

Java Application: TrimFile

Written by: Keith Fenske, <http://kwfenske.github.io/>

First version: Friday, 18 December 2009

Document revised: Thursday, 11 March 2010

Copyright © 2009 by Keith Fenske. Apache License or GNU General Public License.

Description

TrimFile is a Java 1.4 console application to remove trailing white space (blanks or tabs) from the end of each line in a plain text file. Extra spaces commonly accumulate while editing source programs in a graphical compiler (IDE). They aren't a problem, but they do waste file space and occasionally affect the appearance of programs.

Input and output files are read and written as sequences of characters. You may specify the character set encoding. The local system's default encoding will be assumed if you don't choose an encoding. The input file name is usually the first parameter (argument) on the command line. An output file name is usually the second parameter. If no output file is given, then output will be written on standard output (the console), which may be redirected with the ">" operator. If no input file is given, and no option to read from standard input, then an error message is printed. A typical command line would be:

```
java TrimFile3 -local filename.txt newfilename.txt
```

Options may be given on the command line. They should appear before the file names, but this is not strictly enforced. Most options are related to the characters (bytes) used to separate lines in the text file. Linux/UNIX and most newer systems want a single "newline" character (NL or 0x0A); older Macintosh OS 9 applications used a single "carriage return" (CR or 0x0D); many DOS and Windows applications still use CR immediately followed by "line feed" (LF, which is also 0x0A). All of these separators are recognized on input; you may select the output separator. The options are:

-clean

do not copy unrecognized control codes to the output file. Extra control codes are normally passed through as text characters (unchanged).

-code=name

specifies both **-incode** and **-outcode**. The default is the local encoding. You may need to quote this according to your system's command syntax. Please use only canonical Java character set names, with the exact spelling as found on the following web page:

<http://java.sun.com/j2se/1.5.0/docs/guide/intl/encoding.doc.html>

-copy

copy the input text without trimming. Use this option to change character sets or line separators without removing trailing white space.

-cr (or) -mac

separate output lines with CR characters for Macintosh OS 9 (0x0D).

-crlf (or) -dos

separate output lines with CR/LF pairs for DOS/Windows (0x0D/0x0A).

-help (or) -?

show a summary of the command-line options and syntax.

-incode=name

specifies the input character set. This option is only necessary when the input and output have different character sets, and does not apply when reading from standard input. See the -code option first.

-input=name

specifies the input file name, when the name looks like an option and can't be given as a parameter on the command line. You may need to quote this according to your system's command syntax.

-local (or) -default

use the local system's default line separator on output.

-nl (or) -lf (or) -unix

separate output lines with UNIX newline characters (0x0A).

-outcode=name

specifies the output character set. This option is only necessary when the input and output have different character sets, and does not apply when writing to standard output. See the -code option first. No error is generated if an output encoding can not represent input characters; a replacement character will be arbitrarily chosen.

-output=name

specifies the output file name, when the name looks like an option and can't be given as a parameter on the command line. You may need to quote this according to your system's command syntax.

-same (or) -asis

use the same line separators on output as from the input (default).

-stdin

read input from standard input (pipe) instead of a file. You must specify this option if you aren't using a file.

-stdout

write output on standard output (pipe) instead of a file. This is the default action if an output file name is not given.

You may use the null device for output if you only want to check that a text file has no trailing space (see /dev/null on Linux/UNIX or NUL: on DOS/Windows). The console application will return an exit status equal to the number of white space characters removed (zero or more), or -1 for errors. There is no graphical interface (GUI) for this program; it must be run from a command prompt, command shell, or terminal window.

Apache License or GNU General Public License

TrimFile is free software and has been released under the terms and conditions of the Apache License (version 2.0 or later) and/or the GNU General Public License (GPL, version 2 or later). This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY, without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the license(s) for more details. You should have received a copy of the licenses along with this program. If not, see the <http://www.apache.org/licenses/> and <http://www.gnu.org/licenses/> web pages.

Installation

You must have the Java run-time environment (JRE) installed on your computer. TrimFile was developed with Java 1.4 and should run on later versions. It may also run on earlier versions, but this has not been tested. You can download the JRE from Oracle (formerly Sun Microsystems):

JRE for end users: <http://www.java.com/getjava/>

SDK for programmers: <http://www.oracle.com/technetwork/java/>

IDE for programmers: <http://www.netbeans.org/>

Once Java is installed, you need to put the program files for TrimFile into a folder (directory) on your hard drive. The name of the folder and the location are your choice, except it is easier if the name does not include spaces. Assume that files will go into a "C:\Java" folder. Then create the folder and unpack the Java *.class files into this folder (if you received the program as a ZIP file). The files look something like this:

ApacheLicense20.txt (12 KB, legal notice)
GnuPublicLicense3.txt (35 KB, legal notice)
RunJavaPrograms.pdf (60 KB, more notes about running Java)
ShowCharSets2.java (3 KB, program to generate Java character set names)
ShowCharSets2.txt (19 KB, sample file with Java character set names)
TrimFile3.class (8 KB, executable program)
TrimFile3.doc (35 KB, this documentation in Microsoft Word format)
TrimFile3.jar (5 KB, archive file with same class file inside)
TrimFile3.java (32 KB, source code)
TrimFile3.manifest (1 KB, main class manifest for archive file)
TrimFile3.pdf (75 KB, this documentation in Adobe Acrobat format)

To run the program on Windows, start a DOS command prompt, which is Start button, Programs, Accessories, Command Prompt on Windows XP/Vista/7. Change to the folder with the program files and run the program with a “java” command:

```
c:  
cd \java  
java TrimFile3
```

The program name “TrimFile3” must appear exactly as shown; uppercase and lowercase letters are different in Java names. One complication may arise when trying to run this program. Java looks for an environment variable called CLASSPATH. If it finds this variable, then that is a list of folders where it looks for *.class files. It won’t look anywhere else, not even in the current directory, unless the path contains “.” as one of the choices. The symptom is an error message that says:

```
Exception in thread "main" java.lang.NoClassDefFoundError: TrimFile3
```

To find out if your system has a CLASSPATH variable defined, type the following command in a DOS window:

```
set CLASSPATH
```

To temporarily change the CLASSPATH variable to the current directory, use the following command line:

```
java -cp . TrimFile3
```

To permanently change the CLASSPATH, you must find where it is being set. This is in Control Panel, System, Advanced, Environment Variables on Windows XP/Vista/7.

Removal or Uninstall

To remove this program from your computer, delete the installation files listed above. If the folder that contained the files is now empty, you may also delete the folder ... if you created the folder, of course, not the system. If you created desktop shortcuts or Start menu items, then delete those too. There are no hidden configuration or preference files, and no information is stored in the Windows system registry. You don't need an "uninstall" program.

Restrictions and Limitations

TrimFile does not add, remove, or otherwise detect the Unicode "byte order mark" (BOM, U+FEFF) found at the beginning of some files. The BOM if present will be treated as printable text. Output character sets may not recognize a BOM, and some such as UTF-16 will add the BOM if missing. Consider using UTF-16BE or UTF-16LE to avoid an unwanted BOM.

file: TrimFile3.doc 2024-12-12