TFOR version 1.0 beta 1
by KloroX, August 1999

About text files

If you are familiar with text files and their format, you may skip this section.

Plain text files do not contain information about fonts (Helvetica, Arial etc.), font styles (italics, bold etc.) tab positions, or any other attribute that modifies the way text looks on the screen or on a printout. This has two consequences. The first is that the program you use to read, edit or print the text file must decide how to display it. The second is that plain text files are the closest thing to a file format that is portable among all computer platforms. However, “closest” is sometimes not good enough. If you use a PC, for instance, you may have tried to read a text file that was written on a UNIX system. Depending on the program you used, you may have succeeded in displaying a pretty normal-looking text, or a hopeless garble devoid of any line- and paragraph-ends (Windows Notepad is a notorious offender in this respect). The situation is even worse with non-English alphabets (the Unicode system is supposed to solve these problems, but several countries did not wait for it and developed their own, mutually incompatible encoding schemes).

Virtually all English-language plain text files use the ASCII code system, which assigns a value between 0 and 127 to each character. This allows for uppercase and lowercase letters, numbers, punctuation symbols, and a few special-purpose characters (see below). Each character is stored in one byte, which allows for values between 0 and 255. Thus ASCII leaves 128 values unused. These are used by the extended ASCII character set, and contain a jumble of “foreign” characters (accented vowels, etc.) and common symbols. Windows follows this convention, while DOS used the range 128-255 for different characters, including a set of proprietary characters for drawing the borders of boxes on the screen.

In the context of text files, the most important special characters in the ASCII set are the space or blank (ASCII 32), tab (ASCII 9), carriage return (ASCII 13) and line feed (ASCII 10). The space needs no explanation. The tab is supposed to move the position of text to the right by a number of columns, and is used to align text columns vertically. If you are using a mono-spaced font (in which all characters are the same width) one tab usually corresponds to 8 characters. Inserting a tab in a line of text moves all characters following the tab to the right, so that they start on a column that is distant from the beginning of the line by a multiple of 8 character-widths. If you are using a proportional font, in which the character width varies with each letter, matching the width of a tab exactly with a number of spaces is often impossible.

The carriage return and line feed are important when a line of text must be ended and text must continue on the following line. There are at least three standards for the end-of-line marker (which can be a single character or a pair of characters). These standards are discussed below. The different end-of-line standards are the main reason why your text file may look garbled with certain programs or on certain systems.

What is TFOR?

TFOR is a **T**ext-file **FOR**matter. It reads a text file, analyses it, displays some of its characteristics, and then allows you to do some re-formatting and to save a new version of the file. It is **not** an editor, and it does not allow you to change the contents of the file in any way, other than its formatting. Also, it applies your formatting rules to the **entire file**, so you cannot use it to reformat just one paragraph or one section of the text. The formatting controlled by TFOR is restricted to changing the end-of-line marker, introducing line-breaks at regular intervals in a file devoid of them, adding a blank line after the end of each paragraph (under certain conditions), and a few related operations.

TFOR is freeware. There are no licenses or fees of any kind, and you are free to distribute it in any way you want, with the limitation that you must distribute the file TFOR10.ZIP as you received it. In other words, you cannot distribute parts of the TFOR files, you must distribute all of them. The files in the original package are:

TFOR.EXE
TFOR.PDF (the file you are reading)
TFOR.TXT (a text-only version of TFOR.DOC)

There are no guarantees of any kind, and you get just what you pay for. The author believes the program to work as described in the present file, but does not make any guarantee to this effect. Likewise, there is absolutely no guarantee that the possession or use of this program will not result in loss of information, time, temper, money, property, health, life or afterlife. *Caveat emptor*.

No support for TFOR is available from the author of this program. The author believes that this document, together with some experimentation, will be sufficient for all users. Any user who cannot understand the instructions in this document probably needs to gain more familiarity with the Windows operating system and/or with the use of text files, before attempting to use TFOR.

The creators of some acronyms, like SCSI and TEX, want you to pronounce them in a certain way (often a most unnatural way; for instance, they want you to pronounce TEX like *tesh*). You are entirely free, however, to pronounce TFOR in any way you like, or not to pronounce it at all. I just don't care.

How do I install it?

The following instructions assume that you have some familiarity with the use of a PC and Microsoft Windows 95, 98, NT or 2000.

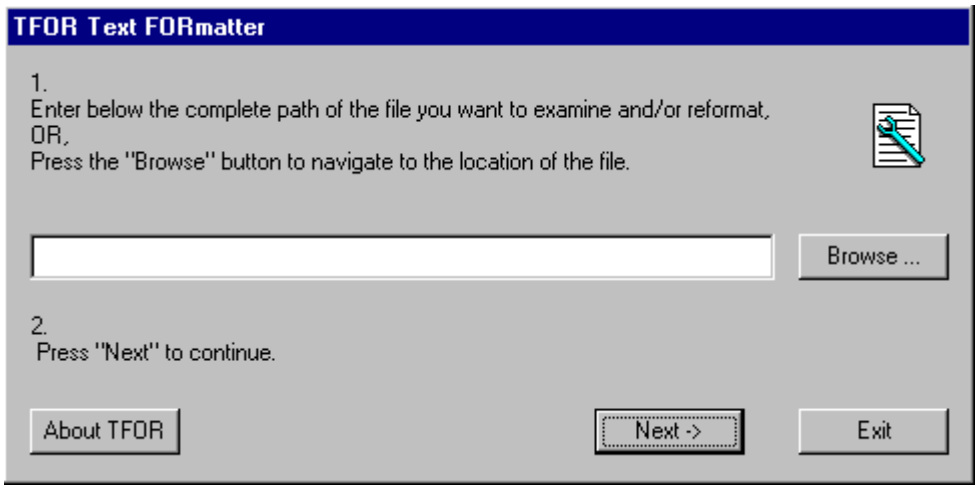
Since you are reading this file, probably you have already unzipped the archive TFOR10.ZIP. If not, do it now. Then copy the file TFOR.EXE to your hard disk. If you wish, you can create a shortcut to TFOR.EXE on your desktop or on the Windows Start menu. No other files are necessary.

How do I start it?

Just run TFOR.EXE.

How do I use it?

When you start TFOR.EXE, the following dialog box appears:

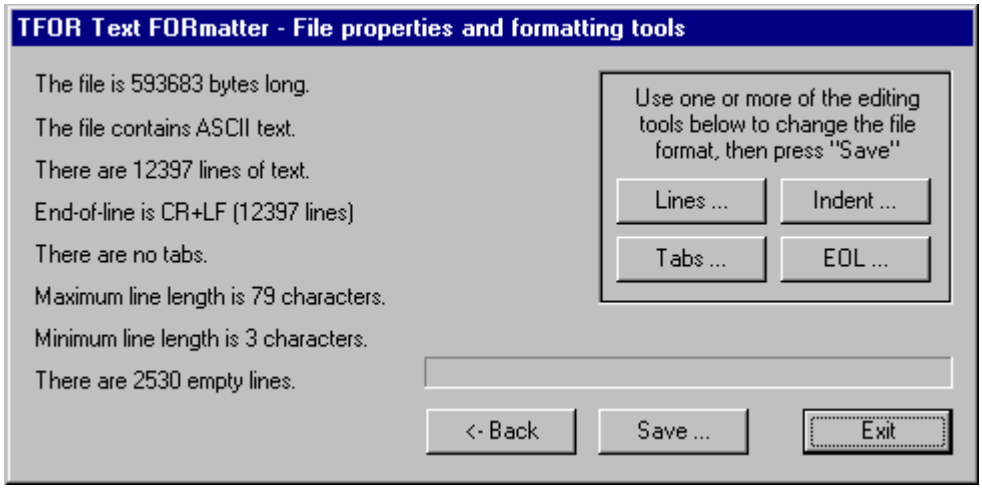


At this point follow the instructions:

1. Enter the complete path of the file you want to examine and/or reformat. You can do this either by writing the path in the edit control (for example: *C:\My Documents\e-books\Tarzan.txt*), or by pressing the "Browse" button. The latter will display a dialog box that allows you to navigate to the location of the file. This dialog is fairly standard, so, as a Windows user, you should have no problem with it.
2. Press the “Next” button.

TFOR will read the file into memory. There are no built-in limitations to the size of text files TFOR can manage. However, it keeps the entire text in memory, so you may be unable to read an unusually large text file (e.g., 5 Megabyte) on a system equipped with little RAM memory.

Subsequently, TFOR analyses the text (large files and/or slow computers may require some patience) and displays the following dialog, which shows a list of characteristics of the text file:



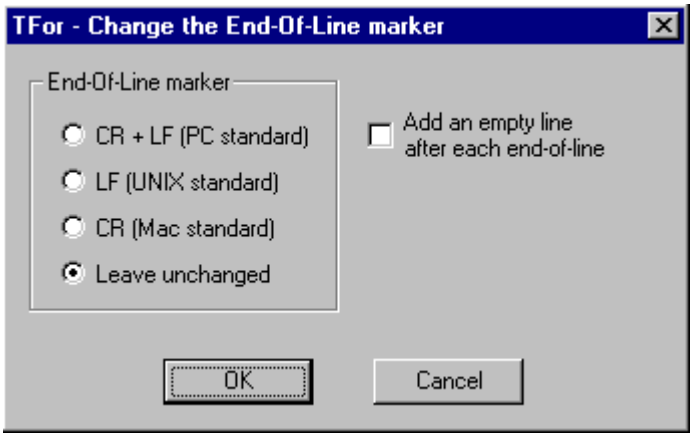
The file characteristics displayed by TFOR are:

1. The length of the file contents in bytes. This can be somewhat less than the space occupied by the file on your hard disk, because Windows allocates file space on a hard disk in chunks of fixed size.
2. Whether the file contains only plain text (i.e., ASCII characters from 32 to 127 and tabs, which is the character set used in English), extended ASCII (i.e., the ASCII characters between 128 and 255, which is used for “foreign” characters and symbols), or non-text characters (i.e, those ASCII characters between 0 and 31, which – with the exception of tabs – do not normally occur in English text files). Knowing that a file contains some “strange” characters may alert you that its contents may not be just plain text.
3. The number of lines of text in the files. This is calculated on the basis of the number of end-of-line markers (see below) found in the file. Note that some text files use just one end-of-line marker at the end of each paragraph, without using hard line-breaks within each paragraph (i.e., without having a fixed line length). In this case, each paragraph is counted as a single line. In other cases, hard line-breaks are used, and each line is counted. Empty lines (i.e., lines containing no characters) are not counted here. However, a line containing only spaces or tabs is not recognised as empty.
4. The type of the end-of-line marker(s) used in the file. TFOR can recognise all three common types: UNIX uses LF (Line Feed, i.e. ASCII 10). Macs use CR (Carriage Return, i.e. ASCII 13). DOS and Windows use a CR followed immediately by an LF. TFOR lists the type(s) of end-of-marker found in the file, and, between parentheses, the number of occurrences of each type.
5. The number of tabs (ASCII 9) in the file. TFOR recognises only explicit tabs (ASCII 7 characters), not sequences of spaces used to simulate tabs.
6. The length in characters of the longest line of text in the file. The length of a line is computed as the number of characters between successive end-of-line markers.

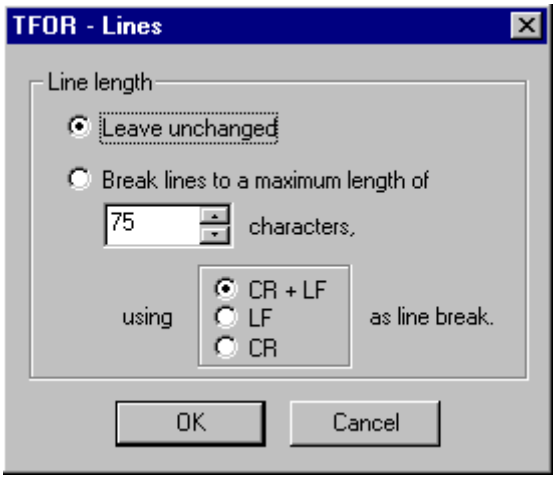
7. The length in characters of the shortest line of text in the file. Empty lines are not listed here, only lines that contain at least one character (including spaces and tabs).
8. The number of empty text lines in the file. An empty text line is detected from the presence of two end-of-line markers in sequence, not separated by any intervening characters.

The above information can tell you a lot about the formatting of a text file. For instance, very long lines indicate that probably the file contains no hard line-breaks, and that the few end-of-line markers are placed only at the end of paragraphs. A maximum line length of 70-80 characters, on the other hand, indicates that hard spaces are used. A rather large number of empty lines may indicate (depending on the type of contents) that each paragraph is followed by an empty line. You will need to confirm these characteristics by looking at the file contents with your favourite editor or reader.

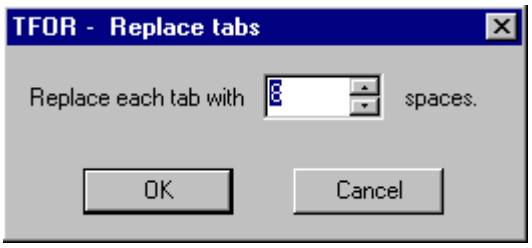
Once you have decided how to alter the text formatting, choose any of the available tools:



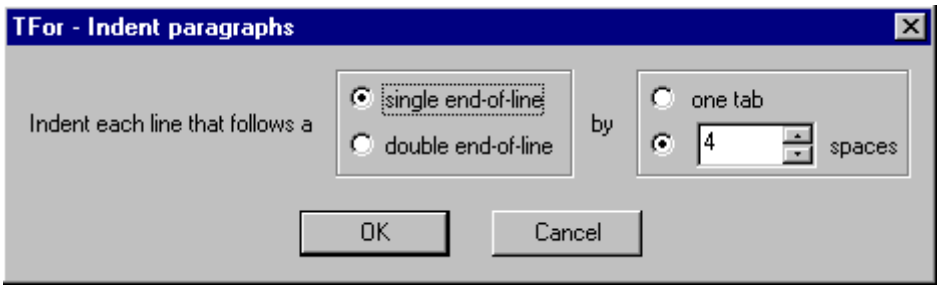
EOL. Here you can choose to change all end-of-line markers to the same standard. TFOR allows you to choose CR, LF or CR+LF (the latter is the PC standard). You have also the option of doubling all existing end-of-line markers by adding an empty line after each of them. When applied to a text that has end-of-line markers only at the end of each paragraph, this option yields a text with an empty line at the end of each paragraph (which is a common standard for e-texts). Remember to perform this operation **before** breaking the text into lines of maximum length (see below), in order to add an empty line only after the end of each paragraph. If the text is already broken into lines, the effect of this operation will be of adding an empty line after **each line** of text.



Lines. You can choose to break the text into lines of a given maximum length. TFOR will break lines between words so that each line fits within the maximum length. If the line contains no spaces, it is broken arbitrarily at the maximum length.



Tabs. This allows you to convert tabs to multiple spaces, and to specify the number of spaces equivalent to one tab. This tool produces predictable results only when the text is viewed with a mono-spaced font.



Indent. This tool adds the specified number of spaces (or a single tab character) at the beginning of each paragraph (i.e., after each end-of-line marker). You can choose to indent the text following a single end-of-line (if the file contains end-of-line markers only at the end of each paragraph) or after a double end-of-line (if the file contains a single end-of-line at the end of each line, and a double one at the end of each paragraph.

Note that the final result often depends on the order in which you apply the tools.

After applying a tool, TFOR analyses the changed text, and displays updated information.

When you are finished with the tools, press “Next” to save the file. TFOR questions you very sternly before allowing you to overwrite an existing file. In practice, it is always best to save to a new file instead (my suggestion is appending 1, 2, 3 etc. at the end of the original file name, so you will be able to recognise successive versions), and/or save in another directory.

After saving a file, you can perform more formatting operations, then save another version of the file.

Revision history

August 6, 1999
Version 0.0 Alpha release.

August 12, 1999
Version 1.0 Beta 1 release.

- Added feature: Path of last opened file is remembered between sessions.
- Corrected bug: Lines were not broken correctly between words.
- Corrected bug: Empty lines were indented together with non-empty ones.
- Corrected bug: Number of empty lines was not displayed correctly.
- Corrected a few small bugs in the routines for formatting end-of-line markers. Operation is slightly speeded up, but the program uses more memory.
- Split the tools among four separate dialog boxes.