Network Working Group
Request for Comments: 2130
Category: Informational

C. Weider
Microsoft
C. Preston
Preston & Lynch
K. Simonsen
DKUUG
H. Alvestrand
UNINETT
R. Atkinson
Cisco Systems
M. Crispin
University of Washington
P. Svanberg
KTH

April 1997

The Report of the IAB Character Set Workshop held 29 February - 1 March, 1996

Status of this Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Acknowledgments

The authors would like to sincerely thank Information Sciences Institute (ISI), and in particular Joyce K. Reynolds for graciously hosting this event; Joe Kemp and Jeanine Yamazaki of ISI made sure the facilities met our needs. We also wish to thank the Internet Society, which underwrote travel for participants who might not otherwise have been able to attend. Of course, we also wish to thank the many experts who participated in the workshop and on the mailing list; a complete list of these people can be found in Appendix D. Bunyip Information Systems was kind enough to provide mailing list facilities for this work.

Table of Contents

Abstract

This report details the conclusions of an IAB-sponsored invitational workshop held 29 February - 1 March, 1996, to discuss the use of character sets on the Internet. It motivates the need to have character set handling in Internet protocols which transmit text, provides a conceptual framework for specifying character sets, recommends the use of MIME tagging for transmitted text, recommends a default character set *without* stating that there is no need for other character sets, and makes a series of recommendations to the IAB, IANA, and the IESG for furthering the integration of the character set framework into text transmission protocols.

0: Executive summary

The term 'Character Set' means many things to many people. Even the MIME registry of character sets registers items that have great differences in semantics and applicability. This workshop provides guidance to the IAB and IETF about the use of character sets on the Internet and provides a common framework for interoperability between the many characters in use there.

The framework consists of four components: an architecture model, which specifies components necessary for on-the-wire transmission of text; recommendations for tagging transmitted (and stored) text; recommended defaults for each level of the model; and a set of

recommendations to the IAB, IANA, and the IESG for furthering the integration of this framework into text transmission protocols.

The architectural model specifies 7 layers, of which only three are required for on-the-wire transmission. The Coded Character Set is a mapping from a set of abstract characters to a set of integers. The Character Encoding Scheme is a mapping from a Coded Character Set (or several) to a set of octets. The Transfer Encoding Syntax is a transformation applied to data which has been encoded using a Character Encoding Scheme to allow it to be transmitted. These layers should be specified in a transmitted text stream by using the MIME encoding mechanisms.

This report recommends the use of ISO 10646 as the default Coded Character Set, and UTF-8 as the default Character Encoding Scheme in the creation of new protocols or new version of old protocols which transmit text. These defaults do not deprecate the use of other character sets when and where they are needed; they are simply intended to provide guidance and a specification for interoperability.

1: Introduction

This is the report of an IAB-sponsored invitational workshop on the use of Character Sets on the Internet, held 29 February - 1 March 1996 at Information Sciences Institute (ISI) in Marina del Rey, California. In addition, this report covers the discussion on the mailing list up to and slightly beyond the workshop itself. The goals of this workshop were to provide guidance to the IAB and the IETF about the use of character sets on the Internet, and if possible a common framework for interoperability between the many character sets in use there. Both goals were achieved.

2: Character sets on the Internet - the problem

The term ’character set’ is typically applied to the contents of a wide variety of text transmission and display protocols used on the Internet. Because the term is used to mean different things, confusion has arisen. For example, the MIME registry of character sets [MIME] contains items that may differ greatly in their applicability and semantics in various Internet protocols.

In addition, there is a vast profusion of different text encoding schemes in use on the Internet. This per se is not a problem; each scheme has evolved to meet real needs. However, information applications such as mail, directories, and the World Wide Web have each developed different techniques for dealing with the growing number of schemes. A robust information architecture for the

Internet requires as much interoperability between these techniques as possible.

2.1: Related topics deemed out of scope for this workshop

Successful display of plain text transmitted over the Internet requires a lot of information about the text itself, such as the underlying character set, language, and so forth. An additional set of formatting information is needed if the receiving application wishes to use local (cultural) conventions when it presents the data to the user. This formatting includes information, that provides the data necessary to format certain types of textual data (dates, times, numbers and monetary notation) into a form which is familiar to the user. The POSIX [POSIX] notation of locale encompasses language, coded character set and cultural conventions.

To avoid unfruitful discussion, and to make the best use of the time available for the workshop, we declared the following issues out of scope for the purposes of this workshop:

- glyphs
- sorting
- culture (e.g. do we present the American or British spelling?)
- user interface issues
- internal representation of textual data
- included characters (why aren't certain characters available in any character set?)
- locale (in the POSIX sense)
- font registration
- semantics
- user input/output issues
- Han unification issues

There are some related issues which were included for discussion, most importantly the 'locale' components necessary for transport and identification of multilingual texts.

2.2: Character Set handling in existing protocols

One of the group's overriding concerns was that the framework developed for character set handling not break existing protocols. With that in mind, the way character sets are being used in existing protocols was examined. See Appendix A for a list of those protocols and some recommendations for change.

2.2.1: General comments

The problem areas here fall into three main categories: protocols,

identifiers, and data.

2.2.1.1: Protocols

The protocol machinery SHOULD NOT be changed; allowing, for instance, SMTP [SMTP] to use both MAIL FROM and POST FRA is dangerous to the protocols' stability. However, many protocols carry error messages and other information that is intended for human consumption; it MIGHT be an advantage to allow these to be localized into a specific language and character set, rather than staying in English and US-ASCII [ASCII]. If this is done, new extensions should follow the framework outlined below.

2.2.1.2: Identifiers.

There is a strong statement of direction from the IAB, RFC 1958 [RFC 1958], which states:

> 4.3 Public (i.e. widely visible) names should be in case independent ASCII. Specifically, this refers to DNS names, and to protocol elements that are transmitted in text format.
> ...
> 5.4 Designs should be fully international, with support for localization (adaptation to local character sets). In particular, there should be a uniform approach to character set tagging for information content.

In protocols that up to now have used US-ASCII only, UTF-8 [UTF-8] forms a simple upgrade path; however, its use should be negotiated either by negotiating a protocol version or by negotiating charset usage, and a fallback to a US-ASCII compatible representation such as UTF-7 [UTF-7] MUST be available.

The need for passing application data such as language on individual identifiers varies between applications; protocols SHOULD attempt to evaluate this need when designing mechanisms. Applying the ASCII requirement for identifiers that are only used in a local context (such as private mailbox folder names) is both unrealistic and unreasonable; in such cases, methods for consistency in the handling of character set should be considered.

2.2.1.3: Data

Data that require character set handling includes text, databases, and HTML [HTML] pages, for example. In these the support for multiple character sets and proper application information is absolutely vital, and MUST be supported.

2.3: Architectural requirements

To address the issues enumerated for this work, first an architectural model was created which establishes the components that are required to fully specify the transmission of textual data. Many of these components are already familiar to the users of encoding protocols such as MIME. Not all of these are discussed in detail in this report; we restrict ourselves primarily to those components which are required to specify the 'on-the-wire' phase of text transmission.

Mandating a single, all-encompassing character set would not fit well with the IETF philosophy of planning for architectural diversity. So, the best that can be done is to provide a common *framework* for identifying and using the multitude of character sets available on the Internet. It would be an advantage if the total number of Coded Character Sets could be kept to a minimum. This framework should meet the following requirements:

- it should not break existing protocols (because then the likelihood of deployment is very small),
- it should allow the use of character sets currently used on the Internet, and
- it should be relatively easy to build into new protocols.

3: Architectural model

The basic architectural model which guided our discussions is shown in below. A distinction was made between those segments which were necessary to successfully transmit character set data on-the-wire and those needed to present that data to a user in a comprehensible manner. The discussions were primarily restricted to those segments of the model which specify the 'on-the-wire' transmission of textual data.

User interface issues: these are briefly discussed in Section 3.1.1.
- Layout
- Culture
- Locale
- Language

On-the-wire: see section 3.2 for detailed discussion.
- Transfer Syntax
- Character Encoding Scheme
- Coded Character Set

3.1: Segments defined

3.1:1: User interface

3.1.1.1: Layout

Layout includes the elements needed for displaying text to the user, such as font selection, word-wrapping, etc. It is similar to the 'presentation' layer in the 7-layer ISO telecommunications model [ISO-7498].

3.1.1.2: Culture

Culture includes information about cultural preferences, which affect spelling, word choice, and so forth.

3.1.1.3: Locale

The locale component includes the information necessary to make choices about text manipulation which will present the text to the user in an expected format. This information may include the display of date, time and monetary symbol preferences. Notice that locale modifications are typically applied to a text stream before it is presented to the user, although they also are used to specify input formats.

3.1.1.4: Language

This component specifies the language of the transmitted text. At times and in specific cases, language information may be required to achieve a particular level of quality for the purpose of displaying a text stream. For example, UTF-8 encoded Han may require transmission of a language tag to select the specific glyphs to be displayed at a particular level of quality.

Note that information other than language may be used to achieve the required level of quality in a display process. In particular, a font tag is sufficient to produce identical results. However, the association of a language with a specific block of text has usefulness far beyond its use in display. In particular, as the amount of information available in multiple languages on the World Wide Web grows, it becomes critical to specify which language is in use in particular documents, to assist automatic indexing and retrieval of relevant documents.

The term 'language tag' should be reserved for the short identifier of RFC 1766 [RFC-1766] that only serves to identify the language. While there may be other text attributes intimately associated with the language of the document, such as desired font or text direction, these should be specified with other identifiers rather than overloading the language tag.

3.2: On the wire

There are three segments of the model which are required for completely specifying the content of a transmitted text stream (with the occasional exception of the Language component, mentioned above). These components are:

1) Coded Character Set,
2) Character Encoding Scheme, and
3) Transfer Encoding Syntax.

Each of these abstract components must be explicitly specified by the transmitter when the data is sent. There may be instances of an implicit specification due to the protocol/standard being used (i.e. ANSI/NISO Z39.50). Also, in MIME, the Coded Character Set and Character Encoding Scheme are specified by the Charset parameter to the Content-Type header field, and Transfer Encoding Syntax is specified by the Content-Transfer-Encoding header field.

3.2.1: Coded Character Set

A Coded Character Set (CCS) is a mapping from a set of abstract characters to a set of integers. Examples of coded character sets are ISO 10646 [ISO-10646], US-ASCII [ASCII], and ISO-8859 series [ISO-8859].

3.2.2: Character Encoding Scheme

A Character Encoding Scheme (CES) is a mapping from a Coded Character Set or several coded character sets to a set of octets. Examples of Character Encoding Schemes are ISO 2022 [ISO-2022] and UTF-8 [UTF-8]. A given CES is typically associated with a single CCS; for example, UTF-8 applies only to ISO 10646.

3.2.3: Transfer Encoding Syntax

It is frequently necessary to transform encoded text into a format which is transmissible by specific protocols. The Transfer Encoding Syntax (TES) is a transformation applied to character data encoded using a CCS and possibly a CES to allow it to be transmitted. Examples of Transfer Encoding Syntaxes are Base64 Encoding [Base64], gzip encoding, and so forth.

3.3: Determining which values of CCS, CES, and TES are used

To completely specify which CCS, CES, and TES are used in a specific text transmission, there needs to be a consistent set of labels for specifying which CCS, CES, and TES are used. Once the appropriate mechanisms have been selected, there are six techniques for attaching these labels to the data.

The labels themselves are named and registered, either with IANA [IANA] or with some other registry. Ideally, their definitions are retrievable from some registration authority.

Labels may be determined in one of the following ways:

- Determined by guessing, where the receiver of the text has to guess the values of the CCS, CES, and TES. For example: "I got this from Sweden so it's probably ISO-8859-1." This is obviously not a very foolproof way to decode text.
- Determined by the standard, where the protocol used to transmit the data has made documented choices of CCS, CES, and TES in the standard. Thus, the encodings used are known through the access protocol, for example HTTP [HTTP] uses (but is not limited to) ISO-8859-1, SMTP uses US-ASCII.
- Attached to the transfer envelope, where the descriptive labels are attached to the wrapper placed around the text for transport. MIME headers are a good example of this technique.
- Included in the data stream, where the data stream itself has been encoded in such a way as to signal the character set used. For example, ISO-2022 encodes the data with escape sequences to provide information on the character subset currently being used.
- Agreed by prior bilateral agreement, where some out-of-band negotiation has allowed the text transmitter and receiver to determine the CCS, CES, and TES for the transmitted text.
- Agreed to by negotiation during some phase, typically initialization of the protocol.

3.3.1: Recommendations for value specification mechanisms

While each of these techniques (with the exception of guessing) is useful in particular situations, interoperability requires a more consistent set of techniques. Thus, we recommend that MIME registered values be used for all tagging of character sets and languages UNLESS there is an existing mechanism for determining the required information using one of the other techniques (except guessing). This recommendation will require a fair bit of work on the part of protocol designers, implementors, the IETF, the IESG, and the IAB.

However, it is important to point out that the MIME concept of 'charset' in some cases cuts across several layers of components in our model. While this can be accepted in existing registrations, we also recommend that the MIME registration procedure for character sets be modified to show how a proposed character set deals with the CCS and the CES. Most 'charsets' have a well defined CCS and CES, they should merely be teased apart for the registration.

There are a number of other recommendations, but these will be covered in the next sections.

3.4: Recommended Defaults

For a number of reasons, one cannot define a mandatory set of defaults for all Internet protocols. There is a mass of current practice, future protocols are likely to have different purposes, which may determine their handling of text, and protocols may need specific variation support. For example, in mail, text is a predominant data type and coded character sets then become a major issue for the protocol. Also, since e-mail is ubiquitous and users expect to be able to send it to everyone, the mail protocols need to be quite adept at handling different character set encodings. On the other hand, if strings are seldom used in a given protocol, there is no need to weigh the protocol down with a sophisticated apparatus for handling multiple character sets, assuming that the predicated character set can handle all the protocol's needs. This observation also applies to the specification techniques for character set parameters. If only one character set encoding is needed, it can be made explicit in the protocol specification. Protocols with a greater need for character set support will need a more elaborate specification technique.

3.4.1: Clarity of specification

We recommend that each protocol clearly specify what it is using for each of the layers of the transmission model. Users (or clients) should never have to guess what the parameter is for a given layer.

3.4.2: Default Coded Character Set:

The default Coded Character Set is the repertoire of ISO-10646.

3.4.3: Default Character Encoding Scheme

For text-oriented protocols, new protocols should use UTF-8, and protocols that have a backwards compatibility requirement should use the default of the existing protocol, e.g. US-ASCII for mail, and ISO-8859-1 for HTTP. The recommended specification scheme is the MIME "charset" specification, using the IANA "charset" specifications. The MIME specifications will need to be clarified to meet this model in the future.

For other protocols, the default should be UTF-8 as this initially allows US-ASCII to be entered as-is, and enables the full repertoire of ISO 10646.

Some protocols, such as those descended from SGML [SGML], have other natural notations for characters outside their "natural" repertoire; for instance, HTML [HTML] allows the use of &#nnnn to refer to any ISO 10646 character. Note that this, like all other encodings that depend on "escape characters", redefines at least one character from the base character set for use as an indicator of "foreign" characters. Use of this approach must be weighed very carefully.

3.4.4: Default Transport Encoding Scheme

There is no recommended default for this level. For plain text oriented protocols, the bytestream transport format should be 8-bit clean, possibly with normalization of end-of-line indicators. Some special cases could be made for protocols that are not 8-bit clean, such as encoding it for transport over 7-bit connections. For binary the same recommendation holds as above. The specification technique should either be defined in the protocol, if only one way is permitted, or by use of MIME content-transfer-encoding (CTE) techniques, using IANA registered values.

3.4.5: Default Language

There is no recommended default for the language level. For human readable text, there should always be a way to specify the natural language. The specification technique should be a MIME identifier with IANA registered values for languages. If headers are used, the header should be 'Content-Language'.

3.4.6: Default Locale

The default should be the POSIX locale. The specification technique should use the Cultural register of CEN ENV 12005 [CEN] for the values. If headers are used, the header should be 'Content-Locale'.

3.4.7: Default Culture

There is no recommended default for the Culture level. The specification technique should be a MIME or MIME-like identifier (e.g. Content-Culture) and should use the Cultural register of CEN ENV 12005 for its values.

3.4.8: Default Presentation

There is no recommended default for the Presentation level. The specification technique should be a MIME or MIME-like identifier (e.g. Content-Layout) and use the glyph register of ISO 10036 and other registers for its values.

3.4.9: Multiplexing

In some cases, text transmission may require the use of a number of different values for a given parameter; for example, English annotation of Japanese text might well require shifting the Content-Language parameter. The way to switch the value of parameters within a single body of text depends on the application. For instance, the HTML I18N [I18N] work defines a language attribute on most of its elements, including <SPAN>, <HTML>, and <BODY>, for the purpose of switching between different languages. When only one value is needed, this value should be as general as possible, and specified in the protocol standard with reference to the IANA or other registry value. All levels should be specified explicitly.

3.4.10: Storage

Because stored text may very well be stored without any of the additional information necessary for decoding, stored text SHOULD be tagged in a MIME compliant fashion. This alleviates the problem of being unable to interpret text which has been stored for a long time,

or text whose provenance is not available.

3.5: Guidelines for conversions between coded character sets

This section covers various algorithms to convert a source text S, encoded in the coded character set CCS(S), to a target text T, encoded in the coded character set CCS(T).

Rep(X) is the character repertoire of coded character set X, i.e. the set of characters which can be represented with X.

3.5.1: Exact conversion

When Rep(CCS(S)) and Rep(CCS(T)) are equal or Rep(CCS(S)) is a subset of Rep(CCS(T)), exact conversion is possible; i.e. T is equal to S. The octets just need to be remapped. The algorithm for performing this remapping is simple, if the IANA-registered definition tables for CCS(S) and CCS(T) are available.

3.5.2: Approximate conversion

In all other cases, any conversion creates a text T which differs from S. There are different principles for how this inevitable difference should be handled. A choice between them should be made, depending on the purpose and requirements of the conversion. Where possible, the client application should be given mechanisms to determine what has been done to the text.

3.5.2.1: Length-modifying conversion for human display

When the length of the target text T is allowed to differ from the length of the source text S, one should use a conversion method in which each source character is converted to one or several target character(s), using a best resemblance criteria in the choice of that target character(s).

Examples:

```
LATIN CAPITAL LETTER [*] ->  AE
COPYRIGHT SIGN       [*] -> (c)
```

3.5.2.2: Length-preserving conversion for human display

Where the text T must be presented and the length of T cannot differ from the length of S, one should use a conversion method where each source character is converted to one target character, using some kind of best resemblance criteria in the choice of target character.

Examples:
LATIN CAPITAL LETTER [*] -> A
COPYRIGHT SIGN [*] -> C

3.5.2.3: Conversion without data loss

Where the conversion of the text S into T must be completely reversible, apply a Character Encoding Syntax or other reversible transformation method. This case is most frequently met in data storage requirements.

Examples:
LATIN CAPITAL LETTER [*] -> &AE
COPYRIGHT SIGN [*] -> &(C

An alternate method, which can be used if the size of Rep(CCS(T)) >= Rep(CCS(S)), then for each character in Rep(CCS(S)) which is not present in Rep(CCS(T)), define a mapping into a character in Rep(CCS(T)) which is not present in Rep(CCS(S)).

Examples:
LATIN CAPITAL LETTER [*] -> CYRILLIC CAPITAL LETTER [*]
COPYRIGHT SIGN [*] -> PARTIAL DIFFERENTIAL SIGN [*]

Note that conversion without data loss requires redefining some member of T to indicate "the introduction of character data outside T". This effectively adds another level of CES on top of CES(T).

4: Presentation issues

There are a number of considerations to make in selecting the base character set. One such consideration is the protocol's convenience to users with limited equipment (for example only ISO 8859-1 or a keyboard without the ability to enter all the characters in ISO 10646). Alternative representation should be considered for these users, both for input and output. Possible options for the representation of characters that can not be displayed include transliteration (a la CEN/TC304 or ISO TC46/SC2), RFC 1345 [RFC-1345] representative icons, or the WG2 short name (u+xxxx).

5: Open issues

In addition to the issues declared out of scope and enumerated in section 2.1, the following issues are still open and will need to be addressed in other forums. These issues: language tags, public identifiers such as URL names, and bi-directionality are briefly discussed below as they repeatedly encroached the discussion.

5.1: Language tags

Although the workshop decided not to explicitly address the so-called "CJK issue", a few members felt it was necessary to have some mechanism to address the problem of correct Han character display in the ISO-10646 issue, and that saying that it was a "font issue" would not suffice.

The "CJK issue" refers to the extended discussion about "Han unification", the use of a single ISO-10646 codepoint to represent multiple national variants of a Chinese (Han) character. ISO-10646 can map uniquely to any single CJK national character set, but in the absence of additional information an application can not display an ISO-10646 text using the proper national variants for that text.

It was agreed that language tags would be sufficient to disambiguate unified characters. There was not, in our opinion, a significant technical difference between the use of different coded character sets with overlapping codepoints, and a single coded character set with language tags. Either way, the application has sufficient information to display the text properly.

It was observed that in contemporary usage of MIME charsets, the language is implied as well as the coded character set and the character encoding syntax. We agreed that this is excessive overloading of MIME charsets.

To specify the language used in a particular block of text, we recommend that the MIME tag "Content-Language" be used. There are a number of questions about this approach that need to be worked out, however:

- Is Content-Language: actually suitable?
- Is there an overload between this function and the other intended functions of Content-Language: as described in RFC 1766?
- What, precisely, does "Content-Language: zh-tw, ja, ko, zh-cn" mean in this context? We believe it means that, in drawing a Han character, the Taiwanese variant (presumably traditional Han) is preferred, followed by the Japanese, Korean, and mainland Chinese (presumably simplified Han) variants. It does *NOT* mean "mixed text containing Taiwanese, Japanese, Korean, and mainland Chinese text with all the national variants in each of these".

Mixed CJK text, that simultaneously displays different variants occupying the same codepoint, requires language tags embedded in the data. Ohta and Handa propose in RFC 1554 [RFC-1554] a MIME charset

using ISO-2022 shifts between multiple coded character sets; in effect this is an encoding that uses coded character sets for displaying the appropriate glyphs.

There is some speculation that states that mixed CJK text is relatively infrequent, and that therefore it is acceptable to require that such text be represented using a rich text format that can support language tags. In other words, that a simplifying assumption can be made for TEXT/PLAIN in email using ISO-10646 that will not require multiple display representations for the same codepoint. A mechanism such as RFC 1554 could address this need if it was important; although arguably RFC 1554 should really be identified as TEXT/ISO-2022.

Note again that we recommend that support for language tagging SHOULD be built into new protocols, as this will become a critical component of the automated indexing and retrieval in information applications of the future.

5.2: Public identifiers

There is a considerable demand from the user community for the ability to use non-ASCII characters in URL names, IMAP mailbox names, file names, and other public identifiers. This is still an open problem.

5.3: Bi-directionality

It was realized that a consistent framework for bi-directional text was needed but there was no attempt to work on it in this workshop.

6: Security Considerations

There are no security considerations associated with character sets.

7: Conclusions

This paper provides a conceptual framework and a set of recommendations which, if adopted, should provide a solid foundation for interoperability on the Internet. There are, however, a number of open issues which will need to be addressed to provide ever better use of text on the Internet.

8: Recommendations

8.1: To the IAB

There were a number of recommendations to the IAB about making the standards process more aware of the need for character set interoperability, and about the framework itself.

A: The IAB should trigger the examination of all RFCs to determine the way they handle character sets, and obsolete or annotate the RFCs where necessary.

B: The IESG should trigger the recommendation of procedures to the RFC editor to encourage RFCs to specify character set handling if they specify the transmission of text.

C: The IAB should trigger the production of a perspectives document on the character set work that has gone on in the past and relate it to the current framework.

D: Full ISO 10646 has a sufficiently broad repertoire, and scope for further extension, that it is sufficient for use in Internet Protocols (without excluding the use of existing alternatives). There is no need for specific development of character set standards for the Internet.

E: The IAB should encourage the IRTF to create a research group to explore the open issues of character sets on the Internet. This group should set its sights much higher than this workshop did.

F: The IANA (perhaps with the help of an IETF or IRTF group) should develop procedures for the registration of new character sets for use in the Internet.

G: Register UTF-8 as a Character Encoding Scheme for MIME.

H: The current use of the "x-*" format for distinguishing experimental tags should be continued for private use among consenting parties. All other namespaces should be allocated by IANA.

I: Application protocol RFCs SHOULD include a section on "multilingual Considerations".

J: Application Protocol RFCs SHOULD indicate how to transfer ’on the wire’ all characters in the character sets they use. They SHOULD also specify how to transfer other information that applications may need to know about the data.

K: The IESG should trigger a set of extensions to RFC 1522 to allow language tagging of the free text parts of message headers.

8.2: For new Internet protocols

New protocols do not suffer from the need to be compatible with old 7-bit pipes. New protocol specifications SHOULD use ISO 10646 as the base charset unless there is an overriding need to use a different base character set.

New protocols SHOULD use values from the IANA registries when referring to parameter values. The way these values are carried in the protocols is protocol dependent; if the protocol uses RFC-822-like headers, the header names already in use SHOULD be used.

For protocols with only a single choice for each component, the protocol should use the most general specification and should be specified with reference to the registered value in the protocol standard.

Protocols SHOULD tag text streams with the language of the text.

8.3: For the registration of new character sets

Ned Freed will be releasing a new MIME registration document in conjunction with this paper.

8.3.1: A definition table for a coded character set

A definition table for a coded character set A must for each character C that is in the repertoire of A give:

a) if C is present in ISO 10646, the code value (in hexadecimal form) for that character.

b) If C is not present in ISO 10646, but may be constructed using ISO 10646 combining characters, the series of code values (in hexadecimal form) used to construct that character.

c) if C is not present in ISO 10646, a textual description of the character, and a reference to its origin.

8.3.2: A definition of a character encoding scheme

A definition of a character encoding scheme consists of:

- A description of an algorithm which transforms every possible sequence of octets to either a sequence of pairs <CCS, code value> or to the error state "illegal octet sequence"
- Specifications, either by reference to CCS's registered by IANA or in text, of each CCS upon which this CES is based.

Appendix A:

A-1: IETF Protocols

The following list describes how various existing protocols handle multiple character set information.

Email

SMTP
See 8.2. ESMTP makes it easy to negotiate the use of alternate language and encoding if it is needed.

Headers
RFC 1522 forms an adequate framework for supporting text; UTF-8 alone is not a possible solution, because the mail pathways are assumed to be 7-bit 'forever'. However, RFC 1522 should be extended to allow language tagging of the free text parts of message headers.

Bodies
Selection of charset parameters for Email text bodies is reasonably well covered by the charset= parameter on Text/* MIME types. Language is defined by the Content-language header of RFC 1766. Other information will have to be added using body part headers; due to the way MIME differentiates between body part headers and message headers, these will all have to have names starting with Content- .

NetNews

NNTP
See 8.2. No strong tradition for negotiation of encoding in NNTP exists.

NetNews Messages
These should be able to leverage off the mechanisms defined for Email. One difference is that nearly all NNTP channels are 8-bit clean; some NNTP newsgroups have a tradition of using 8-bit charsets in both headers and bodies. Defining character set default on a per newsgroup basis might be a suitable approach.

RTCP
The identifiers carried as information about parties are already defined to be in UTF-8.

FTP

Protocol

See 8.2. The common use of welcome banners in the login response means that there might be strong reason here to allow client and server to negotiate a language different from the default for greetings and error messages. This should be a simple protocol extension.

Filenames

Many fileservers now how have the capability of using non-ASCII characters in filenames, while the "dir" and "get" commands of are defined in terms of US-ASCII only. One possible solution would be to define a "UTF-8" mode for the transfer of filenames and directory information; this would need to be a negotiated facility, with fallback to US-ASCII if not negotiated. The important point here is consistency between all implementations; a single charset is better here than the ability to handle multiple charsets.

World Wide Web

HTTP

See 8.2. The single-shot stype of HTTP makes negotiation more complex than it would otherwise be.

HTML

Internationalization of HTML [I18N] seems fairly well covered in the current "I18N" document. It needs review to see if it needs more specific details in order to carry application information apart from the language.

URLs

URLs are "input identifiers", and powerful arguments should be made if they are ever to be anything but US-ASCII.

IMAP

IMAP’s information objects are MIME Email objects, and therefore are able to use that standard’s methods. However, IMAP folder names are local identifiers; there is strong reason to allow non-ASCII characters in these. A UTF-8 negotiation might be the most appropriate thing, however, UTF-8 is awkward to use. Unfortunately, UTF-7 isn’t suitable because it conflicts with popular hierarchy delimiters. The most recent IMAP work in progress specification describes a modified UTF-7 which avoids this problem.

DNS
: DNS names are the prime example of identifiers that need to stay in US-ASCII for global interoperability. However, some DNS information, in particular TXT records, may represent information (such as names) that is outside the ASCII range. A single solution is the best; problems resulting from UTF-8 should be investigated.

WHOIS++
: WHOIS++ version 1 is defined to use ISO 8859-1. The next version will use UTF-8. The currently designed changes will also allow the specification of individual attributes on attribute names; these will make the passing of application information about the values (such as language) easier. No immediate action seems necessary.

WHOIS
: This has been a stable protocol for so many years now that it seems unwise to suggest that it be modified. Furthermore, compatible extensions exist in RWHOIS and WHOIS++; modification should rather be made to these protocols than to the WHOIS protocol itself.

Telnet
: This is a prime example of protocol where character set support is necessary and nonexistent. The current work in progress on character set negotiation in Telnet seems adequate to the task; the question of passing other application data that might be useful is still open.

A-2: Non-IETF protocols

For these protocols, the IETF does not have any power to change them. However, the guidelines developed by the workshop may still be useful as input to the further development of the protocols.

Gopher: Gopher, Gopher+

Prospero (Archie)

NFS: Filesystem

CORBA, Finger, GEDI, IRC, ISO 10160/1, Kerberos, LPR, RSTAT, RWhois, SGML, TFTP, X11, X.500, Z39.50

Appendix B: Acronyms

ASCII	American National Standard Code for Information Character Sets
CCS	Coded Character Sets
CEN ENV	European Committee for Standardisation (CEN) European pre-standard (ENV)
CES	Character Encoding Scheme
CJK	Chinese Japanese Korean
CORBA	Common Object Request Broker Architecture
CTE	Content Transfer Encoding
DNS	Domain Name Service
ESMTP	Extended SMTP
FTP	File Transfer Protocol
HTML	Hypertext Transfer Protocol
I18N	Internationalization (or 18 characters between the first (I) and last (n)character)
IAB	Internet Activities Board
IANA	Internet Assigned Numbers Authority
IESG	Internet Engineering Steering Group
IETF	Internet Engineering Task Force
IMAP	Internet Message Access Protocol
IRC	Internet Relay Chat
IRTF	Internet Research Task Force
ISI	Information Sciences Institute
ISO	International Standards Organization
MIME	Multipurpose Internet Mail Extensions
NFS	Networked File Server
NNTP	Net News Transfer Protocol
POSIX	Portable Operating System Interface
RFC	Request for Comments (Internet standards documents)
RPC	Remote Procedure Call
RSTAT	Remote Statistics
RTCP	Real-Time Transport Control Protocol
Rwhois	Referral Whois
SGML	Standard Generalized Mark-up Language
SMTP	Simple Mail Transfer Protocol
TES	Transfer Encoding Syntax
TFTP	Trivial File Transfer Protocol
URL	Uniform Resource Locator
UTF	Universal Text/Translation Format

Appendix C: Glossary

Bi-directionality - A property of some text where text written right-to- left (Arabic or Hebrew) and text written left-to-right (e.g. Latin) are intermixed in one and the same line.

Character - A single graphic symbol represented by sequence of one or more bytes.

Character Encoding Scheme - The mapping from a coded character set to an encoding which may be more suitable for specific purpose. For example, UTF-8 is a character encoding scheme for ISO 10646.

Character Set - An enumerated group of symbols (e.g., letters, numbers or glyphs)

Coded Character Set - The mapping from a set of integers to the characters of a character set.

Culture - Preferences in the display of text based on cultural norms, such as spelling and word choice.

Language - The words and combinations of words the constitute a system of expression and communication among people with a shared history or set of traditions.

Layout - Information needed to display text to the user, similar to the presentation layer in the ISO telecommunications model.

Locale - The attributes of communication, such as language, character set and cultural conventions.

On-the-wire - The data that actually gets put into packets for transmission to other computers.

Transfer Encoding Syntax - The mapping from a coded character set which has been encoded in a Character Encoding Scheme to an encoding which may be more suitable for transmission using specific protocols. For example, Base64 is a transfer encoding syntax.

Appendix D: References

[*] Non-ASCII character

[ASCII] ANSI X3.4:1986 "Coded Character Sets - 7 Bit American National Standard Code for Information Interchange (7-bit ASCII)"

[Base64] Freed, N., and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[CEN] see http://tobbi.iti.is/TC304/welcome.html for current status.

[HTML] Berners-Lee, T., and D. Connolly, "Hypertext Markup Language - 2.0", RFC 1866, November 1995.

[HTTP] Berners-Lee, T., Fielding, R., and H. Nielsen, "Hypertext Transfer Protocol -- HTTP/1.0", RFC 1945, May 1996.

[I18N] Yergeau, F., et.al., "Internationalization of the Hypertext Markup Language", RFC 2070, January 1997.

[IANA] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1700, ISI, October 1994.

[ISO-2022] ISO/IEC 2022:1994, "Information technology -- Character Code Structure and Extension Techniques", JTC1/SC2.

[ISO-7498] ISO/IEC 7498-1:1994, "Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model".

[ISO-8859] Information Processing -- 8-bit Single-Byte Coded Graphic Character Sets -- Part 1: Latin Alphabet no. 1, ISO 8859-1:1987(E). Part 2: Latin Alphabet no. 2, ISO 8859-2 1987(E). Part 3: Latin Alphabet no. 3, ISO 8859-3:1988(E). Part 4: Latin Alphabet no. 4, ISO 8859-4, 1988(E). Part 5: Latin/Cyrillic Alphabet ISO 8859-5, 1988(E). Part 6: Latin/Arabic Alphabet, ISO 8859-6, 1987(E). Part 7: Latin/Greek Alphabet, ISO 8859-7, 1987(E). Part 8: Latin/Hebrew Alphabet, ISO 8859-8-1988(E).Part 9: Latin Alphabet no. 5, ISO 8859-9, 1990(E). Part 10: Latin Alphabet no. 6, ISO 8859-10:1992(E).

[ISO-10646] ISO/IEC 10646-1:1993(E), "Information technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane". JTC1/SC2, 1993

[MIME] See [Base64]

[POSIX] Institute of Electrical and Electronics Engineers. "IEEE standard interpretations for IEEE standard portable operating systems interface for computer environments". IEEE Std 1003.1 -1988/Int, 1992 edition. Sponsor, Technical Committee on Operating Systems of the IEEE Computer Society. New York, NY: Institute of Electrical and Electronic Engineers, 1992.

RFC 1340 See [IANA]

[RFC-1345] Simonsen, K., "Character Mnemonics & Character Sets", RFC 1345, Rationel Alim Planlaegning, June 1992.

[RFC-1554] Ohta, M., and K. Handa, "ISO-2022-JP-2: Multilingual Extension of ISO-2022-JP", Tokyo Institute of Technology, ETL, December 1993.

RFC 1642 See [UTF-7]

[RFC-1766] Alvestrad, H., "Tags for the Identification of Languages", RFC 1766, UNINETT, March 1995.

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

[SGML] ISO 8879:1986 "Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML)"

[SMTP] Postel, J., "Simple Mail Transfer Protocol", STD 10, RFC 821, August, 1982.

[Unicode] "The Unicode standard, version 2.0. Unicode Consortium. Reading, Mass.: Addison-Wesley Developers Press, 1996

[UTF-7] Goldsmith, D., and M. Davis, "UTF-7: A Mail Safe Transformation Format of Unicode", RFC 1642, Taligent, Inc., July 1994.

[UTF-8] International Standards Organization, Joint Technical Committee 1 (ISO/JTC1), "Amendment 2:1993, UCS Transformation Format 8 (UTF-8)", in ISO/IEC 10646-1:1993 Information technology - Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane. JTC1/SC2, 1993.

Appendix E: Recommended reading

Alvestrand, H., "Tags for the Identification of Languages", RFC 1766, UNINETT, March 1995.

Alvestrand, H., "X.400 Use of Extended Character Sets", RFC 1502, SINTEF DELAB, August 1993.

Borenstein, N., "Implications of MIME for Internet Mail Gateways", RFC 1344, Bellcore, June 1992.

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

Chernov, A., "Registration of a Cyrillic Character Set", RFC 1489, RELCOM Development Team, July 1993.

Choi, U., and K. Chan, "Korean Character Encoding for Internet Messages", RFC 1557, KAIST, December 1993.

Freed, N., and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.

Goldsmith, D., and M. Davis, "Transformation Format for Unicode", RFC 1642, Taligent, Inc., July 1994.

Goldsmith, D., and M. Davis, "Using Unicode with MIME", RFC 1641, Taligent, Inc., July 1994.

Jerman-Blazic, B. "Character handling in computer communication" in "user needs in information technology standards", Computer Weekly Professional service, eds. C.D. Evans, B.L. Meed & R.S. Walker, P.C. Butterworth Heineman, 1993, Oxford, Boston, p. 102-129.

Jerman-Blazic, B. "Tool supporting the internationalization of the generic network services", Computer Networks and ISDN Systems, No. 27 (1994), p. 429-435.

Jerman-Blazic, B., A. Gogala and D. Gabrijelcic, "Transparent language processing: A solution for internationalization of Internet services", The LISA Forum Newsletter, 5 (1996) p. 12-21

Lee, F., "HZ - A Data Format for Exchanging Files of Arbitrarily Mixed Chinese and ASCII Characters", RFC 1843, Stanford University, August 1995.

McCarthy, J., "Arbitrary Character Sets", RFC 373, Stanford University, July 1972.

Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Two: Message Header Extensions for Non-ASCII Text", RFC 1522, September 1993. (Obsoleted by RFC 2047.)

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

Murai, J., Crispin, M., and E. von der Poel. "Japanese Character Encoding for Internet Messages", RFC 1468, Keio University & Panda Programming, June 1993.

Nussbacher, H., "Handling of Bi-directional Texts in MIME", Israeli Inter-University, December 1993.

Nussbacher, H., and Y. Bourvine, "Hebrew Character Encoding for Internet Messages", RFC 1555, Israeli Inter-University and Hebrew University, December 1993.

Ohta, M., "Character Sets ISO-10646 and ISO-10646-J-1", RFC 1815, Tokyo Institute of Technology, July 1995.

Postel, J., and J. Reynolds, "File Transfer Protocol (FTP)", STD 9, RFC 959, ISI, October 1985.

Postel, J., and J. Reynolds, "Telnet Protocol Specification", STD 8, RFC 854, ISI, May 1983.

Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1700, ISI, October 1994. p.100-117.

Rose, M., "The Internet Message", Prentice Hall, 1992.

Simonsen, K., "Character Mnemonics & Character Sets", RFC 1345, Rationel Almen Planlaegning, June 1992.

Unicode Consortium. "The Unicode standard, version 2.0. Reading, Mass.: Addison-Wesley Developers Press, 1996

Wei, U., et.al. "ASCII Printable Characters-Based Chinese Character Encoding for Internet Messages", RFC 1842, AsiInfo Services, Inc., et.al. August 1995.

Yergeau, F. "UTF-8, a transformation format of Unicode and ISO 10646", RFC 2044, ALIS Technologies, October 1996.

Zhu, H., et.al., "Chinese Character Encoding for Internet Messages", RFC 1922, Tsinghua University, et.al., March 1996.

Appendix F: Workshop attendee list

These people were participants on the workshop mailing list. An * indicates that the person attended the workshop in person.

Glenn Adams <glenn@spyglass.com>
* Joan Aliprand <joan@unicode.org>
* Harald Alvestrand <Harald.T.Alvestrand@uninett.no>
* Ran Atkinson <ran@cisco.com>
* Bert Bos <bert@w3.org>
* Brian Carpenter <brian@dxcoms.cern.ch>
* Mark Crispin <mrc@panda.com>
Makx Dekkers <dekkers@pica.nl>
Robert Elz <kre@munnari.oz.au>
Patrik Faltstrom <paf@paf.se>
* Zhu Haifeng <zhf@net.tsinghua.edu.cn>
Keniichi Handa<handa@etl.go.jp>
Olle Jarnefors <ojarnef@admin.kth.se>
Borka Jerman-Blazic <borka@e5.ijs.si>
John Klensin <klensin@mail1.reston.mci.net>
* Larry Masinter <masinter@parc.xerox.com>
* Rick McGowan <Rick_McGowan@next.com>
* Keith Moore <moore+charsets@cs.utk.edu>
* Lisa Moore <lisam@vnet.ibm.com>
Ruth Moulton <ruth@muswell.demon.co.uk>
* Cecilia Preston <cecilia@well.com>
* Joyce K. Reynolds <jkrey@isi.edu>
* Keld Simonsen <keld@dkuug.dk>
* Gary Smith <Gary_Smith@oclc.org>
* Peter Svanberg <psv@nada.kth.se>
* Chris Weider <cweider@microsoft.com >

Appendix G: Authors' Addresses

Chris Weider
Microsoft Corp.
1 Microsoft Way
Redmond, WA 98052
USA

EMail: cweider@microsoft.com

Cecilia Preston
Preston & Lynch
PO Box 8310
Emeryville, CA 94662
USA

EMail: cecilia@well.com

Keld Simonsen
DKUUG
Freubjergvey 3
DK-2100 Kxbenhavn X
Danmark

EMail: Keld@dkuug.dk

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

EMail: Harald.T.Alvestrand@uninett.no

Randall Atkinson
cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

EMail: rja@cisco.com

Mark Crispin
Networks & Distributed Computing
University of Washington
4545 15th Avenue NE
Seattle, WA 98105-4527
USA

EMail: mrc@cac.washington.edu

Peter Svanberg
Dept. of Numberical Analysis and Computing Science (Nada)
Royal Institute of Technology
SE-100 44 STOCKHOLM
SWEDEN

EMail: psv@nada.kth.se