

NAME

Encode::MIME::Header -- MIME 'B' and 'Q' header encoding

SYNOPSIS

```
use Encode qw/encode decode/;
$utf8    = decode('MIME-Header', $header);
$header  = encode('MIME-Header', $utf8);
```

ABSTRACT

This module implements RFC 2047 Mime Header Encoding. There are 3 variant encoding names; MIME-Header, MIME-B and MIME-Q. The difference is described below

| | decode () | encode () |
|-------------|------------------|------------------|
| MIME-Header | Both B and Q | =?UTF-8?B?....?= |
| MIME-B | B only; Q croaks | =?UTF-8?B?....?= |
| MIME-Q | Q only; B croaks | =?UTF-8?Q?....?= |

DESCRIPTION

When you `decode(=?encoding?X?ENCODED WORD?)`, *ENCODED WORD* is extracted and decoded for *X* encoding (B for Base64, Q for Quoted-Printable). Then the decoded chunk is fed to `decode(encoding)`. So long as *encoding* is supported by Encode, any source encoding is fine.

When you encode, it just encodes UTF-8 string with *X* encoding then quoted with `=?UTF-8?X?....?=`. The parts that RFC 2047 forbids to encode are left as is and long lines are folded within 76 bytes per line.

BUGS

It would be nice to support encoding to non-UTF8, such as `=?ISO-2022-JP?` and `=?ISO-8859-1?` but that makes the implementation too complicated. These days major mail agents all support `=?UTF-8?` so I think it is just good enough.

Due to popular demand, 'MIME-Header-ISO_2022_JP' was introduced by Makamaka. There are still too many MUAs especially cellular phone handsets which does not grok UTF-8.

SEE ALSO

Encode

RFC 2047, <http://www.faqs.org/rfcs/rfc2047.html> and many other locations.