



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'Encode::MIME::Header.3perl'

\$ man Encode::MIME::Header.3perl

Encode::MIME::Header(3perl) Perl Programmers Reference Guide Encode::MIME::Header(3perl)

NAME

Encode::MIME::Header -- MIME encoding for an unstructured email header

SYNOPSIS

```
use Encode qw(encode decode);

my $mime_str = encode("MIME-Header", "Sample:Text \N{U+263A}");
# $mime_str is "=?UTF-8?B?U2FtcGxIOIRleHQg4pi6?="

my $mime_q_str = encode("MIME-Q", "Sample:Text \N{U+263A}");
# $mime_q_str is "=?UTF-8?Q?Sample=3AText_=E2=98=BA?="

my $str = decode("MIME-Header",
    "=?ISO-8859-1?B?SWYgeW91IGNhbiByZWFKIHRoaXMgeW8=?=\r\n " .
    "=?ISO-8859-2?B?dSB1bmRlcnN0YW5kIHRoZSBleGFtcGxILg==?="
);
# $str is "If you can read this you understand the example."

use Encode qw(decode :fallbacks);
use Encode::MIME::Header;
local $Encode::MIME::Header::STRICT_DECODE = 1;
```

```
my $strict_string = decode("MIME-Header", $mime_string, FB_CROAK);  
# use strict decoding and croak on errors
```

ABSTRACT

This module implements RFC 2047 <<https://tools.ietf.org/html/rfc2047>> MIME encoding for an unstructured field body of the email header. It can also be used for RFC 822 <<https://tools.ietf.org/html/rfc822>> 'text' token. However, it cannot be used directly for the whole header with the field name or for the structured header fields like From, To, Cc, Message-Id, etc... There are 3 encoding names supported by this module: "MIME-Header", "MIME-B" and "MIME-Q".

DESCRIPTION

Decode method takes an unstructured field body of the email header (or RFC 822 <<https://tools.ietf.org/html/rfc822>> 'text' token) as its input and decodes each MIME encoded-word from input string to a sequence of bytes according to RFC 2047 <<https://tools.ietf.org/html/rfc2047>> and RFC 2231 <<https://tools.ietf.org/html/rfc2231>>. Subsequently, each sequence of bytes with the corresponding MIME charset is decoded with the Encode module and finally, one output string is returned. Text parts of the input string which do not contain MIME encoded-word stay unmodified in the output string. Folded newlines between two consecutive MIME encoded-words are discarded, others are preserved in the output string. "MIME-B" can decode Base64 variant, "MIME-Q" can decode Quoted-Printable variant and "MIME-Header" can decode both of them. If Encode module does not support particular MIME charset or chosen variant then an action based on CHECK flags is performed (by default, the MIME encoded-word is not decoded).

Encode method takes a scalar string as its input and uses strict UTF-8 encoder for encoding it to UTF-8 bytes. Then a sequence of UTF-8 bytes is encoded into MIME encoded-words ("MIME-Header" and "MIME-B" use a Base64 variant while "MIME-Q" uses a Quoted-Printable variant) where each MIME encoded-word is limited to 75 characters. MIME encoded-words are separated by "CRLF SPACE" and joined to one output string. Output string is suitable for unstructured field body of the email header.

charset.

BUGS

Versions prior to 2.22 (part of Encode 2.83) have a malfunctioning decoder and encoder.

The MIME encoder infamously inserted additional spaces or discarded white spaces between consecutive MIME encoded-words, which led to invalid MIME headers produced by this module.

The MIME decoder had a tendency to discard white spaces, incorrectly interpret data or attempt to decode Base64 MIME encoded-words as Quoted-Printable. These problems were fixed in version 2.22. It is highly recommended not to use any version prior 2.22!

Versions prior to 2.24 (part of Encode 2.87) ignored CHECK flags. The MIME encoder used not strict utf8 encoder for input Unicode strings which could lead to invalid UTF-8 sequences. MIME decoder used also not strict utf8 decoder and additionally called the decode method with a "Encode::FB_PERLQQ" flag (thus user-specified CHECK flags were ignored). Moreover, it automatically croaked when a MIME encoded-word contained unknown encoding. Since version 2.24, this module uses strict UTF-8 encoder and decoder. And CHECK flags are correctly propagated.

Since version 2.22 (part of Encode 2.83), the MIME encoder should be fully compliant to RFC 2047 <<https://tools.ietf.org/html/rfc2047>> and RFC 2231 <<https://tools.ietf.org/html/rfc2231>>. Due to the aforementioned bugs in previous versions of the MIME encoder, there is a less strict compatible mode for the MIME decoder which is used by default. It should be able to decode MIME encoded-words encoded by pre 2.22 versions of this module. However, note that this is not correct according to RFC 2047 <<https://tools.ietf.org/html/rfc2047>>.

In default not strict mode the MIME decoder attempts to decode every substring which looks like a MIME encoded-word. Therefore, the MIME encoded-words do not need to be separated by white space. To enforce a correct strict mode, set variable `$Encode::MIME::Header::STRICT_DECODE` to 1 e.g. by localizing:

```
use Encode::MIME::Header;
```

```
local $Encode::MIME::Header::STRICT_DECODE = 1;
```

AUTHORS

Pali <pali@cpan.org>

SEE ALSO

Encode, RFC 822 <<https://tools.ietf.org/html/rfc822>>, RFC 2047

<<https://tools.ietf.org/html/rfc2047>>, RFC 2231 <<https://tools.ietf.org/html/rfc2231>>

perl v5.34.0

2023-11-23

Encode::MIME::Header(3perl)