Network Working Group Nathaniel Borenstein
Internet Draft Ned Freed
<draft-ietf-822ext-mime-conf-02.txt>

Multipurpose Internet Mail Extensions (MIME) Part Five:

Conformance Criteria and Examples

August 27, 1995

Status of this Memo

1. Abstract

STD 11, RFC 822, defines a message representation protocol specifying considerable detail about US-ASCII message headers, and leaves the message content, or message body, as flat US-ASCII text. This set of documents, collectively called the Multipurpose Internet Mail Extensions, or MIME, redefines the format of messages to allow for

(1) textual message bodies in character sets other than US-ASCII,

(2) non-textual message bodies,

(3) multi-part message bodies, and

(4) textual header information in character sets other than US-ASCII.

These documents are based on earlier work documented in RFC 934, STD 11, and RFC 1049, but extends and revises them. Because RFC 822 said so little about message bodies, these documents are largely orthogonal to (rather than a revision of) RFC 822.

In particular, these documents are designed to provide facilities to include multiple parts in a single message, to represent body and header text in character sets other than US-ASCII, to represent formatted multi-font text messages, to represent non-textual material such as images and audio clips, and generally to facilitate later extensions defining new types of Internet mail for use by cooperating mail agents.

The initial document in this set, RFC MIME-IMB, specifies the various headers used to describe the structure of MIME messages. The second document defines the general structure of the MIME media typing system and defines an initial set of media types. The third document, RFC MIME-HEADERS, describes extensions to RFC 822 to allow non-US-ASCII text data in Internet mail header fields. The fourth document, RFC MIME-REG, specifies various IANA registration procedures for MIME-related facilities. This fifth and final document describes MIME conformance criteria as well as providing some illustrative examples of MIME message formats, acknowledgements, and the bibliography.

These documents are revisions of RFCs 1521, 1522, and 1590, which themselves were revisions of RFCs 1341 and 1342. Appendix B of this document describes differences and changes from previous versions.

2. Table of Contents

3. Introduction

The first and second documents in this set defined MIME header field and the initial set of MIME media types. This document describes what portions of MIME must be supported by a conformant MIME implementation. It also describes various pitfalls of contemporary messaging systems as well as the canonical encoding model MIME is based on.

4. MIME Conformance

The mechanisms described in these documents are open-ended. It is definitely not expected that all implementations will support all available media types, nor that they will all share the same extensions. In order to promote interoperability, however, it is useful to define the concept of "MIME-conformance" to define a certain level of implementation that allows the useful interworking of messages with content that differs from US-ASCII text. In this section, we specify the requirements for such conformance.

A mail user agent that is MIME-conformant MUST:

(1) Always generate a "MIME-Version: 1.0" header field in any message it creates.

(2) Recognize the Content-Transfer-Encoding header field and decode all received data encoded by either quoted-printable or base64 implementations. The identity transformations 7bit, 8bit, and binary must also be recognized.

Any non-7bit data that is sent without encoding must be properly labelled with a content-transfer-encoding of 8bit or binary, as appropriate. If the underlying transport does not support 8bit or binary (as SMTP [RFC821] does not), the sender is required to both encode and label data using an appropriate Content-Transfer-Encoding such as quoted-printable or base64.

Any unrecognized Content-Transfer-Encoding must be treated as it it had a Content-Type of "application/octet-stream", regardless of whether or not the actual Content-Type is recognized.

(3) Recognize and interpret the Content-Type header field, and avoid showing users raw data with a Content-Type field other than text. Be able to send at least text/plain messages, with the character set specified as a parameter if it is not US-ASCII.

(4) Explicitly handle the following media type values, to at least the following extents:

Text:

-- Recognize and display "text" mail with the character set "US-ASCII."

-- Recognize other character sets at least to the extent of being able to inform the user about what character set the message uses.

-- Recognize the "ISO-8859-*" character sets to the extent of being able to display those characters that are common to ISO-8859-* and US-ASCII, namely all characters represented by octet values 1-127.

-- For unrecognized subtypes in a known character set, show or offer to show the user the "raw" version of the data after conversion of the content from

canonical form to local form.

-- Treat material in an unknown character set as if it were "application/octet-stream".

Image, audio, and video:

-- At a minumum provide facilities to treat any unrecognized subtypes as if they were "application/octet-stream".

Application:

-- Offer the ability to remove either of the quoted-printable or base64 encodings defined in this document if they were used and put the resulting information in a user file.

Multipart:

-- Recognize the mixed subtype. Display all relevant information on the message level and the body part header level and then display or offer to display each of the body parts individually.

-- Recognize the "alternative" subtype, and avoid showing the user redundant parts of multipart/alternative mail.

-- Recognize the "multipart/digest" subtype, specifically using "message/rfc822" rather than "text/plain" as the default media type for body parts inside "multipart/digest" entities.

-- Treat any unrecognized subtypes as if they were "mixed".

Message:

-- Recognize and display at least the primary (RFC822) encapsulation in such a way as to preserve any recursive structure, that is, displaying or offering to display the encapsulated data in accordance with its media type.

-- Treat any unrecognized subtypes as if they were "application/octet-stream".

(5) Upon encountering any unrecognized Content-Type field, an implementation must treat it as if it had a media type of "application/octet-stream" with no parameter sub-arguments. How such data are handled is up to an implementation, but likely options for handling such unrecognized data include offering the user to write it into a file (decoded from its mail transport format) or offering the user to name a program to which the decoded data should be passed as input.

A user agent that meets the above conditions is said to be MIME-conformant. The meaning of this phrase is that it is assumed to be "safe" to send virtually any kind of properly-marked data to users of such mail systems, because such systems will at least be able to treat the data as undifferentiated binary, and will not simply splash it onto the screen of unsuspecting users.

There is another sense in which it is always "safe" to send data in a format that is MIME-conformant, which is that such data will not break or be broken by any known systems that are conformant with RFC 821 and RFC 822. User agents that are MIME-conformant have the additional guarantee that the user will not be shown data that were never intended to be viewed as text.

5. Guidelines for Sending Email Data

Internet email is not a perfect, homogeneous system. Mail may become corrupted at several stages in its travel to a final destination. Specifically, email sent throughout the Internet may travel across many networking technologies. Many networking and mail technologies do not support the full functionality possible in the SMTP transport environment. Mail traversing these systems is likely to be modified in order that it can be transported.

There exist many widely-deployed non-conformant MTAs in the Internet. These MTAs, speaking the SMTP protocol, alter messages on the fly to take advantage of the internal data structure of the hosts they are implemented on, or are just

plain broken.

The following guidelines may be useful to anyone devising a data format (media type) that is supposed to survive the widest range of networking technologies and known broken MTAs unscathed. Note that anything encoded in the base64 encoding will satisfy these rules, but that some well-known mechanisms, notably the UNIX uuencode facility, will not. Note also that anything encoded in the Quoted-Printable encoding will survive most gateways intact, but possibly not some gateways to systems that use the EBCDIC character set.

(1) Under some circumstances the encoding used for data may change as part of normal gateway or user agent operation. In particular, conversion from base64 to quoted-printable and vice versa may be necessary. This may result in the confusion of CRLF sequences with line breaks in text bodies. As such, the persistence of CRLF as something other than a line break must not be relied on.

(2) Many systems may elect to represent and store text data using local newline conventions. Local newline conventions may not match the RFC822 CRLF convention -- systems are known that use plain CR, plain LF, CRLF, or counted records. The result is that isolated CR and LF characters are not well tolerated in general; they may be lost or converted to delimiters on some systems, and hence must not be relied on.

(3) The transmission of NULs (US-ASCII value 0) is problematic in Internet mail. (This is largely the result of NULs being used as a termination character by many of the standard runtime library routines in the C programming language.) The practice of using NULs as termination characters is so entrenched now that messages should not rely on them being preserved.

(4) TAB (HT) characters may be misinterpreted or may be automatically converted to variable numbers of spaces. This is unavoidable in some environments, notably those not based on the US-ASCII character set. Such conversion is STRONGLY DISCOURAGED, but it may occur, and mail formats must not rely on the persistence of TAB (HT) characters.

(5) Lines longer than 76 characters may be wrapped or truncated in some environments. Line wrapping and line truncation are STRONGLY DISCOURAGED, but unavoidable in some cases. Applications which require long lines must somehow differentiate between soft and hard line breaks. (A simple way to do this is to use the quoted-printable encoding.)

(6) Trailing "white space" characters (SPACE, TAB (HT)) on a line may be discarded by some transport agents, while other transport agents may pad lines with these characters so that all lines in a mail file are of equal length. The persistence of trailing white space, therefore, must not be relied on.

(7) Many mail domains use variations on the US-ASCII character set, or use character sets such as EBCDIC which contain most but not all of the US-ASCII characters. The correct translation of characters not in the "invariant" set cannot be depended on across character converting gateways. For example, this situation is a problem when sending uuencoded information across BITNET, an EBCDIC system. Similar problems can occur without crossing a gateway, since many Internet hosts use character sets other than US-ASCII internally. The definition of Printable Strings in X.400 adds further restrictions in certain special cases. In particular, the only characters that are known to be consistent across all gateways are the 73 characters that correspond to the upper and lower case letters A-Z and a-z, the 10 digits 0-9, and the following eleven special characters:

"'" (US-ASCII decimal value 39)
"(" (US-ASCII decimal value 40)
")" (US-ASCII decimal value 41)
"+" (US-ASCII decimal value 43)
"," (US-ASCII decimal value 44)
"-" (US-ASCII decimal value 45)
"." (US-ASCII decimal value 46)
"/" (US-ASCII decimal value 47)
":" (US-ASCII decimal value 58)
"=" (US-ASCII decimal value 61)
"?" (US-ASCII decimal value 63)

A maximally portable mail representation will confine itself to relatively short lines of text in which the only meaningful characters are taken from this set of 73 characters. The base64 encoding follows this rule.

(8) Some mail transport agents will corrupt data that includes certain literal strings. In particular, a period (".") alone on a line is known to be corrupted by some (incorrect) SMTP implementations, and a line that starts with the five characters "From " (the fifth character is a SPACE) are commonly corrupted as well. A careful composition agent can prevent these corruptions by encoding the data (e.g., in the quoted-printable encoding using "=46rom " in place of "From " at the start of a line, and "=2E" in place of "." alone on a line).

Please note that the above list is NOT a list of recommended practices for MTAs. RFC 821 MTAs are prohibited from altering the character of white space or wrapping long lines. These BAD and invalid practices are known to occur on established networks, and implementations should be robust in dealing with the bad effects they can cause.

6. Canonical Encoding Model

There was some confusion, in earlier drafts of these documents, regarding the model for when email data was to be converted to canonical form and encoded, and in particular how this process would affect the treatment of CRLFs, given that the representation of newlines varies greatly from system to system. For this reason, a canonical model for encoding is presented below.

The process of composing a MIME entity can be modeled as being done in a number of steps. Note that these steps are roughly similar to those steps used in PEM [RFC1421] and are performed for each "innermost level" body:

(1) Creation of local form.

The body to be transmitted is created in the system's native format. The native character set is used and, where appropriate, local end of line conventions are

used as well. The body may be a UNIX-style text file, or a Sun raster image, or a VMS indexed file, or audio data in a system-dependent format stored only in memory, or anything else that corresponds to the local model for the representation of some form of information. Fundamentally, the data is created in the "native" form that corresponds to the type specified by the media type.

(2) Conversion to canonical form.

The entire body, including "out-of-band" information such as record lengths and possibly file attribute information, is converted to a universal canonical form. The specific media type of the body as well as its associated attributes dictate the nature of the canonical form that is used. Conversion to the proper canonical form may involve character set conversion, transformation of audio data, compression, or various other operations specific to the various media types. If character set conversion is involved, however, care must be taken to understand the semantics of the media type, which may have strong implications for any character set conversion, e.g. with regard to syntactically meaningful characters in a text subtype other than "plain".

For example, in the case of text/plain data, the text must be converted to a supported character set and lines must be delimited with CRLF delimiters in accordance with RFC 822. Note that the restriction on line lengths implied by RFC 822 is eliminated if the next step employs either quoted-printable or base64 encoding.

(3) Apply transfer encoding.

A Content-Transfer-Encoding appropriate for this body is applied. Note that there is no fixed relationship between the media type and the transfer encoding. In particular, it may be appropriate to base the choice of base64 or quoted-printable on character frequency counts which are specific to a given instance of a body.

(4) Insertion into entity.

The encoded object is inserted into a MIME entity with appropriate headers. The entity is then inserted into the body of a higher-level entity (message or multipart) if needed.

It is vital to note that these steps are only a model; they are specifically NOT a blueprint for how an actual system would be built. In particular, the model fails to account for two common designs:

(1) In many cases the conversion to a canonical form prior to encoding will be subsumed into the encoder itself, which understands local formats directly. For example, the local newline convention for text bodies might be carried through to the encoder itself along with knowledge of what that format is.

(2) The output of the encoders may have to pass through one or more additional steps prior to being transmitted as a message. As such, the output of the encoder may not be conformant with the formats specified by RFC 822. In particular, once again it may be appropriate for the converter's output to be expressed using local newline conventions rather than using the standard RFC 822 CRLF delimiters.

Other implementation variations are conceivable as well. The vital aspect of this discussion is that, in spite of any optimizations, collapsings of required steps, or insertion of additional processing, the resulting messages must be consistent with those produced by the model described here. For example, a message with the following header fields:

```
Content-type: text/foo; charset=bar
Content-Transfer-Encoding: base64
```

must be first represented in the text/foo form, then (if necessary) represented in the "bar" character set, and finally transformed via the base64 algorithm into a mail-safe form.

NOTE: Some confusion has been caused by systems that represent messages in a format which uses local newline conventions which differ from the RFC822 CRLF convention. It is important

to note that these formats are not canonical RFC822/MIME. These formats are instead *encodings* of RFC822, where CRLF sequences in the canonical representation of the message are encoded as the local newline convention. Note that formats which encode CRLF sequences as, for example, LF are not capable of representing MIME messages containing binary data which contains LF octets not part of CRLF line separation sequences.

7. Summary

This document defines what is meant by MIME Conformance. It also details various problems known to exist in the Internet email system and how to use MIME to overcome them. Finally, it describes MIME's canonical encoding model.

8. Security Considerations

Security issues are discussed in the second document in this set, RFC MIME-IMT.

9. Authors' Addresses

For more information, the authors of this document are best contacted via Internet mail:

Nathaniel S. Borenstein
First Virtual Holdings
25 Washington Avenue
Morristown, NJ 07960
USA

Email: nsb@nsb.fv.com
Phone: +1 201 540 8967
Fax: +1 201 993 3032

Ned Freed
Innosoft International, Inc.
1050 East Garvey Avenue South
West Covina, CA 91790
USA

Email: ned@innosoft.com
Phone: +1 818 919 3600
Fax: +1 818 919 3614

MIME is a result of the work of the Internet Engineering Task Force Working Group on Email Extensions. The chairman of that group, Greg Vaudreuil, may be reached at:

Gregory M. Vaudreuil
Tigon Corporation
17060 **Dallas Parkway**
Dallas Texas, 75248

Email: greg.vaudreuil@ons.octel.com
Phone: +1 214 733 2722

10. Acknowledgements

This document is the result of the collective effort of a large number of people, at several IETF meetings, on the IETF-SMTP and IETF-822 mailing lists, and elsewhere. Although any enumeration seems doomed to suffer from egregious omissions, the following are among the many contributors to this effort:

Harald Tveit Alvestrand	Marc Andreessen
Randall Atkinson	Bob Braden
Philippe Brandon	Brian Capouch
Kevin Carosso	Uhhyung Choi
Peter Clitherow	Dave Collier-Brown
Cristian Constantinof	John Coonrod
Mark Crispin	Dave Crocker
Stephen Crocker	Terry Crowley
Walt Daniels	Jim Davis
Frank Dawson	Axel Deininger
Hitoshi Doi	Kevin Donnelly
Steve Dorner	Keith Edwards
Chris Eich	Dana S. Emery
Johnny Eriksson	Craig Everhart
Patrik Faltstrom	Erik E. Fair
Roger Fajman	Alain Fontaine
Martin Forssen	James M. Galvin
Stephen Gildea	Philip Gladstone
Thomas Gordon	Keld Simonsen
Terry Gray	Phill Gross
James Hamilton	David Herron
Mark Horton	Bruce Howard
Bill Janssen	Olle Jarnefors
Risto Kankkunen	Phil Karn
Alan Katz	Tim Kehres
Neil Katin	Steve Kille
Kyuho Kim	Anders Klemets
John Klensin	Valdis Kletniek
Jim Knowles	Stev Knowles
Bob Kummerfeld	Pekka Kytolaakso
Stellan Lagerstrom	Vincent Lau
Timo Lehtinen	Donald Lindsay
Warner Losh	Carlyn Lowery
Laurence Lundblade	Charles Lynn
John R. MacMillan	Larry Masinter
Rick McGowan	Michael J. McInerny

Leo Mclaughlin	Goli Montaser-Kohsari
Tom Moore	John Gardiner Myers
Erik Naggum	Mark Needleman
Chris Newman	John Noerenberg
Mats Ohrman	Julian Onions
Michael Patton	David J. Pepper
Erik van der Poel	Blake C. Ramsdell
Christer Romson	Luc Rooijakkers
Marshall T. Rose	Jonathan Rosenberg
Guido van Rossum	Jan Rynning
Harri Salminen	Michael Sanderson
Yutaka Sato	Markku Savela
Richard Alan Schafer	Masahiro Sekiguchi
Mark Sherman	Bob Smart
Peter Speck	Henry Spencer
Einar Stefferud	Michael Stein
Klaus Steinberger	Peter Svanberg
James Thompson	Steve Uhler
Stuart Vance	Peter Vanderbilt
Greg Vaudreuil	Ed Vielmetti
Larry W. Virden	Ryan Waldron
Rhys Weatherly	Jay Weber
Dave Wecker	Wally Wedel
Sven-Ove Westberg	Brian Wideen
John Wobus	Glenn Wright
Rayan Zachariassen	David Zimmerman

The authors apologize for any omissions from this list, which are certainly unintentional.

Appendix A -- A Complex Multipart Example

What follows is the outline of a complex multipart message. This message has five parts to be displayed serially: two introductory plain text parts, an embedded multipart message, a text/enriched part, and a closing encapsulated text message in a non-ASCII character set. The embedded multipart message has two parts to be displayed in parallel, a picture and an audio fragment.

```
MIME-Version: 1.0
From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Date: Fri, 07 Oct 1994 16:15:05 -0700 (PDT)
Subject: A multipart example
Content-Type: multipart/mixed;
              boundary=unique-boundary-1

This is the preamble area of a multipart message.
Mail readers that understand multipart format
should ignore this preamble.

If you are reading this text, you might want to
consider changing to a mail reader that understands
how to properly display multipart messages.

--unique-boundary-1

  ... Some text appears here ...

[Note that the blank between the boundary and the start
 of the text in this part means no header fields were
 given and this is text in the US-ASCII character set.
 It could have been done with explicit typing as in the
 next part.]

--unique-boundary-1
Content-type: text/plain; charset=US-ASCII

This could have been part of the previous part, but
illustrates explicit versus implicit typing of body
parts.
```

```
--unique-boundary-1
Content-Type: multipart/parallel; boundary=unique-boundary-2

--unique-boundary-2
Content-Type: audio/basic
Content-Transfer-Encoding: base64

  ... base64-encoded 8000 Hz single-channel
      mu-law-format audio data goes here ...

--unique-boundary-2
Content-Type: image/jpeg
Content-Transfer-Encoding: base64

  ... base64-encoded image data goes here ...

--unique-boundary-2--

--unique-boundary-1
Content-type: text/enriched

This is <bold><italic>enriched.</italic></bold>
<smaller>as defined in RFC 1563</smaller>

Isn't it
<bigger><bigger>cool?</bigger></bigger>

--unique-boundary-1
Content-Type: message/rfc822

From: (mailbox in US-ASCII)
To: (address in US-ASCII)
Subject: (subject in US-ASCII)
Content-Type: Text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: Quoted-printable

  ... Additional text in ISO-8859-1 goes here ...

--unique-boundary-1--
```

Appendix B -- Changes from RFC 1521, 1522, and 1590

These documents are a revision of RFC 1521, 1522, and 1590. For the convenience of those familiar with the earlier documents, the changes from those documents are summarized in this appendix. For further history, note that Appendix H in RFC 1521 specified how that document differed from its predecessor, RFC 1341.

(1) This document has been completely reformatted and split into multiple documents. This was done to improve the quality of the plain text version of this document, which is required to be the reference copy.

(2) BNF describing the overall structure of MIME message and part headers has been added. This is a documentation change only -- the underlying syntax has not changed in any way.

(3) The specific BNF for the seven media types in MIME has been removed. This BNF was incorrect, incomplete, amd inconsistent with the type-indendependent BNF. And since the type-independent BNF already fully specifies the syntax of the various MIME headers, the type-specific BNF was, in the final analysis, completely unnecessary and caused more problems than it solved.

(4) The more specific "US-ASCII" character set name has replaced the use of the term ASCII in many parts of this specification.

(5) The informal concept of a primary subtype has been removed.

(6) The term "object" was being used inconsistently. This term has been replaced with the more precise terms "body", "body part", and "entity" where appropriate.

(7) The BNF for the multipart media type has been rearranged to make it clear that the CRLF preceeding the boundary marker is actually part of the marker itself rather than the preceeding body part.

(8) The prose and BNF describing the multipart media type have been changed to make it clear that the body parts within a multipart entity MUST NOT contain any lines beginning with the boundary parameter string.

(9) In the rules on reassembling "message/partial" MIME entities, "Subject" is added to the list of headers to take from the inner message, and the example is modified to clarify this point.

(10) In the discussion of the application/postscript type, an additional paragraph has been added warning about possible interoperability problems caused by embedding of binary data inside a PostScript MIME entity.

(11) Added a clarifying note to the basic syntax rules for the Content-Type header field to make it clear that the following two forms:

```
Content-type: text/plain; charset=us-ascii (comment)

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

are completely equivalent.

(12) The following sentence has been removed from the discussion of the MIME-Version header: "However, conformant software is encouraged to check the version number and at least warn the user if an unrecognized MIME-version is encountered."

(13) A typo was fixed that said "application/external-body" instead of "message/external-body".

(14) The definition of a character set has been reorganized to make the requirements clearer.

(15) The definition of the "image/gif" media type has been moved to a separate document. This change was made because of potential conflicts with IETF rules governing the standardization of patented technology.

(16) The definitions of "7bit" and "8bit" have been tightened so that use of bare CR, LF can only be used as end-of-line sequences. The document also no longer

requires that NUL characters be preserved, which brings MIME into alignment with real-world implementations.

(17) The definition of canonical text in MIME has been tightened so that line breaks must be represented by a CRLF sequence. CR and LF characters are not allowed outside of this usage. The definition of quoted-printable encoding has been altered accordingly.

(18) Prose was added to clarify the use of the "7bit", "8bit", and "binary" transfer-encodings on multipart or message entities encapsulating "8bit" or "binary" data.

(19) In the section on MIME Conformance, "multipart/digest" support was added to the list of requirements for minimal MIME conformance. Also, the requirement for "message/rfc822" support were strengthened to clarify the importance of recognizing recursive structure.

(20) The various restrictions on subtypes of "message" are now specified entirely on a subtype by subtype basis.

(21) The definition of "message/rfc822" was changed to indicate that at least one of the "From", "Subject", or "Date" headers must be present.

(22) The required handling of unrecognized subtypes as "application/octet-stream" has been made more explicit in both the type definitions sections and the conformance guidelines.

(23) Examples using text/richtext were changed to text/enriched.

(24) The BNF definition of subtype has been changed to make it clear that either an IANA registered subtype or a nonstandard "X-" subtype must be used in a Content-Type header field.

(25) The use of escape and shift mechanisms in the US-ASCII and ISO-8859-X character sets this specification defines has been clarified: Such mechanisms should never be used in conjunction with these character sets and their effect if they are used is undefined.

(26) The definition of the AFS access-type for message/external-body has been removed.

(27) Entities that are simply registered for use and those that are standardized by the IETF are now distinguished in the MIME BNF.

(28) The handling of the combination of multipart/alternative and message/external-body is now specifically addressed.

(29) Security issues specific to message/external-body are now discussed in some detail.

Appendix C -- References

[ATK]
Borenstein, Nathaniel S., Multimedia Applications Development with the Andrew Toolkit, Prentice-Hall, 1990.

[ISO-2022]
International Standard -- Information Processing -- ISO 7-bit and 8-bit Coded Character Sets -- Code Extension Techniques, ISO 2022:1986.

[ISO-8859]
International Standard -- Information Processing -- 8-bit Single-Byte Coded Graphic Character Sets -- Part 1: Latin Alphabet No. 1, ISO 8859-1:1987. Part 2: Latin alphabet No. 2, ISO 8859-2, 1987. Part 3: Latin alphabet No. 3, ISO 8859-3, 1988. Part 4: Latin alphabet No. 4, ISO 8859-4, 1988. Part 5: Latin/Cyrillic alphabet, ISO 8859-5, 1988. Part 6: Latin/Arabic alphabet, ISO 8859-6, 1987. Part 7: Latin/Greek alphabet, ISO 8859-7, 1987. Part 8: Latin/Hebrew alphabet, ISO 8859-8, 1988. Part 9: Latin alphabet No. 5, ISO 8859-9, 1990.

[ISO-646]
International Standard -- Information Processing -- ISO 7-bit Coded Character Set For Information Interchange, ISO 646:1983.

[JPEG]
JPEG Draft Standard ISO 10918-1 CD.

[MPEG]
Video Coding Draft Standard ISO 11172 CD, ISO IEC/JTC1/SC2/WG11 (Motion Picture Experts Group), May, 1991.

[PCM]
CCITT, Fascicle III.4 - Recommendation G.711, "Pulse Code Modulation (PCM) of Voice Frequencies", Geneva, 1972.

[POSTSCRIPT]
Adobe Systems, Inc., PostScript Language Reference Manual, Addison-Wesley, 1985.

[POSTSCRIPT2]
Adobe Systems, Inc., PostScript Language Reference Manual, Addison-Wesley, Second Edition, 1990.

[RFC-783]
Sollins, K.R., "TFTP Protocol (revision 2)", RFC-783, MIT, June 1981.

[RFC-821]
Postel, J.B., "Simple Mail Transfer Protocol", STD 10, RFC 821, USC/Information Sciences Institute, August 1982.

[RFC-822]
Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, RFC 822, UDEL, August 1982.

[RFC-934]
Rose, M. and E. Stefferud, "Proposed Standard for Message Encapsulation", RFC 934, Delaware and NMA, January 1985.

[RFC-959]
Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC 959, USC/Information Sciences Institute, October 1985.

[RFC-1049]
Sirbu, M., "Content-Type Header Field for Internet Messages", RFC 1049, CMU, March 1988.

[RFC-1154]
Robinson, D. and R. Ullmann, "Encoding Header Field for Internet Messages", RFC 1154, Prime Computer, Inc., April 1990.

[RFC-1341]
Borenstein, N., and N. Freed, "MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies", RFC 1341, Bellcore, Innosoft, June 1992.

[RFC-1342]
Moore, K., "Representation of Non-Ascii Text in Internet Message Headers", RFC 1342, University of Tennessee, June 1992.

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

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

[RFC-1421]
Linn, J., "Privacy Enhancement for Internet Electronic Mail: Part I -- Message Encryption and Authentication Procedures", RFC 1421, IAB IRTF PSRG, IETF PEM WG, February 1993.

[RFC-1422]
Kent, S., "Privacy Enhancement for Internet Electronic Mail: Part II -- Certificate-Based Key Management", RFC 1422, IAB IRTF PSRG, IETF PEM WG, February 1993.

[RFC-1423]
Balenson, D., "Privacy Enhancement for Internet Electronic Mail: Part III -- Algorithms, Modes, and Identifiers", IAB IRTF PSRG, IETF PEM WG, February 1993.

[RFC-1424]
Kaliski, B., "Privacy Enhancement for Internet Electronic Mail: Part IV -- Key Certification and Related Services", IAB IRTF PSRG, IETF PEM WG, February 1993.

[RFC-1521]
Borenstein, N. and Freed, N., "MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies", RFC 1521, Bellcore, Innosoft, September, 1993.

[RFC-1522]
Moore, K., "Representation of Non-ASCII Text in Internet Message Headers", RFC 1522, University of Tennessee, September 1993.

[RFC-1524]
Borenstein, N., "A User Agent Configuration Mechanism for Multimedia Mail Format Information", RFC 1524, Bellcore, September 1993.

[RFC-1543]
Postel, J., "Instructions to RFC Authors", RFC 1543, USC/Information Sciences Institute, October 1993.

[RFC-1563]
Borenstein, N., "The text/enriched MIME Content-type", RFC 1563, Bellcore, January, 1994.

[RFC-1590]
Postel, J., "Media Type Registration Procedure", RFC 1590, USC/Information Sciences Institute, March 1994.

[RFC-1602]
Internet Architecture Board, Internet Engineering Steering Group, Huitema, C., Gross, P., "The Internet Standards Process -- Revision 2", March 1994.

[RFC-1652]
Klensin, J., (WG Chair), Freed, N., (Editor), Rose, M., Stefferud, E., and Crocker, D., "SMTP Service Extension for 8bit-MIME transport", RFC 1652, United Nations University, Innosoft, Dover Beach Consulting, Inc., Network Management Associates, Inc., The Branch Office, March 1994.

[RFC-1700]
Reynolds, J. and J. Postel, "Assigned Numbers", STD 2, RFC 1700, USC/Information Sciences Institute, October 1994.

[RFC-1741]
Faltstrom, P., Crocker, D., and Fair, E., "MIME Content Type for BinHex Encoded Files", December 1994.

[RFC-MIME-IMB]
Borenstein, N. and Freed, N., "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC MIME-IMB, Bellcore, Innosoft, August, 1995.

[RFC-MIME-IMT]
Borenstein, N. and Freed, N., "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC MIME-IMT, Bellcore, Innosoft, August, 1995.

[RFC-MIME-HEADERS]
Moore, K., "Multipurpose Internet Mail Extensions (MIME) Part Three: Representation of Non-Ascii Text in Internet Message Headers", RFC MIME-HEADERS, University of Tennessee, ?.

[RFC-MIME-REG]
Postel, J. and Freed, N., "Multipurpose Internet Mail Extensions (MIME) Part Four: Media Type Registration Procedure", RFC MIME-REG, ISI, Innosoft, August, 1995.

[RFC-MIME-CONF]
Borenstein, N. and Freed, N., "Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples", RFC MIME-CONF, Bellcore, Innosoft, August, 1995.

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

[X400]
Schicker, Pietro, "Message Handling Systems, X.400", Message Handling Systems and Distributed Applications, E. Stefferud, O-j. Jacobsen, and P. Schicker, eds., North-Holland, 1989, pp. 3-41.