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.

Besides line breaks Dos2unix can also convert the encoding of files. A few DOS code pages can be converted to Unix Latin−1. And Windows Unicode (UTF−16) files can be converted to Unix Unicode (UTF−8) files.

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. Writing to a symbolic link target is not supported on Windows.

Dos2unix was modelled after dos2unix under SunOS/Solaris. There is one important difference with the original SunOS/Solaris version. This version does by default in-place conversion (old file mode), while the original SunOS/Solaris version only supports paired conversion (new file mode). See also options −o and −n. Another difference is that the SunOS/Solaris version uses by default *iso* mode conversion while this version uses by default *ascii* mode conversion.

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
```

−−allow−chown

Allow file ownership change in old file mode.

When this option is used, the conversion will not be aborted when the user and/or group ownership of the original file can't be preserved in old file mode. Conversion will continue and the converted file will get the same new ownership as if it was converted in new file mode. See also options −o and −n. This option is only available if dos2unix has support for preserving the user and group ownership of files.

−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.

−b, −−keep−bom
> Keep Byte Order Mark (BOM). When the input file has a BOM, write a BOM in the output file. This is the default behavior when converting to DOS line breaks. See also option −r.

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

−D, −−display−enc ENCODING
> Set encoding of displayed text. Where ENCODING is one of: *ansi*, *unicode*, *unicodebom*, *utf8*, *utf8bom* with ansi being the default.
>
> This option is only available in dos2unix for Windows with Unicode file name support. This option has no effect on the actual file names read and written, only on how they are displayed.
>
> There are several methods for displaying text in a Windows console based on the encoding of the text. They all have their own advantages and disadvantages.
>
> **ansi**
> > Dos2unix's default method is to use ANSI encoded text. The advantage is that it is backwards compatible. It works with raster and TrueType fonts. In some regions you may need to change the active DOS OEM code page to the Windows system ANSI code page using the chcp command, because dos2unix uses the Windows system code page.
> >
> > The disadvantage of ansi is that international file names with characters not inside the system default code page are not displayed properly. You will see a question mark, or a wrong symbol instead. When you don't work with foreign file names this method is OK.
>
> **unicode, unicodebom**
> > The advantage of unicode (the Windows name for UTF−16) encoding is that text is usually properly displayed. There is no need to change the active code page. You may need to set the console's font to a TrueType font to have international characters displayed properly. When a character is not included in the TrueType font you usually see a small square, sometimes with a question mark in it.
> >
> > When you use the ConEmu console all text is displayed properly, because ConEmu automatically selects a good font.
> >
> > The disadvantage of unicode is that it is not compatible with ASCII. The output is not easy to handle when you redirect it to another program.
> >
> > When method unicodebom is used the Unicode text will be preceded with a BOM (Byte Order Mark). A BOM is required for correct redirection or piping in PowerShell.
>
> **utf8, utf8bom**
> > The advantage of utf8 is that it is compatible with ASCII. You need to set the console's font to a TrueType font. With a TrueType font the text is displayed similar as with the unicode encoding.
> >
> > The disadvantage is that when you use the default raster font all non-ASCII characters are displayed wrong. Not only unicode file names, but also translated messages become unreadable.

On Windows configured for an East-Asian region you may see a lot of flickering of the console when the messages are displayed.

In a ConEmu console the utf8 encoding method works well.

When method `utf8bom` is used the UTF–8 text will be preceded with a BOM (Byte Order Mark). A BOM is required for correct redirection or piping in PowerShell.

The default encoding can be changed with environment variable DOS2UNIX_DISPLAY_ENC by setting it to `unicode`, `unicodebom`, `utf8`, or `utf8bom`.

–f, −−force
Force conversion of binary files.

–gb, −−gb18030
On Windows UTF–16 files are by default converted to UTF–8, regardless of the locale setting. Use this option to convert UTF–16 files to GB18030. This option is only available on Windows. See also section GB18030.

–h, −−help
Display help and exit.

–i[FLAGS], −−info[=FLAGS] FILE ...
Display file information. No conversion is done.

The following information is printed, in this order: number of DOS line breaks, number of Unix line breaks, number of Mac line breaks, byte order mark, text or binary, file name.

Example output:

```
     6         0         0   no_bom    text    dos.txt
     0         6         0   no_bom    text    unix.txt
     0         0         6   no_bom    text    mac.txt
     6         6         6   no_bom    text    mixed.txt
    50         0         0   UTF-16LE  text    utf16le.txt
     0        50         0   no_bom    text    utf8unix.txt
    50         0         0   UTF-8     text    utf8dos.txt
     2       418       219   no_bom    binary  dos2unix.exe
```

Note that sometimes a binary file can be mistaken for a text file. See also option −s.

Optionally extra flags can be set to change the output. One or more flags can be added.

0   Print the file information lines followed by a null character instead of a newline character. This enables correct interpretation of file names with spaces or quotes when flag c is used. Use this flag in combination with **xargs**(1) option −0 or −−null.

d   Print number of DOS line breaks.

u   Print number of Unix line breaks.

m   Print number of Mac line breaks.

b   Print the byte order mark.

t   Print if file is text or binary.

c   Print only the files that would be converted.

With the `c` flag dos2unix will print only the files that contain DOS line breaks, unix2dos will print only file names that have Unix line breaks.

h   Print a header.

p   Show file names without path.

Examples:

Show information for all *.txt files:

```
dos2unix −i *.txt
```

Show only the number of DOS line breaks and Unix line breaks:

```
dos2unix −idu *.txt
```

Show only the byte order mark:

```
dos2unix −−info=b *.txt
```

List the files that have DOS line breaks:

```
dos2unix −ic *.txt
```

List the files that have Unix line breaks:

```
unix2dos −ic *.txt
```

Convert only files that have DOS line breaks and leave the other files untouched:

```
dos2unix −ic0 *.txt | xargs −0 dos2unix
```

Find text files that have DOS line breaks:

```
find −name '*.txt' −print0 | xargs −0 dos2unix −ic
```

−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 a Byte Order Mark (BOM) in the output file. By default an UTF−8 BOM is written.

When the input file is UTF−16, and the option −u is used, an UTF−16 BOM will be written.

Never use this option when the output encoding is other than UTF−8, UTF−16, or GB18030. 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.

−−no−allow−chown
Don't allow file ownership change in old file mode (default).

Abort conversion when the user and/or group ownership of the original file can't be preserved in old file mode. See also options −o and −n. This option is only available if dos2unix has support for preserving the user and group ownership of files.

−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 another 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.
>
> To check if dos2unix has support for preserving the user and group ownership of files type `dos2unix -V`.
>
> Conversion is always done via a temporary file. When an error occurs halfway the conversion, the temporary file is deleted and the original file stays intact. When the conversion is successful, the original file is replaced with the temporary file. You may have write permission on the original file, but no permission to put the same user and/or group ownership properties on the temporary file as the original file has. This means you are not able to preserve the user and/or group ownership of the original file. In this case you can use option `--allow-chown` to continue with the conversion:
>
>   dos2unix --allow-chown foo.txt
>
> Another option is to use new file mode:
>
>   dos2unix -n foo.txt foo.txt
>
> The advantage of the `--allow-chown` option is that you can use wildcards, and the ownership properties will be preserved when possible.

−q, −−quiet
> Quiet mode. Suppress all warnings and messages. The return value is zero. Except when wrong command-line options are used.

−r, −−remove−bom
> Remove Byte Order Mark (BOM). Do not write a BOM in the output file. This is the default behavior when converting to Unix line breaks. See also option −b.

−s, −−safe
> Skip binary files (default).
>
> The skipping of binary files is done to avoid accidental mistakes. Be aware that the detection of binary files is not 100% foolproof. Input files are scanned for binary symbols which are typically not found in text files. It is possible that a binary file contains only normal text characters. Such a binary file will mistakenly be seen as a text file.

−u, −−keep−utf16
> Keep the original UTF−16 encoding of the input file. The output file will be written in the same UTF−16 encoding, little or big endian, as the input file. This prevents transformation to UTF−8. An UTF−16 BOM will be written accordingly. This option can be disabled with the −ascii option.

−ul, −−assume−utf16le
> Assume that the input file format is UTF−16LE.
>
> When there is a Byte Order Mark in the input file the BOM has priority over this option.
>
> When you made a wrong assumption (the input file was not in UTF−16LE format) and the conversion succeeded, you will get an UTF−8 output file with wrong text. You can undo the wrong conversion with **iconv** (1) by converting the UTF−8 output file back to UTF−16LE. This will bring back the original file.
>
> The assumption of UTF−16LE works as a *conversion mode*. By switching to the default *ascii* mode the UTF−16LE assumption is turned off.

−ub, −−assume−utf16be
> Assume that the input file format is UTF−16BE.
>
> This option works the same as option −ul.

−v, −−verbose
> Display verbose messages. Extra information is displayed about Byte Order Marks and the amount of converted line breaks.

−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

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.

On Unix/Linux UTF−16 encoded files are converted to the locale character encoding. Use the **locale** (1) command to find out what the locale character encoding is. When conversion is not possible a conversion error will occur and the file will be skipped.

On Windows UTF−16 files are by default converted to UTF−8. UTF−8 formatted text files are well supported on both Windows and Unix/Linux.

UTF−16 and UTF−8 encoding are fully compatible, there will no text be lost in the conversion. 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.

When option −u is used, the output file will be written in the same UTF−16 encoding as the input file. Option −u prevents conversion to UTF−8.

Dos2unix and unix2dos have no option to convert UTF−8 files to UTF−16.

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 option `-ul` or `-ub` to convert an UTF–16 file without BOM.

Dos2unix writes by default no BOM in the output file. With option `-b` Dos2unix writes a BOM when the input file has a BOM.

Unix2dos writes by default a BOM in the output file when the input file has a BOM. Use option `-r` to remove the BOM.

Dos2unix and unix2dos write always a BOM when option `-m` is used.

Unicode file names on Windows

Dos2unix has optional support for reading and writing Unicode file names in the Windows Command Prompt. That means that dos2unix can open files that have characters in the name that are not part of the default system ANSI code page. To see if dos2unix for Windows was built with Unicode file name support type `dos2unix -V`.

There are some issues with displaying Unicode file names in a Windows console. See option `-D`, `--display-enc`. The file names may be displayed wrongly in the console, but the files will be written with the correct name.

Unicode examples

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

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

Convert from Windows UTF–16LE (without BOM) to Unix UTF–8:

```
dos2unix -ul -n in.txt 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
```

GB18030

GB18030 is a Chinese government standard. A mandatory subset of the GB18030 standard is officially required for all software products sold in China. See also <http://en.wikipedia.org/wiki/GB_18030>.

GB18030 is fully compatible with Unicode, and can be considered an unicode transformation format. Like UTF–8, GB18030 is compatible with ASCII. GB18030 is also compatible with Windows code page 936, also known as GBK.

On Unix/Linux UTF–16 files are converted to GB18030 when the locale encoding is set to GB18030. Note that this will only work if the locale is supported by the system. Use command `locale -a` to get the list of supported locales.

On Windows you need to use option `-gb` to convert UTF–16 files to GB18030.

GB18030 encoded files can have a Byte Order Mark, like Unicode files.

EXAMPLES

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

```
dos2unix < a.txt
cat a.txt | dos2unix
```

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

In a Unix shell the **find**(1) and **xargs**(1) commands can be used to run dos2unix recursively over all text files in a directory tree. For instance to convert all .txt files in the directory tree under the current directory type:

```
find . −name '*.txt' −print0 |xargs −0 dos2unix
```

The **find**(1) option −print0 and corresponding **xargs**(1) option −0 are needed when there are files with spaces or quotes in the name. Otherwise these options can be omitted. Another option is to use **find**(1) with the −exec option:

```
find . −name '*.txt' −exec dos2unix {} \;
```

In a Windows Command Prompt the following command can be used:

```
for /R %G in (*.txt) do dos2unix "%G"
```

PowerShell users can use the following command in Windows PowerShell:

```
get−childitem −path . −filter '*.txt' −recurse | foreach−object {dos2unix $_.Ful
```

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/html_node/Usual−Language−Codes.html>

On Unix systems you can use the 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/html_node/The−LANGUAGE−variable.html>

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/>

SEE ALSO

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