

NAME

MIME::QuotedPrint - Encoding and decoding of quoted-printable strings

SYNOPSIS

```
use MIME::QuotedPrint;

$encoded = encode_qp($decoded);
$decoded = decode_qp($encoded);
```

DESCRIPTION

This module provides functions to encode and decode strings into and from the quoted-printable encoding specified in RFC 2045 - *MIME (Multipurpose Internet Mail Extensions)*. The quoted-printable encoding is intended to represent data that largely consists of bytes that correspond to printable characters in the ASCII character set. Each non-printable character (as defined by English Americans) is represented by a triplet consisting of the character "=" followed by two hexadecimal digits.

The following functions are provided:

```
encode_qp( $str )
encode_qp( $str, $eol )
encode_qp( $str, $eol, $binmode )
```

This function returns an encoded version of the string (`$str`) given as argument.

The second argument (`$eol`) is the line-ending sequence to use. It is optional and defaults to `"\n"`. Every occurrence of `"\n"` is replaced with this string, and it is also used for additional "soft line breaks" to ensure that no line end up longer than 76 characters. Pass it as `"\015\012"` to produce data suitable for external consumption. The string `"\r\n"` produces the same result on many platforms, but not all.

The third argument (`$binmode`) will select binary mode if passed as a TRUE value. In binary mode `"\n"` will be encoded in the same way as any other non-printable character. This ensures that a decoder will end up with exactly the same string whatever line ending sequence it uses. In general it is preferable to use the base64 encoding for binary data; see *MIME::Base64*.

An `$eol` of `""` (the empty string) is special. In this case, no "soft line breaks" are introduced and binary mode is effectively enabled so that any `"\n"` in the original data is encoded as well.

```
decode_qp( $str )
```

This function returns the plain text version of the string given as argument. The lines of the result are `"\n"` terminated, even if the `$str` argument contains `"\r\n"` terminated lines.

If you prefer not to import these routines into your namespace, you can call them as:

```
use MIME::QuotedPrint ();
$encoded = MIME::QuotedPrint::encode($decoded);
$decoded = MIME::QuotedPrint::decode($encoded);
```

Perl v5.8 and better allow extended Unicode characters in strings. Such strings cannot be encoded directly, as the quoted-printable encoding is only defined for single-byte characters. The solution is to use the Encode module to select the byte encoding you want. For example:

```
use MIME::QuotedPrint qw(encode_qp);
use Encode qw(encode);

$encoded = encode_qp(encode("UTF-8", "\x{FFFF}\n"));
print $encoded;
```

COPYRIGHT

Copyright 1995-1997,2002-2004 Gisle Aas.

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself.

SEE ALSO

MIME::Base64