

NAME

iconv - character set conversion

SYNOPSIS

```
iconv [OPTION...] [-f encoding] [-t encoding] [inputfile ...]
iconv -l
```

DESCRIPTION

The **iconv** program converts text from one encoding to another encoding. More precisely, it converts **from** the encoding given for the **-f** option **to** the encoding given for the **-t** option. Either of these encodings defaults to the encoding of the current locale. All the *inputfiles* are read and converted in turn; if no *inputfile* is given, the standard input is used. The converted text is printed to standard output.

The encodings permitted are system dependent. For the libiconv implementation, they are listed in the [iconv_open\(3\)](#) manual page.

Options controlling the input and output format:

-f *encoding*, **--from-code=***encoding*
 Specifies the encoding of the input.

-t *encoding*, **--to-code=***encoding*
 Specifies the encoding of the output.

Options controlling conversion problems:

-c When this option is given, characters that cannot be converted are silently discarded, instead of leading to a conversion error.

--unicode-subst=*formatstring*

When this option is given, Unicode characters that cannot be represented in the target encoding are replaced with a placeholder string that is constructed from the given *formatstring*, applied to the Unicode code point. The *formatstring* must be a format string in the same format as for the *printf* command or the *printf()* function, taking either no argument or exactly one unsigned integer argument.

--byte-subst=*formatstring*

When this option is given, bytes in the input that are not valid in the source encoding are replaced with a placeholder string that is constructed from the given *formatstring*, applied to the byte's value. The *formatstring* must be a format string in the same format as for the *printf* command or the *printf()* function, taking either no argument or exactly one unsigned integer argument.

--widechar-subst=*formatstring*

When this option is given, wide characters in the input that are not valid in the source encoding are replaced with a placeholder string that is constructed from the given *formatstring*, applied to the byte's value. The *formatstring* must be a format string in the same format as for the *printf* command or the *printf()* function, taking either no argument or exactly one unsigned integer argument.

Options controlling error output:

-s, --silent

When this option is given, error messages about invalid or unconvertible characters are omitted, but the actual converted text is unaffected.

The **iconv -lor icon v --list**command lists the names of the supported encodings, in a system dependent format. For the libiconv implementation, the names are printed in upper case, separated by whitespace, and alias names of an encoding are listed on the same line as the encoding itself.

EXAMPLES

iconv -f ISO-8859-1 -t UTF-8

converts input from the old West-European encoding ISO-8859-1 to Unicode.

iconv -f KOI8-R --byte-subst=<0x%x>

--unicode-subst=<U+%04X>

converts input from the old Russian encoding KOI8-R to the locale encoding, substituting an angle bracket notation with hexadecimal numbers for invalid bytes and for valid but unconvertible characters.

iconv --list

lists the supported encodings.

CONFORMING TO

POSIX:2001

SEE ALSO

[iconv_open\(3\)](#), [locale\(7\)](#)