Network Working Group
Request for Comments: 2277
BCP: 18
Category: Best Current Practice

H. Alvestrand
UNINETT

January 1998

IETF Policy on Character Sets and Languages

Status of this Memo

This document specifies an Internet Best Current Practices for the Internet Community, and requests discussion and suggestions for improvements. Distribution of this memo is unlimited.

Copyright Notice

1. Introduction

The Internet is international.

With the international Internet follows an absolute requirement to interchange data in a multiplicity of languages, which in turn utilize a bewildering number of characters.

This document is the current policies being applied by the Internet Engineering Steering Group (IESG) towards the standardization efforts in the Internet Engineering Task Force (IETF) in order to help Internet protocols fulfill these requirements.

The document is very much based upon the recommendations of the IAB Character Set Workshop of February 29-March 1, 1996, which is documented in RFC 2130 [WR]. This document attempts to be concise, explicit and clear; people wanting more background are encouraged to read RFC 2130.

The document uses the terms 'MUST', 'SHOULD' and 'MAY', and their negatives, in the way described in [RFC 2119]. In this case, 'the specification' as used by RFC 2119 refers to the processing of protocols being submitted to the IETF standards process.

2. Where to do internationalization

Internationalization is for humans. This means that protocols are not subject to internationalization; text strings are. Where protocol elements look like text tokens, such as in many IETF application layer protocols, protocols MUST specify which parts are protocol and which are text. [WR 2.2.1.1]

Names are a problem, because people feel strongly about them, many of them are mostly for local usage, and all of them tend to leak out of the local context at times. RFC 1958 [RFC 1958] recommends US-ASCII for all globally visible names.

This document does not mandate a policy on name internationalization, but requires that all protocols describe whether names are internationalized or US-ASCII.

NOTE: In the protocol stack for any given application, there is usually one or a few layers that need to address these problems.

It would, for instance, not be appropriate to define language tags for Ethernet frames. But it is the responsibility of the WGs to ensure that whenever responsibility for internationalization is left to "another layer", those responsible for that layer are in fact aware that they HAVE that responsibility.

3. Definition of Terms

This document uses the term "charset" to mean a set of rules for mapping from a sequence of octets to a sequence of characters, such as the combination of a coded character set and a character encoding scheme; this is also what is used as an identifier in MIME "charset=" parameters, and registered in the IANA charset registry [REG]. (Note that this is NOT a term used by other standards bodies, such as ISO).

For a definition of the term "coded character set", refer to the workshop report.

A "name" is an identifier such as a person’s name, a hostname, a domainname, a filename or an E-mail address; it is often treated as an identifier rather than as a piece of text, and is often used in protocols as an identifier for entities, without surrounding text.

3.1. What charset to use

All protocols MUST identify, for all character data, which charset is in use.

Protocols MUST be able to use the UTF-8 charset, which consists of the ISO 10646 coded character set combined with the UTF-8 character encoding scheme, as defined in [10646] Annex R (published in Amendment 2), for all text.

Protocols MAY specify, in addition, how to use other charsets or other character encoding schemes for ISO 10646, such as UTF-16, but lack of an ability to use UTF-8 is a violation of this policy; such a violation would need a variance procedure ([BCP9] section 9) with clear and solid justification in the protocol specification document before being entered into or advanced upon the standards track.

For existing protocols or protocols that move data from existing datastores, support of other charsets, or even using a default other than UTF-8, may be a requirement. This is acceptable, but UTF-8 support MUST be possible.

When using other charsets than UTF-8, these MUST be registered in the IANA charset registry, if necessary by registering them when the protocol is published.

(Note: ISO 10646 calls the UTF-8 CES a "Transformation Format" rather than a "character encoding scheme", but it fits the charset workshop report definition of a character encoding scheme).

3.2. How to decide a charset

When the protocol allows a choice of multiple charsets, someone must make a decision on which charset to use.

In some cases, like HTTP, there is direct or semi-direct communication between the producer and the consumer of data containing text. In such cases, it may make sense to negotiate a charset before sending data.

In other cases, like E-mail or stored data, there is no such communication, and the best one can do is to make sure the charset is clearly identified with the stored data, and choosing a charset that is as widely known as possible.

Note that a charset is an absolute; text that is encoded in a charset cannot be rendered comprehensibly without supporting that charset.

(This also applies to English texts; charsets like EBCDIC do NOT have ASCII as a proper subset)

Negotiating a charset may be regarded as an interim mechanism that is to be supported until support for interchange of UTF-8 is prevalent; however, the timeframe of "interim" may be at least 50 years, so there is every reason to think of it as permanent in practice.

4. Languages

4.1. The need for language information

All human-readable text has a language.

Many operations, including high quality formatting, text-to-speech synthesis, searching, hyphenation, spellchecking and so on benefit greatly from access to information about the language of a piece of text. [WC 3.1.1.4].

Humans have some tolerance for foreign languages, but are generally very unhappy with being presented text in a language they do not understand; this is why negotiation of language is needed.

In most cases, machines will not be able to deduce the language of a transmitted text by themselves; the protocol must specify how to transfer the language information if it is to be available at all.

The interaction between language and processing is complex; for instance, if I compare "name-of-thing(lang=en)" to "name-of-thing(lang=no)" for equality, I will generally expect a match, while the word "ask(no)" is a kind of tree, and is hardly useful as a command verb.

4.2. Requirement for language tagging

Protocols that transfer text MUST provide for carrying information about the language of that text.

Protocols SHOULD also provide for carrying information about the language of names, where appropriate.

Note that this does NOT mean that such information must always be present; the requirement is that if the sender of information wishes to send information about the language of a text, the protocol provides a well-defined way to carry this information.

4.3. How to identify a language

The RFC 1766 language tag is at the moment the most flexible tool available for identifying a language; protocols SHOULD use this, or provide clear and solid justification for doing otherwise in the document.

Note also that a language is distinct from a POSIX locale; a POSIX locale identifies a set of cultural conventions, which may imply a language (the POSIX or "C" locale of course do not), while a language tag as described in RFC 1766 identifies only a language.

4.4. Considerations for language negotiation

Protocols where users have text presented to them in response to user actions MUST provide for support of multiple languages.

How this is done will vary between protocols; for instance, in some cases, a negotiation where the client proposes a set of languages and the server replies with one is appropriate; in other cases, a server may choose to send multiple variants of a text and let the client pick which one to display.

Negotiation is useful in the case where one side of the protocol exchange is able to present text in multiple languages to the other side, and the other side has a preference for one of these; the most common example is the text part of error responses, or Web pages that are available in multiple languages.

Negotiating a language should be regarded as a permanent requirement of the protocol that will not go away at any time in the future.

In many cases, it should be possible to include it as part of the connection establishment, together with authentication and other preferences negotiation.

4.5. Default Language

When human-readable text must be presented in a context where the sender has no knowledge of the recipient’s language preferences (such as login failures or E-mailed warnings, or prior to language negotiation), text SHOULD be presented in Default Language.

Default Language is assigned the tag "i-default" according to the procedures of RFC 1766. It is not a specific language, but rather identifies the condition where the language preferences of the user cannot be established.

Messages in Default Language MUST be understandable by an English-speaking person, since English is the language which, worldwide, the greatest number of people will be able to get adequate help in interpreting when working with computers.

Note that negotiating English is NOT the same as Default Language; Default Language is an emergency measure in otherwise unmanageable situations.

In many cases, using only English text is reasonable; in some cases, the English text may be augumented by text in other languages.

5. Locale

The POSIX standard [POSIX] defines a concept called a "locale", which includes a lot of information about collating order for sorting, date format, currency format and so on.

In some cases, and especially with text where the user is expected to do processing on the text, locale information may be usefully attached to the text; this would identify the sender’s opinion about appropriate rules to follow when processing the document, which the recipient may choose to agree with or ignore.

This document does not require the communication of locale information on all text, but encourages its inclusion when appropriate.

Note that language and character set information will often be present as parts of a locale tag (such as no_NO.iso-8859-1; the language is before the underscore and the character set is after the dot); care must be taken to define precisely which specification of character set and language applies to any one text item.

The default locale is the "POSIX" locale.

6. Documenting internationalization decisions

In documents that deal with internationalization issues at all, a synopsis of the approaches chosen for internationalization SHOULD be collected into a section called "Internationalization considerations", and placed next to the Security Considerations section.

This provides an easy reference for those who are looking for advice on these issues when implementing the protocol.

7. Security Considerations

Apart from the fact that security warnings in a foreign language may cause inappropriate behaviour from the user, and the fact that multilingual systems usually have problems with consistency between language variants, no security considerations relevant have been identified.

8. References

[10646]
ISO/IEC, Information Technology - Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane, May 1993, with amendments

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

[WR] Weider, C., Preston, C., Simonsen, K., Alvestrand, H, Atkinson, R., Crispin, M., and P. Svanberg, "The Report of the IAB Character Set Workshop held 29 February - 1 March, 1996", RFC 2130, April 1997.

[RFC 1958]
Carpenter, B., "Architectural Principles of the Internet", RFC 1958, June 1996.

[POSIX]
ISO/IEC 9945-2:1993 Information technology -- Portable Operating System Interface (POSIX) -- Part 2: Shell and Utilities

[REG]
Freed, N., and J. Postel, "IANA Charset Registration Procedures", BCP 19, RFC 2278, January 1998.

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

[BCP9]
Bradner, S., "The Internet Standards Process -- Revision 3," BCP 9, RFC 2026, October 1996.

9. Author's Address

Harald Tveit Alvestrand
UNINETT
P.O.Box 6883 Elgeseter
N-7002 TRONDHEIM
NORWAY

Phone: +47 73 59 70 94
EMail: Harald.T.Alvestrand@uninett.no

10. Full Copyright Statement