

NAME

Encode::JP - Japanese Encodings

SYNOPSIS

```
use Encode qw/encode decode/;
$euc_jp = encode("euc-jp", $utf8);  # loads Encode::JP implicitly
$utf8 = decode("euc-jp", $euc_jp); # ditto
```

ABSTRACT

This module implements Japanese charset encodings. Encodings supported are as follows.

```
Canonical Alias Description
                           .____
        /\beuc.*jp$/i EUC (Extended Unix Character)
euc-jp
          /\bjp.*euc/i
      /\bujis$/i
shiftjis /\bshift.*jis$/i Shift JIS (aka MS Kanji)
       /\bsjis$/i
         /\bjis$/i 7bit JIS
7bit-jis
iso-2022-jp ISO-2022-JP
                                       [RFC1468]
            = 7bit JIS with all Halfwidth Kana
              converted to Fullwidth
iso-2022-jp-1 ISO-2022-JP-1
                                        [RFC2237]
                           = ISO-2022-JP with JIS X 0212-1990
              support. See below
                         Shift JIS + Apple vendor mappings
MacJapanese
cp932
      /\bwindows-31j$/i Code Page 932
                           = Shift JIS + MS/IBM vendor mappings
jis0201-raw
                           JIS0201, raw format
jis0208-raw
                          JIS0201, raw format
jis0212-raw
                          JIS0201, raw format
```

DESCRIPTION

To find out how to use this module in detail, see Encode.

Note on ISO-2022-JP(-1)?

ISO-2022-JP-1 (RFC2237) is a superset of ISO-2022-JP (RFC1468) which adds support for JIS X 0212-1990. That means you can use the same code to decode to utf8 but not vice versa.

```
$utf8 = decode('iso-2022-jp-1', $stream);
and
$utf8 = decode('iso-2022-jp', $stream);

yield the same result but
$with_0212 = encode('iso-2022-jp-1', $utf8);
is now different from
$without_0212 = encode('iso-2022-jp', $utf8);
```

In the latter case, characters that map to 0212 are first converted to U+3013 (0xA2AE in EUC-JP; a



white square also known as 'Tofu' or 'geta mark') then fed to the decoding engine. U+FFFD is not used, in order to preserve text layout as much as possible.

BUGS

The ASCII region (0x00-0x7f) is preserved for all encodings, even though this conflicts with mappings by the Unicode Consortium.

SEE ALSO

Encode