                    X.400 Use of Extended Character Sets

Status of this Memo

   This RFC specifies an IAB standards track protocol for the Internet
   community, and requests discussion and suggestions for improvements.
   Please refer to the current edition of the "IAB Official Protocol
   Standards" for the standardization state and status of this protocol.
   Distribution of this memo is unlimited.

1. Introduction

   Since 1988, X.400 has had the capacity for carrying a large number of
   different character sets in a message by using the body part
   "GeneralText" defined by ISO/IEC 10021-7.

   Since 1992, the Internet also has the means of passing around
   messages containing multiple character sets, by using the mechanism
   defined in RFC-MIME.

   This RFC defines a suggested method of using "GeneralText" in order
   to harmonize as much as possible the usage of this body part.

2. General principles

2.1. Goals

   The target of this memo is to define a way of using existing
   standards to achieve:

   (1)   in the short term, a standard for sending E-mail in the
         European languages (Latin letters with European accents,
         Greek and Cyrillic)

   (2)   in the medium term, extending this to cover the Hebrew and
         Arabic character sets

   (3)   in the long term, opening up true international E-mail by
         allowing the full character set specified in ISO-10646 to be
         used.

The author believes that this document gives a specification that can
easily accomodate the use of any character set in the ISO registry,
and, by giving guidance rules for choosing character sets, will help
interworking.

2.2. Families of character sets

2.2.1. ISO 6937/T.61

ISO 6937 is a code technique used and recommended in T.51 and T.101
(Teletex and Videotex service) and in X.500, providing a repertoire
of 333 characters from the Latin script by use of non- spacing
diacritical marks. It corresponds closely to CCITT recommendation
T.61.

The problem with that technique is that the character stream comes in
two modes, i.e., some characters are coded with one byte and some
with two (composite characters). This makes information processing
systems such as an E-mail UA or GW more complex.

It is also not extensible to other languages like Korean or Chinese,
or even Greek, without invoking the character set switching
techniques of ISO 2022.

2.2.2. ISO 8859

ISO 8859 defines a set of character sets, each suitable for use in
some group of languages. Each character in ISO 8859 is coded in a
single byte.

There are currently 11 parts of ISO 8859, plus a "supplementary" set,
registered as ISO IR 154. Most languages using single-byte characters
can be written in one or another of the ISO 8859 sets. There are
sets covering Greek, Hebrew and Arabic, but there is still
controversy over the problem of the rendering direction for Hebrew
and Arabic.

All the ISO 8859 sets include US-ASCII as a subset. All use 8 bits.

ISO 8859 is regarded by many as a solution; for instance, the X
windows system now comes with ISO-8859-1 as the "standard" character
set, with the possibility of specifying others. But since the same
applications often do not support character set switching within
text, it is problematic to use these in a truly multilingual
environment. (Also, most fonts claiming to be "ISO- 8859-1" in X11R5
are actually 7-bit fonts. The implied lie is very unfortunate.)

It turns out to work fine, however, if the second language is
English, since this can be written in all ISO 8859 sets.

The parts 3 and 4 have not seen wide acceptance, and it is expected
that they will be discarded. They should therefore not be used.

Note that an ISO 8859 set is actually 2 sets in the ISO sense: US-
ASCII in the G0 set and another character set in the G1 set. The
overloading of the word "character set" is unfortunate, but
traditional.

2.2.3. ISO 10646

At the moment of writing, ISO 10646 has just been accepted as an
International Standard. It is basically a 32-bit character set, with
all of the currently used characters being numbered by the first 16
bits, leaving some room for expansion.

It is not possible to use ISO 10646 as a normal character set,
because it does not conform to the rules for usage of byte values set
down in ISO 2022 and other places; it uses the "control space" for
(parts of) graphic character codes.

There are a number of ways to encode ISO 10646 characters "on the
wire". There are methods within the ISO 2022 standard to switch to
these, either as "other coding system without return" or as "other
coding system with return" (that is, you can go back from it to the
one you came from using an ISO 2022 escape sequence).

The following registrations have been made:

 ISO 10646 UCS-2 Level 1 has been registered with ESC 2/5 2/15 4/0,
 ISO 10646 UCS-4 Level 1 has been registered with ESC 2/5 2/15 4/1,

The following are applied for:

 Reg# Escape sequence   Standard/Sponsor   Description
 174 ESC 2/5 2/15 F   ISO/IEC 10646   UCS-2, Level 2
 175 ESC 2/5 2/15 F   ISO/IEC 10646   UCS-4, Level 2
 176 ESC 2/5 2/15 F   ISO/IEC 10646   UCS-2, Level 3
 177 ESC 2/5 2/15 F   ISO/IEC 10646   UCS-4, Level 3
 178 ESC 2/5 F   ISO/IEC 10646   UTF-1

 << NOTE: The registration numbers for UCS-2 level 1 and UCS-4
 level 1 are not known. Neither are the assigned final characters
 for the other sets. Information requested!>>

This character set will become very important in the future, but at
the moment, few systems are able to support this directly.

The GeneralText body part can be used for carrying any of these
character sets.

2.3. Body parts that can be used in X.400

At the moment, no established way of transferring a full set of
characters in X.400-based E-mail exists. In the future, it is likely
that a new body part, based in ISO 10646, will be available, or
GeneralText may be able to use ISO 10646, but this matter has not yet
been clarified.

In the short term, the deployed and available body parts are:

   (1) IA5Text

   (2) For X.400/84: ISO6937Text and Teletex

   (3) For X.400/88: GeneralText

IA5Text is the method of choice for E-mail that contains only
characters from IA5 (equivalent to US-ASCII).

The ISO6937Text body part is defined in the ISO DIS documents
corresponding to X.400(84) [10]; these never became a standard, so
they are now quite difficult to find. It is in principle limited to
using text that can be presented in ISO 6937, but since ISO 6937
refers to the ISO 2022 method of changing character sets, it is
theoretically possible to use any ISO registered character set, but
there is no facility for announcing the character sets used. This
makes interworking with equipment that does not support the same
character sets complex.

It is still, however, the only body part suitable for carrying non-
paginated text in non-basic character sets in X.400(84).

Teletex, which is identical in all versions of the X.400 standard,
has the same problem of implicit ISO6937, but has the added problem
that it also specifies a page format, with, for instance, a left
margin of 5 character positions. This is often not desirable.

The details of Teletex are specified in recommendation T.51 and its
relatives.

GeneralText is defined in ISO 10021-8, the part of [9] that
corresponds to CCITT recommendation [11]. It is an Extended body

part, so no modification to CCITT implementations is needed to carry it.

GeneralText is suitable for interchange, since it has got proper announcement facilities. It can use any number of character sets, and announces them both in the Encoded Information Types of the X.400 envelope and the parameters of the body part.

We recommend this body part for carrying unformatted text in X.400/88.

3.   GUIDELINES FOR THE GENERATION OF GENERALTEXT

3.1.   Formal definition of GeneralText

A GeneralText message is a byte stream that contains characters and character switching sequences according to [12].

The X.400 ASN.1 definition of the GeneralText body part is:

```
general-text-body-part EXTENDED-BODY-PART-TYPE
    PARAMETERS GeneralTextParameters IDENTIFIED BY id-ep-general-text
    DATA       GeneralTextData
    ::= id-et-general-text

GeneralTextParameters ::= SET OF CharacterSetRegistration

CharacterSetRegistration ::= INTEGER (1..32767)

GeneralTextData ::= GeneralString
```

The definition is from ISO/IEC 10021-7 [9], Annex I, with modifications made in the MHS Implementor' Guide, version 8, chapter 3.6.3, bullet F130. It does not appear in the CCITT version of the standards.

3.2.   Brief description of ISO 2022 character set switching

There are 4 graphic character sets active at any time in a GeneralText message, called G0, G1, G2 and G3. In addition, there are 2 control character sets, called C0 and C1.

At any moment, one of the sets G0-G3 is active in code positions 2/1 to 7/14, and another is active in code positions 10/0 to 15/15. The setting is achieved by so-called "locking shift" sequences.

(Formally, code positions 2/0 and 7/15 are reserved for "space" and "DEL" respectively, and only 94-character character sets can be used

in G0. In practice, this restriction is sometimes ignored)

Single characters from the non-active sets may be invoked by the use
of "single shift" sequences.

The control character sets always occupy the code positions 0/0 to
1/15 (C0) and 8/0 to 9/15 (C1).

The character sets currently active as G0-G3 and C0-C1 may be changed
using "character set designating sequences".

At the beginning of a GeneralText message, one must always assume
that set 2 (IA5) is active as G0, shifted into the lower half, that
set 1 (standard) is active as C0, and that no G1-G3 or C1 set is
invoked. This is specified in the definition of "GeneralString" in
[5], the definition of ASN.1 encoding (section 23.5.2).

If this is not a suitable initial state, a message must always start
with the necessary announcers and escape sequences to designate and
invoke the character sets that are actually used. The character sets
in use may be changed later in the message by use of escape
sequences.

The parameters of a GeneralText message always list all the character
sets used, by quoting their ISO reference numbers.

It is impossible to use a character set not registered with ISO in a
GeneralText message.

It is also impossible to decide on the true meaning of a byte in a
GeneralText message without scanning the whole message for shift and
escape sequences.

3.3. How to use the character sets

   RECOMMENDATION:

   When the text to be rendered is representable in one of the character
   sets of ISO-8859, the G0 set should be set to ISO 646 International
   Reference Version (1991), also called US-ASCII, ISO-IR-6.

   The older character set ISO-IR-2, ISO 646 IRV(1983), should NOT be
   used. This means that the escape sequence ESC 2/8 4/2 (designating
   US-ASCII as G0) should always occur at the beginning of the message.

   The G1 set should be set to the character set identified by the
   relevant ISO-8859 part. G2 and G3 are not used.

This corresponds to the first level of ISO 4873 usage.

For the currently defined parts of ISO 8859, the character set
designations for the G1 set are (relative to ISO 8859:1987):

```
Part    ISO IR name             Escape sequence Remarks
                                for G1 use

1       ISO-IR-100              Esc 2D 41    West Europe (Latin-1)
2       ISO-IR-101              Esc 2D 42    East European (Latin-2)
3       ISO-IR-109              Esc 2D 43    (Latin-3)
4       ISO-IR-110              Esc 2D 44    (Latin-4)
5       ISO-IR-144              Esc 2D 4C    Cyrillic
6       ISO-IR-127              Esc 2D 47    Arabic
7       ISO-IR-126              Esc 2D 46    Greek
8       ISO-IR-138              Esc 2D 48    Hebrew
9       ISO-IR-148              Esc 2D 4D    Turkish (Latin-5)
10      ISO-IR-157              Not listed   Sami (Latin-6)
```

The escape sequence for 8859-10 (Latin-6) is not listed in RFC 1345.

NOTE: The use of ISO 8859-3 and ISO 8859-4 is NOT recommended if
other possibilities exist.

NOTE: There is a debate about the Arabic and Hebrew character sets.
These languages are normally read right to left, but encodings have
been done in both "visual" (left to right) and "phonetic" (right to
left) ordering, there is significant disagreement about what the
"right" way to do it is, and the character sets mentioned do not
specify it. So, one should be careful not to use these character sets
until a standard is agreed upon, or the result will probably be
unreadable (siht ekil).

(Note that there is some confusion as to what parts are actually
standardized; the Norwegian standards institute reports that only
part 1, 2, 3, 4, 6, 7 and 8 are currently standards. Other reports
claim that both 8859-10 and 8859-11 are standards, and I definitely
think that 8859-9 is.)

NOTE: ISO has not ruled out the possibility of changing the ISO 8859
standard. This would involve changing the registry information in
this table, so this should be assumed valid for ISO 8859 versions
that are current in 1993.

The G1 set should be permanently shifted into the upper half of the
code page.

When the text is not representable in one of the ISO-8859 character
sets, the following rules may be applied:

  (1)   If any Latin characters are used, keep IA5 as the G0 set.

  (2)   If a mainstream character set is used (Greek, Cyrillic,
        Hebrew, Arabic), designate this as the G1 character set,
        and permanently shift this into the upper half of the code
        page (LS1R).
        EXCEPTION: The Japanese community has a long tradition of
        switching between the Japanese 16-bit character set
        ISO-IR-87 and US-ASCII as the G0 set. See [7]
        for details. If ISO-IR-87 is used, that technique should be
        used instead of the one recommended here.

  (3)   If occasional extensions to a character set that is
        basically Latin occur (like accents, national variants
        and so on), and these are available in a single character
        set, designate the relevant set as G2 and use single
        shift (SS2) to invoke characters from this character set.

        The ISO 8859 supplementary set, ISO-IR-154, is recommended
        for this purpose.

        This corresponds to the ISO 4873 "second level" application.

  (4)   If two non-Latin character sets are used, the second should
        be designated as G3, and shifted into the upper half of the
        code page by the use of Locking Shift 3 Right (LS3R).

        This corresponds to the ISO 4873 "third level" application.

  (5)   If avoidable, use of character sets with floating accents,
        like ISO 6937, should be avoided.

  (6)   The shifts changing the lower half of the code table (SI/SO,
        LS2 and LS3) should NOT be used.

RATIONALE: Keeping the G0 set reserved for US-ASCII will ensure that
text in US-ASCII has the same bit representation always.

The use of the upper code page for other scripts ensures that both
text in these languages and text of this type mixed with English can
be represented without the use of shift sequences.

If the language and/or content of a text is completely unknown,
chapter 5 gives an algorithm that may be used to decide upon the
character sets. This might, for instance, be suitable for use at

automatic mail gateways.

NOTE: At the time of this writing, few applications that use ISO 4873
level 2 and level 3 encoding exist. It has been estimated that
implementing them in an application that already uses a rich
repertoire of characters is a matter of programmer-days, not
programmer-months, but this has not been proven.

4.   GUIDELINES FOR THE RENDERING OF GENERALTEXT

As a basic rule, one should NOT assume that any of the rules above
are followed.

An user agent capable of rendering GeneralText should:

   (1)   ALWAYS be able to identify and render characters in IA5, no
         matter how they are designated and invoked.

   (2)   ALWAYS be able to identify and render characters in the
         "native" character sets, no matter how they are designated
         and invoked.

   (3)   ALWAYS indicate the presence of characters that cannot be
         adequately represented on the current output device.

   (4)   NEVER render a character in an unknown or unrepresentable
         character set by displaying the character in the same bit
         position in the native character set.

   (5)   PREFERABLY be able to identify and render characters that are
         the same as characters in the "native" character sets, even
         though they are designated and invoked as part of other
         character sets. This applies in particular to the
         "invariant" part of ISO 8859, parts 1 through 6.

   (6)   PREFERABLY be able to combine the floating accents of ISO
         6937 with their base characters for suitable rendering using
         the capabilities of the current output device.

   (7)   PREFERABLY be able to display text both in a mode using
         fallbacks for nonrenderable characters and in a mode
         designating nonrenderable characters as such.

   (8)   PREFERABLY be able to save the content of a GeneralText
         message to a file or other suitable media, saving all
         character set information, for later processing by other
         means. It is not illegal to render the character set
         information into a different format; however, it should be

noted that it is easy to lose vital information if the format
chosen for representing character sets does not offer the
possibility of referencing all character sets in the ISO
registry of character sets.

These requirements also apply to gateways that transform the message
into some other format, for example a gateway that transforms a
message into MIME using [7] for the purpose.

5.   RECOMMENDATION FOR SELECTION OF CHARACTER SETS

5.1. Algorithm for selection of character sets

When one has text in which characters from several character sets
occurs, and wants to process this into a GeneralText document, it is
often hard to guess right at the character sets to select.

The following paragraphs give an algorithm that can be started at the
beginning of a message, and at the end of it, return a set of
character sets that can be used as G0..G3 character sets, OR an
indication that the task is impossible.

 VARIABLES:

 UsedSets
      The set of character sets that MUST be used for this message

 UsableSets
      The set of character sets that MAY be used for this message.
      Each set also contains a counter for each character position.

 PossibleSets
      The set of all the character sets known to be usable in the
      destination format.

      ALGORITHM:

 1)   Add IA5 (ISO-IR-6) to the UsedSets (as G0).

 2)   Get the next character of the text. If the text is
      completely analyzed, go to FINISHED

 3)   If it is in the UsedSets, go to 2).

 4)   Find the set of character sets from PossibleSets in which the
      character occurs. If it does not occur in any, report
      failure.

5)   If it is in a single character set in PossibleSets only, add
      this set to UsedSets, and go to 2).

6)   If it is in more than one character set, add these to
      PossibleSets (if not already present), and increment the
      counter for that character in all the sets. Go to 2).

FINISHED)

1)   (FINAL SELECTION) Remove any character set in UsedSets from
      PossibleSets.

      Zero the counters for any character in PossibleSets that also
      occurs in UsedSets.
      WHILE (more characters left)
        Select one character set and move it from PossibleSets to
        UsedSqets.
        Zero the counters for all characters in this set in the other
        PossibleSets.
      END WHILE
      This step can be "tuned" any way you want, for instance by
      choosing the character sets most likely to be understood at
      the destination first, choosing the character sets covering
      the most characters first, avoiding multi-byte character sets
      as long as possible, or any other scheme suitable for the
      application.

5.2. WHAT TO DO ON FAILURE

Failure will occur in this schema if a character is found that is not
in the PossibleSets. It may then be handled in one of the following
ways:

(1) Replace the character with the SUB control character

(2) Replace the character with Keld Simonsen Mnemonics [8].
     This is a reversible transformation as long as the
     recipient is aware that it has been used, but requires
     passing out-of-band information to indicate this.

(3) Replace the lost characters with any suitable fallback or
     mnemonic scheme intended for human understanding

(4) Bounce the message/refuse the conversion/give up.

The action to be taken may be different based on the percentage of
"lost" characters.

If the message has "controls" like "conversion with loss prohibited",
only the last possibility may be used.

5.3. RECOMMENDED CHARACTER SETS

There are 2 steps in the algorithm above that are left for local
judgement:

   (1) Selection of the sets to appear in PossibleSets.

   (2) The algorithm for deciding which character set to select in
        step 9.

In the context of generating X.400 GeneralText messages, the
following is recommended:

```
 Sets in PossibleSets:
 ISO-IR-6        Esc 28 42 (G0)   US-ASCII, IA5, ISO646
 ISO-IR-100      Esc 2D 41 (G1)   ISO-8859-1    West Europe
 ISO-IR-101      Esc 2D 42 (G1)   ISO-8859-2    Central/Eastern Europe
 ISO-IR-144      Esc 2D 4C (G1)   ISO-8859-5    Cyrillic
 ISO-IR-127      Esc 2D 47 (G1)   ISO-8859-6    Arabic
 ISO-IR-126      Esc 2D 46 (G1)   ISO-8859-7    Greek
 ISO-IR-138      Esc 2D 48 (G1)   ISO-8859-8    Hebrew
 ISO-IR-148      Esc 2D 4D (G1)   ISO-8859-9    Turkish
```

The following multi-byte character sets are recommended:

```
 ISO-IR-87 (Japanese JIS C6226-1983)     Esc 24 29 42 (G1)
 ISO-IR-149 (Korean KS C 5601-1989)      Esc 24 29 43 (G1)
 ISO-IR-58 (Chinese GB 2312-80)          Esc 24 29 41 (G1)
```

It is a STRONG recommendation that character sets not listed above,
which do not add any new characters to the total set of characters
given by the character sets above, should NOT be used in X.400
interchange.

ISO-IR-87 is the Japanese character set that is allowed in a Teletex
string, such as the subject field.

NOTE: ISO-IR-87 has been "superseded" by ISO-IR-168, which allows two
extra Kanji characters. Any application that handles ISO-IR-87 should
also be able to handle ISO-IR-168.

Algorithm for selecting character sets:

Start at the top of the list above, and add each set only if it is
needed.

6. REFERENCES

    [1] Information technology - ISO 8-bit code for information
         interchange - Structure and rules for implementation, Third
         edition, 1991-12-15.

    [2] Information technology - 8-bit single-byte coded graphic
         character sets (parts 1-11; the parts have different dates, the
         ones referenced here are from RFC 1345).

    [3] Information technology - Coded graphic character set for text
         communication (parts 1 and 2; part 2 dated 1983-12-15).

    [4] Code for the representation of names of languages. 1988 version.

    [5] CCITT Recommendation X.209(1988): Specification of Basic
         Encoding Rules for Abstract Syntax Notation One (ASN.1).
         Technically aligned with ISO 8825 and ISO 8825/AD 1.

    [6] Information Technology - Universal Multiple-Octet Coded
         Character Set (UCS) - ISO 10646.

    [7] Murai, J., Crispin M., and E. van der Poel, "Japanese Character
         Encoding for Internet Message Bodies", RFC 1468, Keio
         University, Panda Programming, June 1993.

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

    [9] Information Technology - Text communication - Message- Oriented
         Text Interchange Systems (MOTIS) - ISO 10021 - October 1988.

    [10] ISO DIS documents describing X.400/84 with slight extensions.
         Now very hard to get copies of, since they failed to become
         ISes.

    [11] CCITT Recommendation X.420 (1988), Interpersonal Messaging
         System.

    [12] International Standard--Information Processing-- ISO 7-bit and
         8-bit coded character sets--Code extension techniques, ISO
         2022:1986.

7. Security Considerations

    Security issues are not discussed in this memo.

8. Author's Address

     Harald Tveit Alvestrand
     SINTEF DELAB
     N-7034 Trondheim
     NORWAY

     EMail: Harald.Alvestrand@delab.sintef.no