CHARMOD] W3C, "Character Model for the World Wide Web 1.0", 2005, <http://www.w3.org/TR/charmod/>.

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

[ISO3166] ISO, "ISO 3166-1:2006 - Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes", 2006.

[ISO639] ISO, "ISO 639-1:2002 - Code for the representation of names of languages - Part 1: Alpha-2 code", 2002.

[ISO6429] ISO/IEC, "ISO/IEC, "ISO/IEC 6429:1992. Information technology -- Control functions for coded character sets"", ISO/IEC 6429:1992, 1992.

[RFC0952] Harrenstien, K., Stahl, M., and E. Feinler, "DoD Internet host table specification", RFC 952, October 1985.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, October 1989.

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

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

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

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

[RFC2978] Freed, N. and J. Postel, "IANA Charset Registration Procedures", BCP 19, RFC 2978, October 2000.

[RFC3454] Hoffman, P. and M. Blanchet, "Preparation of Internationalized Strings ("stringprep")", RFC 3454, December 2002.

[RFC3490] Faltstrom, P., Hoffman, P., and A. Costello, "Internationalizing Domain Names in Applications (IDNA)", RFC 3490, March 2003.

[RFC3491] Hoffman, P. and M. Blanchet, "Nameprep: A Stringprep Profile for Internationalized Domain Names (IDN)", RFC 3491, March 2003.

[RFC3492] Costello, A., "Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)", RFC 3492, March 2003.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC3743] Konishi, K., Huang, K., Qian, H., and Y. Ko, "Joint Engineering Team (JET) Guidelines for Internationalized Domain Names (IDN) Registration and Administration for Chinese, Japanese, and Korean", RFC 3743, April 2004.

[RFC4013] Zeilenga, K., "SASLprep: Stringprep Profile for User Names and Passwords", RFC 4013, February 2005.

[RFC4647] Phillips, A. and M. Davis, "Matching of Language Tags", BCP 47, RFC 4647, September 2006.

[RFC4713] Lee, X., Mao, W., Chen, E., Hsu, N., and J. Klensin, "Registration and Administration Recommendations for Chinese Domain Names", RFC 4713, October 2006.

[RFC5137] Klensin, J., "ASCII Escaping of Unicode Characters", BCP 137, RFC 5137, February 2008.

[RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, March 2008.

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

[RFC5646] Phillips, A. and M. Davis, "Tags for Identifying Languages", BCP 47, RFC 5646, September 2009.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

[RFC5892] Faltstrom, P., "The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)", RFC 5892, August 2010.

[RFC5895] Resnick, P. and P. Hoffman, "Mapping Characters for Internationalized Domain Names in Applications (IDNA) 2008", RFC 5895, September 2010.

[RFC6055] Thaler, D., Klensin, J., and S. Cheshire, "IAB Thoughts on Encodings for Internationalized Domain Names", RFC 6055, February 2011.

[UAX34] The Unicode Consortium, "Unicode Standard Annex #34: Unicode Named Character Sequences", 2010, <http://www.unicode.org/reports/tr34>.

[UAX9] The Unicode Consortium, "Unicode Standard Annex #9: Unicode Bidirectional Algorithm", 2010, <http://www.unicode.org/reports/tr9>.

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

[UTN6] The Unicode Consortium, "Unicode Technical Note #5: BOCU-1: MIME-Compatible Unicode Compression", 2006, <http://www.unicode.org/notes/tn6/>.

[UTR15] The Unicode Consortium, "Unicode Standard Annex #15: Unicode Normalization Forms", 2010, <http://www.unicode.org/reports/tr15>.

[UTR18] The Unicode Consortium, "Unicode Standard Annex #18: Unicode Regular Expressions", 2008, <http://www.unicode.org/reports/tr18>.

[UTR22] The Unicode Consortium, "Unicode Technical Standard #22: Unicode Character Mapping Markup Language", 2009, <http://www.unicode.org/reports/tr22>.

[UTR6] The Unicode Consortium, "Unicode Technical Standard #6: A Standard Compression Scheme for Unicode", 2005, <http://www.unicode.org/reports/tr6>.

[W3C-i18n-Def] W3C, "Localization vs. Internationalization", September 2010, <http://www.w3.org/International/questions/qa-i18n.en>.

Appendix A. Additional Interesting Reading

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

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

Dalby, Andrew. Dictionary of Languages: The Definitive Reference to More than 400 Languages. New York: Columbia University Press, 2004. ISBN 978-0231115698

Daniels, Peter, and William Bright. The World's Writing Systems. New York: Oxford University Press, 1996. ISBN 0195079930

DeFrancis, John. The Chinese Language: Fact and Fantasy. Honolulu: University of Hawaii Press, 1984. ISBN 0-8284-085505 and 0-8248-1058-6

Drucker, Joanna. The Alphabetic Labyrinth: The Letters in History and Imagination. London: Thames & Hudson, 1995. ISBN 0-500-28068-1

Fazzioli, Edoardo. Chinese Calligraphy. New York: Abbeville Press, 1986, 1987 (English translation). ISBN 0-89659-774-1

Hooker, J.T., et al. Reading the Past: Ancient Writing from Cuneiform to the Alphabet. London: British Museum Press, 1990. ISBN 0-7141-8077-7

Lunde, Ken. CJKV Information Processing. Sebastopol, CA: O'Reilly & Assoc., 1999. ISBN 1-56592-224-7

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

Robinson, Andrew. The Story of Writing: Alphabets, Hieroglyphs, & Pictograms. London: Thames & Hudson, 1995, 2000. ISBN 0-500-28156-4

Sacks, David. Language Visible. New York: Broadway Books (a division of Random House, Inc.), 2003. ISBN 0-7679-1172-5

Appendix B. 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 RFC 3536. Harald Alvestrand and Martin Duerst made extensive useful comments on early versions. Others who contributed to the development of RFC 3536 include Dan Kohn, Jacob Palme, Johan van Wingen, Peter Constable, Yuri Demchenko, Susan Harris, Zita Wenzel, John Klensin, Henning Schulzrinne, Leslie Daigle, Markus Scherer, and Ken Whistler.

Abdulaziz Al-Zoman, Tim Bray, Frank Ellermann, Antonio Marko, JFC Morphin, Sarmad Hussain, Mykyta Yevstifeyev, Ken Whistler, and others identified important issues with, or made specific suggestions for, this new version.

Appendix C. Significant Changes from RFC 3536

This document mostly consists of additions to RFC 3536. The following is a list of the most significant changes.

- Changed the document’s status to BCP.
- Commonly used synonyms added to several descriptions and indexed.
- A list of terms defined and used in IDNA2008 was added, with a pointer to RFC 5890. Those definitions have not been repeated in this document.
- The much-abused term "variant" is now discussed in some detail.
- A discussion of different subsets of the Unicode repertoire was added as Section 4.2 and associated definitions were included.
- Added a new term, "writing style".
- Discussions of case-folding and mapping were expanded.
- Minor edits were made to some section titles and a number of other editorial improvements were made.
- The discussion of control codes was updated to include additional information and clarify that "control code" and "control character" are synonyms.
- Many terms were clarified to reflect contemporary usage.

- The index to terms by section in RFC 3536 was replaced by an index to pages containing considerably more terms.
- The acknowledgments were updated.
- Some of the references were updated.
- The supplemental reading list was expanded somewhat.

Index

Authors' Addresses

Paul Hoffman
VPN Consortium

EMail: paul.hoffman@vpnc.org

John C Klensin
1770 Massachusetts Ave, Ste 322
Cambridge, MA 02140
USA

Phone: +1 617 245 1457
EMail: john+ietf@jck.com