

Network Working Group  
Request for Comments: 1522  
Obsoletes: 1342  
Category: Standards Track

K. Moore  
University of Tennessee  
September 1993

MIME (Multipurpose Internet Mail Extensions) Part Two:  
Message Header Extensions for Non-ASCII Text

Status of this Memo

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

Abstract

This memo describes an extension to the message format defined in RFC 1521 [1], to allow the representation of character sets other than ASCII in RFC 822 (STD 11) message headers. The extensions described were designed to be highly compatible with existing Internet mail handling software, and to be easily implemented in mail readers that support RFC 1521.

## 1. Introduction

RFC 1521 describes a mechanism for denoting textual body parts which are coded in various character sets, as well as methods for encoding such body parts as sequences of printable ASCII characters. This memo describes similar techniques to allow the encoding of non-ASCII text in various portions of a RFC 822 [2] message header, in a manner which is unlikely to confuse existing message handling software.

Like the encoding techniques described in RFC 1521, the techniques outlined here were designed to allow the use of non-ASCII characters in message headers in a way which is unlikely to be disturbed by the quirks of existing Internet mail handling programs. In particular, some mail relaying programs are known to (a) delete some message header fields while retaining others, (b) rearrange the order of addresses in To or Cc fields, (c) rearrange the (vertical) order of header fields, and/or (d) "wrap" message headers at different places than those in the original message. In addition, some mail reading programs are known to have difficulty correctly parsing message headers which, while legal according to RFC 822, make use of backslash-quoting to "hide" special characters such as "<", ",", or ":", or which exploit other infrequently-used features of that

specification.

While it is unfortunate that these programs do not correctly interpret RFC 822 headers, to "break" these programs would cause severe operational problems for the Internet mail system. The extensions described in this memo therefore do not rely on little-used features of RFC 822.

Instead, certain sequences of "ordinary" printable ASCII characters (known as "encoded-words") are reserved for use as encoded data. The syntax of encoded-words is such that they are unlikely to "accidentally" appear as normal text in message headers. Furthermore, the characters used in encoded-words are restricted to those which do not have special meanings in the context in which the encoded-word appears.

Generally, an "encoded-word" is a sequence of printable ASCII characters that begins with "=?", ends with "=?", and has two "?"s in between. It specifies a character set and an encoding method, and also includes the original text encoded as graphic ASCII characters, according to the rules for that encoding method.

A mail composer that implements this specification will provide a means of inputting non-ASCII text in header fields, but will translate these fields (or appropriate portions of these fields) into encoded-words before inserting them into the message header.

A mail reader that implements this specification will recognize encoded-words when they appear in certain portions of the message header. Instead of displaying the encoded-word "as is", it will reverse the encoding and display the original text in the designated character set.

#### NOTES

This memo relies heavily on notation and terms defined STD 11, RFC 822 and RFC 1521. In particular, the syntax for the ABNF used in this memo is defined in STD 11, RFC 822, as well as many of the terms used in the grammar for the header extensions defined here. Successful implementation of this protocol extension requires careful attention to the details of both STD 11, RFC 822 and RFC 1521.

When the term "ASCII" appears in this memo, it refers to the "7-Bit American Standard Code for Information Interchange", ANSI X3.4-1986. The MIME charset name for this character set is "US-ASCII". When not specifically referring to the MIME charset name, this document uses the term "ASCII", both for brevity and for

consistency with STD 11, RFC 822. However, implementors are warned that the character set name must be spelled "US-ASCII" in MIME message and body part headers.

## 2. Syntax of encoded-words

An "encoded-word" is defined by the following ABNF grammar. The notation of RFC 822 is used, with the exception that white space characters MAY NOT appear between components of an encoded-word.

```
encoded-word = "=?" charset "?" encoding "?" encoded-text "!="
```

```
charset = token ; see section 3
```

```
encoding = token ; see section 4
```

```
token = 1*<Any CHAR except SPACE, CTLs, and specials>
```

```
specials = "(" / ")" / "<" / ">" / "@" / "," / ";" / ":" / "<" / ">" / "/" / "[" / "]" / "?" / "." / "="
```

```
encoded-text = 1*<Any printable ASCII character other
                than "!" or SPACE>
                ; (but see "Use of encoded-words in message
                ; headers", section 5)
```

Both "encoding" and "charset" names are case-independent. Thus the charset name "ISO-8859-1" is equivalent to "iso-8859-1", and the encoding named "Q" may be spelled either "Q" or "q".

An encoded-word may not be more than 75 characters long, including charset, encoding, encoded-text, and delimiters. If it is desirable to encode more text than will fit in an encoded-word of 75 characters, multiple encoded-words (separated by CRLF SPACE) may be used.

While there is no limit to the length of a multiple-line header field, each line of a header field that contains one or more encoded-words is limited to 76 characters.

The length restrictions are included not only to ease interoperability through internetwork mail gateways, but also to impose a limit on the amount of lookahead a header parser must employ (while looking for a final != delimiter) before it can decide whether a token is an encoded-word or something else.

The characters which may appear in encoded-text are further restricted by the rules in section 5.

### 3. Character sets

The "charset" portion of an encoded-word specifies the character set associated with the unencoded text. A charset can be any of the character set names allowed in an RFC 1521 "charset" parameter of a "text/plain" body part, or any character set name registered with IANA for use with the MIME text/plain content-type [3]. (See section 7.1.1 of RFC 1521 for a list of charsets defined in that document).

Some character sets use code-switching techniques to switch between "ASCII mode" and other modes. If unencoded text in an encoded-word contains control codes to switch out of ASCII mode, it must also contain additional control codes such that ASCII mode is again selected at the end of the encoded-word. (This rule applies separately to each encoded-word, including adjacent encoded-words within a single header field.)

When there is a possibility of using more than one character set to represent the text in an encoded-word, and in the absence of private agreements between sender and recipients of a message, it is recommended that members of the ISO-8859-\* series be used in preference to other character sets.

### 4. Encodings

Initially, the legal values for "encoding" are "Q" and "B". These encodings are described below. The "Q" encoding is recommended for use when most of the characters to be encoded are in the ASCII character set; otherwise, the "B" encoding should be used. Nevertheless, a mail reader which claims to recognize encoded-words MUST be able to accept either encoding for any character set which it supports.

Only a subset of the printable ASCII characters may be used in encoded-text. Space and tab characters are not allowed, so that the beginning and end of an encoded-word are obvious. The "?" character is used within an encoded-word to separate the various portions of the encoded-word from one another, and thus cannot appear in the encoded-text portion. Other characters are also illegal in certain contexts. For example, an encoded-word in a "phrase" preceding an address in a From header field may not contain any of the "specials" defined in RFC 822. Finally, certain other characters are disallowed in some contexts, to ensure reliability for messages that pass through internetwork mail gateways.

The "B" encoding automatically meets these requirements. The "Q" encoding allows a wide range of printable characters to be used in non-critical locations in the message header (e.g., Subject), with

fewer characters available for use in other locations.

#### 4.1. The "B" encoding

The "B" encoding is identical to the "BASE64" encoding defined by RFC 1521.

#### 4.2. The "Q" encoding

The "Q" encoding is similar to the "Quoted-Printable" content-transfer-encoding defined in RFC 1521. It is designed to allow text containing mostly ASCII characters to be decipherable on an ASCII terminal without decoding.

- (1) Any 8-bit value may be represented by a "=" followed by two hexadecimal digits. For example, if the character set in use were ISO-8859-1, the "=" character would thus be encoded as "=3D", and a SPACE by "=20". (Upper case should be used for hexadecimal digits "A" through "F".)
- (2) The 8-bit hexadecimal value 20 (e.g., ISO-8859-1 SPACE) may be represented as "\_" (underscore, ASCII 95.). (This character may not pass through some internetwork mail gateways, but its use will greatly enhance readability of "Q" encoded data with mail readers that do not support this encoding.) Note that the "\_" always represents hexadecimal 20, even if the SPACE character occupies a different code position in the character set in use.
- (3) 8-bit values which correspond to printable ASCII characters other than "=", "?", "\_" (underscore), and SPACE may be represented as those characters. (But see section 5 for restrictions.)

### 5. Use of encoded-words in message headers

An encoded-word may appear in a message header or body part header according to the following rules:

- (1) An encoded-word may replace a "text" token (as defined by RFC 822) in any Subject or Comments header field, any extension message header field, or any RFC 1521 body part field for which the field body is defined as "\*text". An encoded-word may also appear in any user-defined ("X-") message or body part header field.

Ordinary ASCII text and encoded-words may appear together in the same header field. However, an encoded-word that appears in a header field defined as "\*text" MUST be separated from any adjacent encoded-word or "text" by linear-white-space.

- (2) An encoded-word may appear within a comment delimited by "(" and ")", i.e., wherever a "ctext" is allowed. More precisely, the RFC 822 ABNF definition for "comment" is amended as follows:

```
comment = "(" *(ctext / quoted-pair / comment / encoded-word) ")"
```

A "Q"-encoded encoded-word which appears in a comment MUST NOT contain the characters "(" , ")" or " encoded-word that appears in a "comment" MUST be separated from any adjacent encoded-word or "ctext" by linear-white-space.

- (3) As a replacement for a "word" entity within a "phrase", for example, one that precedes an address in a From, To, or Cc header. The ABNF definition for phrase from RFC 822 thus becomes:

```
phrase = 1*(encoded-word / word)
```

In this case the set of characters that may be used in a "Q"-encoded encoded-word is restricted to: <upper and lower case ASCII letters, decimal digits, "!", "\*", "+", "-", "/", "=", and "\_" (underscore, ASCII 95.)>. An encoded-word that appears within a "phrase" MUST be separated from any adjacent "word", "text" or "special" by linear-white-space.

These are the ONLY locations where an encoded-word may appear. In particular, an encoded-word MUST NOT appear in any portion of an "addr-spec". In addition, an encoded-word MUST NOT be used in a Received header field.

Each encoded-word MUST encode an integral number of octets. The encoded-text in each encoded-word must be well-formed according to the encoding specified; the encoded-text may not be continued in the next encoded-word. (For example, "=?charset?Q?=? =?charset?Q?AB?=" would be illegal, because the two hex digits "AB" must follow the "=" in the same encoded-word.)

Each encoded-word MUST represent an integral number of characters. A multi-octet character may not be split across adjacent encoded-words.

Only printable and white space character data should be encoded using this scheme. However, since these encoding schemes allow the encoding of arbitrary octet values, mail readers that implement this decoding should also ensure that display of the decoded data on the recipient's terminal will not cause unwanted side-effects.

Use of these methods to encode non-textual data (e.g., pictures or sounds) is not defined by this memo. Use of encoded-words to

represent strings of purely ASCII characters is allowed, but discouraged. In rare cases it may be necessary to encode ordinary text that looks like an encoded-word.

## 6. Support of encoded-words by mail readers

### 6.1. Recognition of encoded-words in message headers

A mail reader must parse the message and body part headers according to the rules in RFC 822 to correctly recognize encoded-words.

Encoded-words are to be recognized as follows:

- (1) Any message or body part header field defined as "\*text", or any user-defined header field, should be parsed as follows: Beginning at the start of the field-body and immediately following each occurrence of linear-white-space, each sequence of up to 75 printable characters (not containing any linear-white-space) should be examined to see if it is an encoded-word according to the syntax rules in section 2. Any other sequence of printable characters should be treated as ordinary ASCII text.
- (2) Any header field not defined as "\*text" should be parsed according to the syntax rules for that header field. However, any "word" that appears within a "phrase" should be treated as an encoded-word if it meets the syntax rules in section 2. Otherwise it should be treated as an ordinary "word".
- (3) Within a "comment", any sequence of up to 75 printable characters (not containing linear-white-space), that meets the syntax rules in section 2, should be treated as an encoded-word. Otherwise it should be treated as normal comment text.

### 6.2. Display of encoded-words

Any encoded-words so recognized are decoded, and if possible, the resulting unencoded text is displayed in the original character set.

When displaying a particular header field that contains multiple encoded-words, any linear-white-space that separates a pair of adjacent encoded-words is ignored. (This is to allow the use of multiple encoded-words to represent long strings of unencoded text, without having to separate encoded-words where spaces occur in the unencoded text.)

In the event other encodings are defined in the future, and the mail reader does not support the encoding used, it may either (a) display the encoded-word as ordinary text, or (b) substitute an appropriate

message indicating that the text could not be decoded.

If the mail reader does not support the character set used, it may (a) display the encoded-word as ordinary text (i.e., as it appears in the header), (b) make a "best effort" to display using such characters as are available, or (c) substitute an appropriate message indicating that the decoded text could not be displayed.

If the character set being used employs code-switching techniques, display of the encoded text implicitly begins in "ASCII mode". In addition, the mail reader must ensure that the output device is once again in "ASCII mode" after the encoded-word is displayed.

### 6.3. Mail reader handling of incorrectly formed encoded-words

It is possible that an encoded-word that is legal according to the syntax defined in section 2, is incorrectly formed according to the rules for the encoding being used. For example:

- (1) An encoded-word which contains characters which are not legal for a particular encoding (for example, a '-' in the "B" encoding), is incorrectly formed.
- (2) Any encoded-word which encodes a non-integral number of characters or octets is incorrectly formed.

A mail reader need not attempt to display the text associated with an encoded-word that is incorrectly formed. However, a mail reader **MUST NOT** prevent the display or handling of a message because an encoded-word is incorrectly formed.

## 7. Conformance

A mail composing program claiming compliance with this specification **MUST** ensure that any string of non-white-space printable ASCII characters within a "\*text" or "\*ctext" that begins with "=?" and ends with "?=" be a valid encoded-word. ("begins" means: at the start of the field-body or immediately following linear-white-space; "ends" means: at the end of the field-body or immediately preceding linear-white-space.) In addition, any "word" within a "phrase" that begins with "=?" and ends with "?=" must be a valid encoded-word.

A mail reading program claiming compliance with this specification must be able to distinguish encoded-words from "text", "ctext", or "word"s, according to the rules in section 6, anytime they appear in appropriate places in message headers. It must support both the "B" and "Q" encodings for any character set which it supports. The program must be able to display the unencoded text if the character

set is "US-ASCII". For the ISO-8859-\* character sets, the mail reading program must at least be able to display the characters which are also in the ASCII set.

## 8. Examples

```
From: =?US-ASCII?Q?Keith_Moore?= <moore@cs.utk.edu>
To: =?ISO-8859-1?Q?Keld_J=F8rn_Simonsen?= <keld@dkuug.dk>
CC: =?ISO-8859-1?Q?Andr=E9_?= Pirard <PIRARD@vml.ulg.ac.be>
Subject: =?ISO-8859-1?B?SWYgeW91IGNhbiByZWZkIHRoaxMgeW8=?=
        =?ISO-8859-2?B?dSB1bmRlcnN0YW5kIHRoZSBleGFtcGxlLg==?=?
```

```
From: =?ISO-8859-1?Q?Olle_J=E4rnefors?= <ojarnef@admin.kth.se>
To: ietf-822@dimacs.rutgers.edu, ojarnef@admin.kth.se
Subject: Time for ISO 10646?
```

```
To: Dave Crocker <dcrocker@mordor.stanford.edu>
Cc: ietf-822@dimacs.rutgers.edu, paf@comsol.se
From: =?ISO-8859-1?Q?Patrik_F=E4ltstr=F6m?= <paf@nada.kth.se>
Subject: Re: RFC-HDR care and feeding
```

```
From: Nathaniel Borenstein <nsb@thumper.bellcore.com>
      (=?iso-8859-8?b?7eXs+SDv4SDp7Oj08A==?=?=)
To: Greg Vaudreuil <gvaudre@NRI.Reston.VA.US>, Ned Freed
      <ned@innosoft.com>, Keith Moore <moore@cs.utk.edu>
Subject: Test of new header generator
MIME-Version: 1.0
Content-type: text/plain; charset=ISO-8859-1
```

## 9. References

- [1] Borenstein N., and N. Freed, "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies", RFC 1521, Bellcore, Innosoft, September 1993.
- [2] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, RFC 822, UDEL, August 1982.
- [3] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1340, USC/Information Sciences Institute, July 1992.

10. Security Considerations

Security issues are not discussed in this memo.

11. Author's Address

Keith Moore  
University of Tennessee  
107 Ayres Hall  
Knoxville TN 37996-1301

EMail: moore@cs.utk.edu