Encoding of ISO/IEC 10646-1/Unicode in MIME

Mark Davis and David Goldsmith
mark_davis@taligent.com
david_goldsmith@taligent.com

Status of this Memo

This document is a preliminary proposal, intended to be eventually submitted as an Internet standard. This draft is for discussion purposes only.

Abstract

ISO/IEC 10646-1:1993(E) and the Unicode Standard, version 1.1, jointly define a 16 bit character set (hereafter referred to as BMP, the Basic Multilingual Plane of 10646) which encompasses most of the world's writing systems. However, Internet mail (STD 11, RFC 822) currently supports only 7-bit US ASCII as a character set. MIME (RFC 1521 and RFC 1522) extends Internet mail to support different media types and character sets, and thus could support BMP in mail messages. However, MIME neither defines BMP as a permitted character set nor specifies how it would be encoded.

This document is a proposed addition to RFC 1521 and RFC 1522 specifying the encoding of ISO/IEC 10646-1/Unicode within MIME. It references a companion document, "UTF-7: A Mail Safe Transformation Format of ISO/IEC 10646-1/Unicode".

Motivation

Since BMP is starting to see widespread commercial adoption, users will want a way to transmit information in this character set in mail messages and other Internet media. Since MIME was expressly designed to allow such extensions and is on the standards track for the Internet, it is the most appropriate means for encoding BMP. RFC 1521 and RFC 1522 currently do not define BMP as an allowed character set.

In addition to allowing use of BMP within MIME bodies, another goal is to specify a way of using BMP that allows text which consists largely, but not entirely, of US-ASCII characters to be represented in a way that can be read by mail clients who do not understand BMP. This is in keeping with the philosophy of MIME.

Overview

Two ways of using BMP are specified. The first is a straightforward use of BMP as specified in the ISO/IEC 10646-1:1993(E) document. The second is based on the transformation format UTF-7.

The first encoding is intended for situations where sender and recipient do not want to do a lot of processing, or when the text does not consist primarily of characters from the US-ASCII character set.

The second encoding is intended for situations where the text consists primarily of US-ASCII, with occasional characters from other parts of BMP. This encoding allows the US-ASCII portion to be read by all recipients without having to support BMP.

Finally, in keeping with the principles set forth in RFC 1521, text which can be encoded using the US-ASCII or ISO-8859-x character sets should be so encoded where possible, for maximum interoperability. [Use of UTF-7 keeps to the spirit if not the letter of this principle, since it reduces to (mostly) US-ASCII in the limiting case.]

Definitions

The definition of character set BMP:

> The 16 bit character set BMP is defined by the international standard ISO/IEC 10646-1:1993(E); Coded Representation Form=UCS-2; Subset=300; Implementation Level=3. This character set is identical with the character repertoire and coding of The Unicode Standard, Version 1.1.
>
> **Note**. Unicode 1.1 further specifies the use and interaction of these character codes beyond the ISO standard. However, any valid BMP sequence is a valid Unicode sequence; Unicode supplies interpretations of sequences on which the ISO standard is silent as to interpretation.
>
> This character set is encoded as sequences of octets, two per 16-bit character, with the most significant octet first. Text with an odd number of octets is ill-formed.
>
> **Rationale**. ISO/IEC 10646-1:1993(E) specifies that when characters in the UCS-2 form are serialized as octets, that the most significant octet appear first. This is also in keeping with common network practice of choosing a canonical format for transmission.

Character set T is the proposed standard transformation format of BMP, as defined in the document "UTF-7: A Mail Safe Transformation Format of ISO/IEC 10646-1/Unicode".

Encoding Character Set BMP Within MIME

Character set BMP uses 16 bit characters, and therefore may only be used with the Binary or Base64 content transfer encodings of MIME. In header fields, it may only be used with the B content transfer encoding. The MIME character set identifier is ISO-10646-UNICODE.

> **Rationale**. There is no other succinct identification for this character set that might not be confused with other variants of ISO/IEC 10646-1. Other choices such as ISO-10646-BMP or ISO-10646-UCS-2 are ambiguous. ISO-10646-1-1993-E-UCS-2-SUBSET-300-LEVEL-3 is fairly unambiguous, but we presumed that people would prefer brevity.

Example. Here is a text portion of a MIME message containing the word “nihongo” (hexadecimal 65E5,672C,8A9E) written in Han characters.

```
Content-Type: text/plain; charset=ISO-10646-UNICODE
Content-Transfer-Encoding: base64
ZeVnLIqe
```

Example. Here is a text portion of a MIME message containing the BMP sequence “A<NOT IDENTICAL TO><ALPHA>.” (hexadecimal 0041,2262,0391,002E)

```
Content-Type: text/plain; charset=ISO-10646-UNICODE
Content-Transfer-Encoding: base64
AEEiYgORAC4=
```

Encoding Character Set T Within MIME

Character set T is safe for mail transmission and therefore may be used with any content transfer encoding in MIME. Specifically, the 7 bit encoding for bodies and the Q encoding for headers are both acceptable. The MIME character set identifier is ISO-10646-UTF7.

Example. Here is a text portion of a MIME message containing the BMP sequence “Hi Mom <WHITE SMILING FACE>!” (hexadecimal 0048, 0069, 0020, 004D, 006F, 004D, 0020, 263A, 0021).

```
Content-Type: text/plain; charset=ISO-10646-UTF7
Hi Mom +Jjo!
```

Example. Here is a text portion of a MIME message containing the BMP sequence representing the Han characters for the Japanese word “nihongo” (hexadecimal 65E5,672C,8A9E).

```
Content-Type: text/plain; charset=ISO-10646-UTF7
+ZeVnLIqe-
```

Example. Here is a text portion of a MIME message containing the BMP sequence “A<NOT IDENTICAL TO><ALPHA>.” (hexadecimal 0041,2262,0391,002E).

```
Content-Type: text/plain; charset=ISO-10646-UTF7
A+ImIDkQ.
```

Example. Here is a text portion of a MIME message containing the BMP sequence “Item 3 is <POUND SIGN>1.” (hexadecimal 0049, 0074, 0065, 006D, 0020, 0033, 0020, 0069, 0073, 0020, 00A3, 0031, 002E).

```
Content-Type: text/plain; charset=ISO-10646-UTF7
Item 3 is +AKM-1.
```

Discussion

In this section we will motivate the introduction of UTF-7 as opposed to the alternative of using the existing encodings of BMP (e.g. UTF-FSS) with MIME's content transfer encodings. Before discussing this, it will be useful to list some assumptions about character frequency within typical natural language text strings that we use to estimate typical storage requirements:

1. Most Western European languages use roughly 7/8 of their letters from US-ASCII and 1/8 from Latin 1 (ISO-8859-1).

2. Most non-European alphabet-based languages (e.g., Greek) use about 1/6 of their letters from ASCII (since white space is in the 7-bit area) and the rest from their alphabets.

3. East Asian ideographic-based languages (including Japanese) use essentially all of their characters from the Han or CJK syllabary area.

4. The = character does not occur frequently enough to affect the results.

Notice that current 8 bit standards, such as ISO-8859-x, require use of a content transfer encoding. For comparison with the subsequent discussion, the costs break down as follows (note that many of these figures are approximate since they depend on the exact composition of the text):

8859-x in Base64

Text type	Average octets/character
All	1.33

8859-x in Quoted Printable

Text type	Average octets/character
US-ASCII	1
Western European	1.25
Other	2.67

Note also that BMP encoded in Base64 takes a constant 2.66 octets per character. For purposes of comparison, we will look at UTF-FSS in Base64 and Quoted Printable, and UTF-7. UTF-1 gives results substantially similar to UTF-FSS. Also note that fixed overhead for long strings is relative to 1/n, where n is the encoded string length in octets.

UTF-FSS in Base64

Text type	Average octets/character
US-ASCII	1.33
Western European	1.5
Some Alphabetics	2.44
All others	4

UTF-FSS in Quoted Printable

Text type	Average octets/character
US-ASCII	1
Western European	1.63
Some Alphabetics	5.17
All others	7-9

UTF-7

Text type	Average octets/character
Most US-ASCII	1
Western European	1.5
All others	2.67+2/n

We feel that the UTF-FSS in Quoted Printable option is not viable due to the very large expansion of all text except Western European. This is only viable in texts consisting of large expanses of US-ASCII or Latin characters with occasional other characters interspersed. We would prefer to introduce one encoding that works reasonably well for all users.

We also feel that UTF-FSS in Base64 has high expansion for non-Western-European users, and is less desirable because it cannot be read directly, even when the content is largely US-ASCII. The base encoding of UTF-7 gives competitive results and is readable for ASCII text.

UTF-7 gives results competitive with ISO-8859-x, with access to all of the BMP character set. We believe this justifies the introduction of a new transformation format of BMP.

As an alternative to use of UTF-7, it is possible to intermix BMP characters with other character sets using an existing MIME mechanism, the multipart/mixed content type (thanks to Nathaniel Borenstein for pointing this out). For instance (repeating an earlier example):

```
Content-type: multipart/mixed; boundary=foo

--foo
Content-type: text/plain; charset=us-ascii

Hi Mom
--foo
Content-type: text/plain; charset=ISO-10646-UNICODE
Content-transfer-encoding: base64

Jjo=
--foo
Content-type: text/plain; charset=us-ascii

!
--foo--
```

Theoretically, this removes the need for UTF-7. However, we feel that as use of the BMP character set becomes more widespread, intermittent use of specialized BMP characters (such as dingbats and mathematical symbols) will occur, and that text will also typically include small snippets from other script systems, such as Cyrillic, Greek, or East Asian languages (anything in the Roman script system is already handled adequately by existing MIME character sets). Although the multipart technique works well for large chunks of text in alternating character sets, we feel it does not adequately support the kinds of uses just discussed, and so we still believe the introduction of UTF-7 is justified.

Summary

To be added later.

References

To be added later.

ISO/IEC 10646-1:1993(E); Unicode v1, v2, 1.1 TR; RFC 822, 1521, 1522; UTF-7; UTF-2 (X/Open)

Acknowledgements

Many thanks to the following people for their helpful comments and suggestions:

Nathaniel Borenstein, Lee Collins, John Jenkins.

[more later]