NAME

piconv -- iconv(1), reinvented in perl

SYNOPSIS

```
piconv [-f from_encoding] [-t to_encoding] [-s string] [files...]
piconv -l
piconv [-C N|-c|-p]
piconv -S scheme ...
piconv -r encoding
piconv -D ...
piconv -h
```

DESCRIPTION

piconv is perl version of **iconv**, a character encoding converter widely available for various Unixen today. This script was primarily a technology demonstrator for Perl 5.8.0, but you can use piconv in the place of iconv for virtually any case.

piconv converts the character encoding of either STDIN or files specified in the argument and prints out to STDOUT.

Here is the list of options. Each option can be in short format (-f) or long (--from).

-f,--from from_encoding

Specifies the encoding you are converting from. Unlike **iconv**, this option can be omitted. In such cases, the current locale is used.

-t,--to to_encoding

Specifies the encoding you are converting to. Unlike **iconv**, this option can be omitted. In such cases, the current locale is used.

Therefore, when both -f and -t are omitted, **piconv** just acts like **cat**.

-s,--string *string*

uses *string* instead of file for the source of text.

-l,--list

Lists all available encodings, one per line, in case-insensitive order. Note that only the canonical names are listed; many aliases exist. For example, the names are case-insensitive, and many standard and common aliases work, such as "latin1" for "ISO-8859-1", or "ibm850" instead of "cp850", or "winlatin1" for "cp1252". See *Encode::Supported* for a full discussion.

-C,--check *N*

Check the validity of the stream if *N* = 1. When *N* = -1, something interesting happens when it encounters an invalid character.

-c

Same as `-C 1`.

-p,--perlqq

--htmlcref

--xmlcref

Applies PERLQQ, HTMLCREF, XMLCREF, respectively. Try

```
piconv -f utf8 -t ascii --perlqq
```

To see what it does.

-h,--help

Show usage.

-D,--debug

Invokes debugging mode. Primarily for Encode hackers.

-S,--scheme scheme

Selects which scheme is to be used for conversion. Available schemes are as follows:

from_to

Uses Encode::from_to for conversion. This is the default.

decode_encode

Input strings are decode()d then encode()d. A straight two-step implementation.

perlio

The new perlIO layer is used. NI-S' favorite.

You should use this option if you are using UTF-16 and others which linefeed is not $/.

Like the *-D* option, this is also for Encode hackers.

SEE ALSO

iconv(1) locale(3) Encode Encode::Supported Encode::Alias PerlIO