

NAME

dos2unix – DOS/Mac to Unix and vice versa text file format converter

SYNOPSIS

```
dos2unix [options] [FILE ...] [-n INFILE OUTFILE ...]
unix2dos [options] [FILE ...] [-n INFILE OUTFILE ...]
```

DESCRIPTION

The Dos2unix package includes utilities `dos2unix` and `unix2dos` to convert plain text files in DOS or Mac format to Unix format and vice versa.

In DOS/Windows text files a line break, also known as newline, is a combination of two characters: a Carriage Return (CR) followed by a Line Feed (LF). In Unix text files a line break is a single character: the Line Feed (LF). In Mac text files, prior to Mac OS X, a line break was single Carriage Return (CR) character. Nowadays Mac OS uses Unix style (LF) line breaks.

Binary files are automatically skipped, unless conversion is forced.

Non-regular files, such as directories and FIFOs, are automatically skipped.

Symbolic links and their targets are by default kept untouched. Symbolic links can optionally be replaced, or the output can be written to the symbolic link target. Symbolic links on Windows are not supported. Windows symbolic links always replaced, keeping the targets unchanged.

Dos2unix was modelled after `dos2unix` under SunOS/Solaris and has similar conversion modes.

OPTIONS

-- Treat all following options as file names. Use this option if you want to convert files whose names start with a dash. For instance to convert a file named “-foo”, you can use this command:

```
dos2unix -- -foo
```

Or in new file mode:

```
dos2unix -n -- -foo out.txt
```

-ascii

Convert only line breaks. This is the default conversion mode.

-iso

Conversion between DOS and ISO-8859-1 character set. See also section CONVERSION MODES.

-1252

Use Windows code page 1252 (Western European).

-437

Use DOS code page 437 (US). This is the default code page used for ISO conversion.

-850

Use DOS code page 850 (Western European).

-860

Use DOS code page 860 (Portuguese).

-863

Use DOS code page 863 (French Canadian).

-865

Use DOS code page 865 (Nordic).

-7 Convert 8 bit characters to 7 bit space.

-c, --convmode CONVMODE

Set conversion mode. Where CONVMODE is one of: *ascii*, *7bit*, *iso*, *mac* with *ascii* being the default.

- f, --force**
Force conversion of binary files.
- h, --help**
Display help and exit.
- k, --keepdate**
Keep the date stamp of output file same as input file.
- L, --license**
Display program's license.
- l, --newline**
Add additional newline.

dos2unix: Only DOS line breaks are changed to two Unix line breaks. In Mac mode only Mac line breaks are changed to two Unix line breaks.

unix2dos: Only Unix line breaks are changed to two DOS line breaks. In Mac mode Unix line breaks are changed to two Mac line breaks.
- m, --add-bom**
Write an UTF-8 Byte Order Mark in the output file. Never use this option when the output encoding is other than UTF-8. See also section UNICODE.
- n, --newfile INFILE OUTFILE ...**
New file mode. Convert file INFILE and write output to file OUTFILE. File names must be given in pairs and wildcard names should *not* be used or you *will* lose your files.

The person who starts the conversion in new file (paired) mode will be the owner of the converted file. The read/write permissions of the new file will be the permissions of the original file minus the *umask*(1) of the person who runs the conversion.
- o, --oldfile FILE ...**
Old file mode. Convert file FILE and overwrite output to it. The program defaults to run in this mode. Wildcard names may be used.

In old file (in-place) mode the converted file gets the same owner, group, and read/write permissions as the original file. Also when the file is converted by an other user who has write permissions on the file (e.g. user root). The conversion will be aborted when it is not possible to preserve the original values. Change of owner could mean that the original owner is not able to read the file any more. Change of group could be a security risk, the file could be made readable for persons for whom it is not intended. Preservation of owner, group, and read/write permissions is only supported on Unix.
- q, --quiet**
Quiet mode. Suppress all warnings and messages. The return value is zero. Except when wrong command-line options are used.
- s, --safe**
Skip binary files (default).
- F, --follow-symlink**
Follow symbolic links and convert the targets.
- R, --replace-symlink**
Replace symbolic links with converted files (original target files remain unchanged).
- S, --skip-symlink**
Keep symbolic links and targets unchanged (default).
- V, --version**
Display version information and exit.

MAC MODE

In normal mode line breaks are converted from DOS to Unix and vice versa. Mac line breaks are not converted.

In Mac mode line breaks are converted from Mac to Unix and vice versa. DOS line breaks are not changed.

To run in Mac mode use the command-line option `-c mac` or use the commands `mac2unix` or `unix2mac`.

CONVERSION MODES

Conversion modes *ascii*, *7bit*, and *iso* are similar to those of `dos2unix/unix2dos` under SunOS/Solaris.

ascii

In mode `ascii` only line breaks are converted. This is the default conversion mode.

Although the name of this mode is ASCII, which is a 7 bit standard, the actual mode is 8 bit. Use always this mode when converting Unicode UTF-8 files.

7bit

In this mode all 8 bit non-ASCII characters (with values from 128 to 255) are converted to a 7 bit space.

iso Characters are converted between a DOS character set (code page) and ISO character set ISO-8859-1 (Latin-1) on Unix. DOS characters without ISO-8859-1 equivalent, for which conversion is not possible, are converted to a dot. The same counts for ISO-8859-1 characters without DOS counterpart.

When only option `-iso` is used `dos2unix` will try to determine the active code page. When this is not possible `dos2unix` will use default code page CP437, which is mainly used in the USA. To force a specific code page use options `-437` (US), `-850` (Western European), `-860` (Portuguese), `-863` (French Canadian), or `-865` (Nordic). Windows code page CP1252 (Western European) is also supported with option `-1252`. For other code pages use `dos2unix` in combination with `iconv(1)`. `Iconv` can convert between a long list of character encodings.

Never use ISO conversion on Unicode text files. It will corrupt UTF-8 encoded files.

Some examples:

Convert from DOS default code page to Unix Latin-1

```
dos2unix -iso -n in.txt out.txt
```

Convert from DOS CP850 to Unix Latin-1

```
dos2unix -850 -n in.txt out.txt
```

Convert from Windows CP1252 to Unix Latin-1

```
dos2unix -1252 -n in.txt out.txt
```

Convert from Windows CP1252 to Unix UTF-8 (Unicode)

```
iconv -f CP1252 -t UTF-8 in.txt | dos2unix > out.txt
```

Convert from Unix Latin-1 to DOS default code page.

```
unix2dos -iso -n in.txt out.txt
```

Convert from Unix Latin-1 to DOS CP850

```
unix2dos -850 -n in.txt out.txt
```

Convert from Unix Latin-1 to Windows CP1252

```
unix2dos -1252 -n in.txt out.txt
```

Convert from Unix UTF-8 (Unicode) to Windows CP1252

```
unix2dos < in.txt | iconv -f UTF-8 -t CP1252 > out.txt
```

See also <http://czyborra.com/charsets/codepages.html> and <http://czyborra.com/charsets/iso8859.html>.

UNICODE

Encodings

There exist different Unicode encodings. On Unix and Linux Unicode files are typically encoded in UTF-8 encoding. On Windows Unicode text files can be encoded in UTF-8, UTF-16, or UTF-16 big endian, but are mostly encoded in UTF-16 format.

Conversion

Unicode text files can have DOS, Unix or Mac line breaks, like regular text files.

All versions of dos2unix and unix2dos can convert UTF-8 encoded files, because UTF-8 was designed for backward compatibility with ASCII.

Dos2unix and unix2dos with Unicode UTF-16 support, can read little and big endian UTF-16 encoded text files. To see if dos2unix was built with UTF-16 support type `dos2unix -V`.

The Windows versions of dos2unix and unix2dos convert UTF-16 encoded files always to UTF-8 encoded files. Unix versions of dos2unix/unix2dos convert UTF-16 encoded files to the locale character encoding when it is set to UTF-8. Use the `locale (1)` command to find out what the locale character encoding is.

Because UTF-8 formatted text files are well supported on both Windows and Unix, dos2unix and unix2dos have no option to write UTF-16 files. All UTF-16 characters can be encoded in UTF-8. Conversion from UTF-16 to UTF-8 is without loss. UTF-16 files will be skipped on Unix when the locale character encoding is not UTF-8, to prevent accidental loss of text. When an UTF-16 to UTF-8 conversion error occurs, for instance when the UTF-16 input file contains an error, the file will be skipped.

ISO and 7-bit mode conversion do not work on UTF-16 files.

Byte Order Mark

On Windows Unicode text files typically have a Byte Order Mark (BOM), because many Windows programs (including Notepad) add BOMs by default. See also http://en.wikipedia.org/wiki/Byte_order_mark.

On Unix Unicode files typically don't have a BOM. It is assumed that text files are encoded in the locale character encoding.

Dos2unix can only detect if a file is in UTF-16 format if the file has a BOM. When an UTF-16 file doesn't have a BOM, dos2unix will see the file as a binary file.

Use dos2unix in combination with `iconv (1)` to convert an UTF-16 file without BOM.

Dos2unix never writes a BOM in the output file, unless you use option `-m`.

Unix2dos writes a BOM in the output file when the input file has a BOM, or when option `-m` is used.

Unicode examples

Convert from Windows UTF-16 (with BOM) to Unix UTF-8

```
dos2unix -n in.txt out.txt
```

Convert from Windows UTF-16 (without BOM) to Unix UTF-8

```
iconv -f UTF-16 -t UTF-8 in.txt | dos2unix > out.txt
```

Convert from Unix UTF-8 to Windows UTF-8 with BOM

```
unix2dos -m -n in.txt out.txt
```

Convert from Unix UTF-8 to Windows UTF-16

```
unix2dos < in.txt | iconv -f UTF-8 -t UTF-16 > out.txt
```

EXAMPLES

Read input from 'stdin' and write output to 'stdout'.

```
dos2unix
dos2unix -l -c mac
```

Convert and replace a.txt. Convert and replace b.txt.

```
dos2unix a.txt b.txt
dos2unix -o a.txt b.txt
```

Convert and replace a.txt in ascii conversion mode.

```
dos2unix a.txt
```

Convert and replace a.txt in ascii conversion mode. Convert and replace b.txt in 7bit conversion mode.

```
dos2unix a.txt -c 7bit b.txt
dos2unix -c ascii a.txt -c 7bit b.txt
dos2unix -ascii a.txt -7 b.txt
```

Convert a.txt from Mac to Unix format.

```
dos2unix -c mac a.txt
mac2unix a.txt
```

Convert a.txt from Unix to Mac format.

```
unix2dos -c mac a.txt
unix2mac a.txt
```

Convert and replace a.txt while keeping original date stamp.

```
dos2unix -k a.txt
dos2unix -k -o a.txt
```

Convert a.txt and write to e.txt.

```
dos2unix -n a.txt e.txt
```

Convert a.txt and write to e.txt, keep date stamp of e.txt same as a.txt.

```
dos2unix -k -n a.txt e.txt
```

Convert and replace a.txt. Convert b.txt and write to e.txt.

```
dos2unix a.txt -n b.txt e.txt
dos2unix -o a.txt -n b.txt e.txt
```

Convert c.txt and write to e.txt. Convert and replace a.txt. Convert and replace b.txt. Convert d.txt and write to f.txt.

```
dos2unix -n c.txt e.txt -o a.txt b.txt -n d.txt f.txt
```

RECURSIVE CONVERSION

Use dos2unix in combination with the *find*(1) and *xargs*(1) commands to recursively convert text files in a directory tree structure. For instance to convert all .txt files in the directory tree under the current directory type:

```
find . -name *.txt |xargs dos2unix
```

LOCALIZATION**LANG**

The primary language is selected with the environment variable LANG. The LANG variable consists out of several parts. The first part is in small letters the language code. The second is optional and is the country code in capital letters, preceded with an underscore. There is also an optional third part: character encoding, preceded with a dot. A few examples for POSIX standard type shells:

```

export LANG=nl                Dutch
export LANG=nl_NL             Dutch, The Netherlands
export LANG=nl_BE             Dutch, Belgium
export LANG=es_ES             Spanish, Spain
export LANG=es_MX             Spanish, Mexico
export LANG=en_US.iso88591    English, USA, Latin-1 encoding
export LANG=en_GB.UTF-8      English, UK, UTF-8 encoding

```

For a complete list of language and country codes see the gettext manual: <http://www.gnu.org/software/gettext/manual/gettext.html#Language-Codes>

On Unix systems you can use to command *locale* (1) to get locale specific information.

LANGUAGE

With the LANGUAGE environment variable you can specify a priority list of languages, separated by colons. Dos2unix gives preference to LANGUAGE over LANG. For instance, first Dutch and then German: LANGUAGE=nl:de. You have to first enable localization, by setting LANG (or LC_ALL) to a value other than "C", before you can use a language priority list through the LANGUAGE variable. See also the gettext manual: <http://www.gnu.org/software/gettext/manual/gettext.html#The-LANGUAGE-variable>

If you select a language which is not available you will get the standard English messages.

DOS2UNIX_LOCALEDIR

With the environment variable DOS2UNIX_LOCALEDIR the LOCALEDIR set during compilation can be overruled. LOCALEDIR is used to find the language files. The GNU default value is `/usr/local/share/locale`. Option `--version` will display the LOCALEDIR that is used.

Example (POSIX shell):

```
export DOS2UNIX_LOCALEDIR=$HOME/share/locale
```

RETURN VALUE

On success, zero is returned. When a system error occurs the last system error will be returned. For other errors 1 is returned.

The return value is always zero in quiet mode, except when wrong command-line options are used.

STANDARDS

http://en.wikipedia.org/wiki/Text_file
http://en.wikipedia.org/wiki/Carriage_return
<http://en.wikipedia.org/wiki/Newline>
<http://en.wikipedia.org/wiki/Unicode>

AUTHORS

Benjamin Lin - blin@socs.uts.edu.au Bernd Johannes Wuebben (mac2unix mode) - wuebben@kde.org, Christian Wurll (add extra newline) - wurll@ira.uka.de, Erwin Waterlander - waterlan@xs4all.nl (Maintainer)

Project page: <http://waterlan.home.xs4all.nl/dos2unix.html>

SourceForge page: <http://sourceforge.net/projects/dos2unix/>

Freecode: <http://freecode.com/projects/dos2unix>

SEE ALSO

file (1) *find* (1) *iconv* (1) *locale* (1) *xargs* (1)