

FIXEDIT ver 3.3

Datafile Text Editor and Formatter

User Guide

copyright askey127
June 2015

User's Guide to Fixedit

Fixedit is freeware and is copyrighted but has no distribution or usage restrictions.

Fixedit is a unique editor. While providing conventional editing capability for plain text, it has unusual ability to make global changes to the loaded text, and perform changes that would take long sessions of manual editing using a conventional program.

The capabilities are similar to some of the possibilities using complex Perl macros, or the Linux origin stream editor *sed*, either of which would require a lot more work..

The way Fixedit usually works is to use **File, Open** to import a text file. You can use Windows Paste as well. Once that original file is imported, the original file is closed and left untouched by the program, unless you overwrite it on purpose. The Global tab is then used to make major changes in the contents of the text within Fixedit. The resulting altered body of text is then saved as another filename using **File, SaveAs**.

A few examples of the simpler capabilities (each of these can be done quickly) :

o Load a text file containing 1000 lines. Remove all the duplicate lines. Change the whole file to upper case. Save as another filename.

*o Load a text file containing 500 lines. Delete all the lines containing the string **C:_OTL** and save the remaining lines to a different file.*

o Load a text file containing 500 lines. Add an asterisk as the first character in every line that contains the word "infected". Sort the lines in the file according to the first character in each line, so all the "infected" lines are at the top. Save results a different file.

o Make a composite HOSTS file by copy/paste of files from multiple sources(>100k lines). Remove duplicates.

At any stage of alteration, clicking on the "Show/Edit Current Text" tab will let you see the condition of the altered text. Every time the **OK** button is clicked on "Make Global Changes", the text on the first page is rewritten to show the programmed changes. Fixedit always changes any tab characters to spaces (modulo 8).

If a major error is made, and the altered text is not useful, you can just Exit and nothing will be saved. The original file can be imported again for another try if desired.

This Guide could not possibly illustrate all the combinations of line tests and changes that are feasible with Fixedit. A complex example is given at the end to illustrate a few of the possibilities. This Guide will focus on explaining how each selection works.

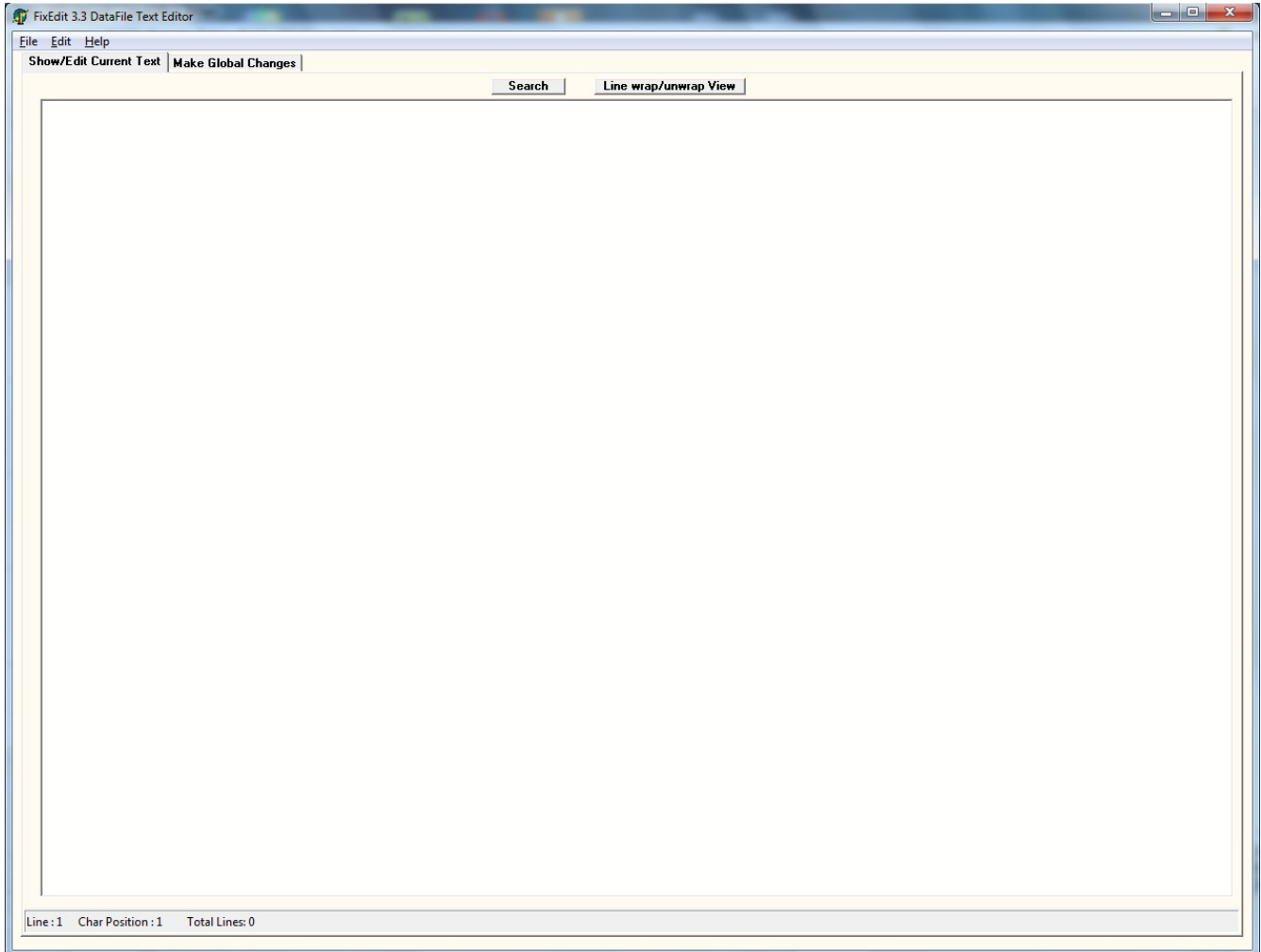
The end usage of the program will be a product of the user's imagination.

Fixedit's main screen looks at first to be a very ordinary text editor, of which there are a great many. As a plain editor, it has a few characteristics not shared by most others.

If you exit, it does **NOT** warn about a Save. That's up to you. It also will **NOT** warn you about overwriting an existing file. If you wish to save, Fixedit does not use **File,Save**. It only has **File,SaveAs** so you will not automatically overwrite the original without filename checking.

You can paste formatted text into Fixedit, or you can change the on-screen font, and it will display and print as such, but if you save it, it will be saved as plain, unformatted text. If you attempt to open a file containing non-text, binary null characters, or Unicode characters for foreign language display, Fixedit will ask if you wish to open the file with the null characters (ASCII 0) removed. Fixedit will only open the file if you choose so, and the file will be opened as plain text. An Example of a file with imbedded nulls are Windows Bootlogs, which use Unicode text.

Main Screen – Show/Edit Current Text



Fixedit has no real file size limit in WinXP, Vista, or Win7. It will load almost any size text file. The clipboard usage also has no realistic limit. Pretty much any size file can be loaded from **File, Open**. Huge files (200k+ lines) may take a few minutes to load on a fast machine.

Fixedit will NOT ever wrap or unwrap lines as you type. If you click the "**Line wrap/unwrap View**" button, it will wrap each line if necessary, but for viewing only. The text will be saved as Plain Unwrapped text. You can, using the "Make Global Changes" tab, change a document into one that has all lines capped at a given width, but the document then will have "hard", fixed carriage returns. Fixedit can not undo those hard carriage returns to set a different width, except manually, or one line at a time.

Unless you purposely overwrite the original name using **File, SaveAs**, any document opened with **File, Open** will, by default, be left unmodified. The original file is closed as soon as its contents are loaded into Fixedit.

Just click the Search button to search the document. No pull-down menus needed.

You can change the font from **Edit, Fonts**. This does change both the printing font and the on-screen font, but only until you exit the program. The text that is saved has no formatting and is pure ASCII text only. The default font set is Trebuchet MS, 10 point.

File, Print dumps the entire resident text file to the printer in a paged image size that will fit almost any laser or inkjet printer. Pages are not numbered. The default font is used unless a different font has been selected.

Second Tab – Make Global Changes

FixEdit 3.3 DataFile Text Editor

File Edit Help

Show/Edit Current Text Make Global Changes

Test Each Line Of Text To See Whether It :

- PASSES - Each Line Passes, will be changed if possible
- Is Contained within a specific range of text line numbers
- Contains the Test Key anywhere
- Does NOT Contain the Test Key anywhere
- Contains the Test Key starting at the position shown in 1st char position #
- Does NOT Contain the Test Key starting at position shown in 1st char position #
- Contains the Test Key Text at the end of the line
- Is Shorter than Linlength Num Characters
- Is Longer than Linlength Num Characters
- Contains only whitespace or is empty
- Occurs right after a blank (whitespace) line
- Appears to contain all or parts of HTML tags < >
- Begins With A Number (ignoring whitespace)
- Is 2nd of 2 sequential blank lines in file

1st Line Num Last Line Num Linlength Num Chars 1st char position # Last char position Repl Pos

Test Key Text Replacement Key Text

Make The Following Change To Each Line Passing The Test, If Possible

- NO CHANGE in the line except spaces in place of Tabs
- Sort Current Text by 1st char position # entry - Ignore Test
- Insert spaces at line beginning to make Linlength Num Chars
- Insert spaces at line end to make Linlength Num Chars
- Insert Replacement Key Text at Repl Pos character #
- Insert a NEW line containing Replacement Key after test line
- Insert a NEW line containing Replacement Key before test line
- Replace the Test Key Text with the Replacement Key Text
- Replace entire line with Replacement Key Text
- Replace line segment left of the Test Key with the Replacement Key Text
- Replace line segment right of the Test Key with the Replacement Key Text
- Delete from the line every character appearing in the Replacement Key Text
- Remove All Duplicate Lines [Case Sensitive] - Ignore Test
- Start Newline (hard CR) on spaces or commas at Linlength Num Chars
- Treat 1st two chars in Replacement key as phrase begin/end markers-delete FIRST pair of markers in the line and chars in between
- Treat 1st two chars in Replacement key as phrase begin/end markers-delete LAST pair of markers in the line and chars in between
- Delete any whitespace at the beginning and end of the line
- DELETE the entire line
- Delete the line segment right of the Test Key Text
- Delete the line segment left of the Test Key Text
- Delete Test Key Text from the line
- Delete a specific range of character positions
- Retain only a specific range of character positions
- Truncate right side of the line to Linlength Num Chars
- Start new line exactly at Linlength Num Characters
- Add Replacement Key Text at end
- Attempt to Strip HTML tags, including multi-line
- Start new line at beginning of each Test Key Text
- Strip Backspaced Overstrikes and Underlines
- Move (Unwrap) next line to end of Test line

Make Changes To Current Text

OK

Retain only the lines that pass the Test Output upper case Test Key case sensitive Output lower case

The "Make Global Changes" tab is where Fixedit departs from other editors.

OVERVIEW:

UPPER (RED) AND LOWER (GREEN) BLOCKS

When edit changes are made using this page, the entire file is evaluated, line by line, and decisions are made about what changes to make to which lines. These decisions are based on the "radiobuttons" and "checkboxes" checked, as well as the entries you made in the text fields when you click the OK button.

Notice that the upper (RED) block of buttons is labeled "Test Each Line Of Text To See Whether It..". You may only choose one of these buttons at any one time.

This block of "radiobuttons" performs the stated test(s) on each line of the file to determine whether (yes/no) that line will be modified per the green block of instructions, or whether the line will be left alone.

The lower (GREEN) block of buttons is labeled "Make This Change to Each Line Passing the Test Where Possible". The selection here determines the change(s) made to the lines that pass the test from the red block. (Any lines not passing the red block test are generally left alone).

CENTRAL TEXT FIELDS

The central block of Text Fields uses manual entries to provide extra data for either the Test (red) block or the Change (green) block, or both.



SPECIAL CASE CHECKBOXES

The four checkboxes at the bottom may take effect anytime the OK button is clicked.

Retain only the lines that pass the Test Output upper case Test Key case sensitive Output lower case

The first checkbox on the Left allows you to *delete* any line not passing the red block test, rather than leaving it in the file.

In case a test involving a "Test Key Text" is chosen, the middle checkbox forces an exact match of a "Test Key Text" upper/lower case characters with some part of the line before the line test "passes".

The upper/lower case checkboxes will change the resulting file to upper or lower case when the **OK** button is clicked, regardless of what tests were performed or what line changes were made.

SPECIAL CASE CHECKBOXES will Reset to **Unchecked** after each Global change (**OK** Button) to the text. You will see a message box as a reminder.

Global Changes Tab Features - UPPER (RED) BLOCK

Let's Examine the Button Functions in the Upper (**RED**) Block.

Block label is "**Test Each Line of Text To See Whether It.**"

Fixedit looks in this **RED** block to see which lines to operate upon. Lines not affected by the chosen button here will be left alone. (There are a couple functions that operate independently of the line test-those are discussed later). Lines passing the "test" described by the chosen button will be changed according to the selection in the **GREEN** block.

"**PASSES-Each Line passes, will be changed if possible**" is the default choice, and means that **EVERY** line will be affected, if possible, by the chosen button in the **GREEN** block.

"**Is contained within a specific range of line numbers**" means Fixedit will look at the entry boxes for "1st Line Num" and "Last Line Num". Lines including and between the two entries will be the only lines affected by the instruction in the **GREEN** block. If "1st Line Num" has a 2 and "Last Line Num" has a 27, only the 2nd-27th lines will be affected.

"**Contains the Key text Anywhere**" means Fixedit will look at the manual entry for "Test Key Text". Any line that contains the key text will be changed according to the **GREEN** block. This is a test involving the "Test Key Text", so unless the checkbox at the bottom of the page is checked, it will use upper and lower case interchangeably to test for a match.

"**Does NOT Contain The Test key Anywhere**" means Fixedit will look at the manual entry for "Test Key Text". Any line that does **NOT** contain the key text will be changed according to the **GREEN** block. Unless the checkbox at the bottom of the page is checked, it will use upper and lower case test key text interchangeably.

Global Changes Tab Features - UPPER (RED) BLOCK , continued

"Contains the Test Key Starting at the Position shown in 1st Character Position" means Fixedit will look at the "Test Key Text" and also the number showing in the "first char position #". If the box "first char position #" contains a 9, then only lines containing the Test Key text starting at the 9th character would pass the test and be changed by the GREEN block. Unless the checkbox at the bottom of the page is checked, it will use upper and lower case test key text interchangeably.

"Does NOT Contain the Test Key Starting at the Position shown in 1st Character Position" means Fixedit will look at the manual entries for "Test Key Text" and also the number showing in the "first char position #". If the box "first char position #" contains a 9, then only lines NOT containing the Test Key text starting at the 9th character would pass the test and be changed by the GREEN block. Unless the checkbox at the bottom of the page is checked, it will use upper and lower case test key text interchangeably.

"Contains the Test Key Text at the end of the line" means Fixedit will look at the "Test Key Text" and determine if the last set of characters in the line matches the "Test Key Text". If it matches, the line would pass the test and be changed by the GREEN block. (Best to make a separate run first with the default "PASSES.." setting and "Delete Any Whitespace.." then hit OK. This to be sure there are no invisible trailing spaces at the end of the line).

"Is Shorter than Linelength Num Characters" means Fixedit will look at the manual entry "Linelength Num Chars" and compare the number entered to the length of the line. Any line that is shorter than the numerical entry in "Linelength Num Chars" will be changed according to the GREEN block.

"Is Longer than Linelength Num Characters" means Fixedit will look at the text box labeled "Linelength Num Chars" and compare the number entered to the length of the line. Any line that is Longer than the numerical entry in "Linelength Num Chars" will be changed according to the GREEN block.

"Contains only whitespace or is empty" means Fixedit will look at the line to see whether it has any characters other than spaces, or tabs in addition to the usual CR/LF. Any line which does NOT have any other characters and would print as a blank line passes the test and will be changed according to the GREEN block.

"Occurs right after a blank (whitespace) line" means Fixedit will look at the *previous* line to see whether it is a blank. If so, then *this* line passes the test and will be changed according to the GREEN block.

"Appears to contain all or parts of HTML tags < >" means Fixedit will look at the line to see whether it has any characters usually associated with HTML formatting (< > characters). Any line which has these characters passes the test and will be changed according to the GREEN block.

"Begins with a Number (ignoring whitespace)" means Fixedit will look at the line to see its first printable character is a numeric. If there are leading spaces/tabs, they are ignored. Any line which has a first recognizable numeric character passes the test and will be changed according to the GREEN block.

"Is 2nd of 2 Sequential blank lines in file" means Fixedit will look at the line to see if the current line AND the previous line are both just whitespace. If so, the current line passes the test and will be changed according to the GREEN block.

Global Changes Tab Features - LOWER (GREEN) BLOCK

Let's Examine the Button Functions in the Green (Lower) Block.
Block label is "Make the Following Change to Each Line Passing the Test, If Possible".
Fixedit looks in the green block to see what changes to make in the lines passing test, based on the radiobutton chosen and the parameters in the central text fields.

Global Changes Tab Features - LOWER (GREEN) BLOCK , continued

These next two buttons in the GREEN block that are completely independent of the settings in the upper RED block line test.

"Sort Current text by 1st char position # entry - Ignore Test"

The Sort Text button does use the manual entry for "1st Char Position". If there is an entry there, the sort is done on that character in each line. Otherwise it uses the first character in each line.

"Remove All Duplicate Lines [Case Sensitive] – Ignore Test"

Removing Duplicate lines is the slowest operation Fixedit performs. For a file with 80,000 lines, it takes about 1.5 minutes on a 2.8Ghz single core machine. A 10, 000 line file takes about 2 seconds. A 500,000 line file, about an hour. Be patient until the "busy" cursor disappears.

If you choose "Remove All Duplicate Lines..", and don't wish to have upper/lower case differentiation, leave the first (default) buttons checked in both Red and Green blocks, open the file, click the checkbox at the bottom to "Output lower case", and click OK.

This changes the whole file to lower case, and you can then Remove the Duplicate lines.

"NO CHANGE in the line except spaces instead of tabs" means Fixedit will not change content of each line(except tabs). This is the default, and is mostly used when you have the SPECIAL CHECKBOX "Retain only the lines that pass the Test Parameter, Discard the Rest" checked to just retain certain lines.

"Sort Current text by 1st char position # entry - Ignore Test" covered above.

"Insert Spaces at the line beginning to make Linelength Num characters" means Fixedit will read the manual entry in CENTRAL TEXT FIELDS for Linelength Num Chars. Any line shorter than the entry will have spaces padded on the front so it comes out the exact length. Lines longer than Linelength Num Chars will not be altered. Only lines passing the test from the RED block will be changed.

"Insert Spaces at line end to make Linelength Num characters" means Fixedit will read the manual entry in CENTRAL TEXT FIELDS for Linelength Num Chars. Any line shorter than the entry will have spaces padded at the end of the line so it comes out the exact length. Lines longer than Linelength Num Chars will not be altered. Only lines passing the test from the RED block will be changed.

"Insert replacement Key Text at Repl Pos character #" means Fixedit will read the manual text entry in the CENTRAL TEXT FIELD "Replacement Key Text" and will also read the number entered in "Repl Pos". It will enter the Replacement Key Text at the character number specified in each line. Only lines passing the test from the RED block will be changed.

"Insert NEW line containing Replacement Key text after test line" means Fixedit will read the manual text entry in CENTRAL TEXT FIELD "Replacement Key Text". After any line passing the test from the RED block, a new line containing the Replacement Key Text will be inserted into the file. Only lines passing the test from the RED block will be affected.

"Insert NEW line containing Replacement Key text before test line" means Fixedit will read the manual text entry in CENTRAL TEXT FIELD "Replacement Key Text". For every line passing the test from the RED block, a new line containing the Replacement Key Text will be inserted just before it in the file. Only lines passing the test from the RED block will be affected.

"Replace the Test Key Text with the Replacement Key Text" means Fixedit will read the two manual text entries in the CENTRAL TEXT FIELDS and substitute in the entire file – this is same as a standard Find and Replace ALL. Any line not containing the Test Key text will not be altered. Only lines passing the test from the RED block will be changed.

"Replace entire line with Replacement Key Text" means Fixedit will read the manual entry in the CENTRAL TEXT FIELDS for Replacement Key Text and substitute that text for any line which passes the RED block test.

Global Changes Tab Features - LOWER (GREEN) BLOCK , continued

"Replace the line segment left of the Test Key Text with the Replacement Key" means Fixedit will check each line for presence of the Test Key Text. If found, it will delete all characters in the line up to the beginning of the Test Key Text, and insert the Replacement Key Text at the beginning of the line. Only lines passing the test from the RED block will be affected.

"Replace the line segment right of the Test Key Text with the Replacement Key" means Fixedit will check each line for presence of the Test Key Text. If found, it will delete all characters in the line after the end of the Test Key Text, and insert the Replacement Key Text at the end of the line. Only lines passing the test from the RED block will be affected.

"Delete from the line every character appearing in the Replacement Key Text" means Fixedit scans the line for all characters appearing in the manual entry for Replacement Key Text. Any matching character will be deleted from the line. Only lines passing the test from the RED block will be affected.

"Remove All Duplicate Lines [Case Sensitive] – Ignore Test" - see above.

"Start NewLine (hard CR) on spaces or commas at Linelength Num Chars" means Fixedit will read the manual entry in CENTRAL TEXT FIELDS for Linelength Num Chars, and attempt to wordwrap each line at the closest space or comma. Only lines passing the test from the RED block will be affected. This is permanent (no undo).

"Treat 1st two chars in Replacement key as phrase begin/end markers-delete 1st pair of markers in the line and chars in between" means Fixedit will look at the manual entry for Replacement Key Text, and pick up the first two characters. Any line containing those two characters, in that order, will have the first pair removed along with any characters in between.

Only lines passing the test from the RED block will be affected.

Example - if Red Block has default "PASSES..", and Replacement Key Text has left and right parentheses in the entry, like so: ()

A Line looking like this:

"Every day (By the way) that I see malware (By the way) I want to fix it."

When the OK button is pressed once, would change to this:

"Every day that I see malware (By the way) I want to fix it."

Pressing the OK button again, it would change to this:

"Every day that I see malware I want to fix it."

Using left and right square brackets in Replacement Key text and clicking OK repeatedly will end up changing bbcode text to unformatted text. Only problem will be the deletion of imbedded URL's and quote attributions.

"Treat 1st two chars in Replacement key as phrase begin/end markers - delete LAST pair of markers in the line and chars in between" means Fixedit will look at the manual entry for Replacement Key Text, and pick up the first two characters. Any line containing those two characters in that order will have the last pair removed along with any characters in between. Only lines passing the test from the RED block will be affected.

This is the same as the last example, except it starts at the end of the line and looks backwards. The last instance in each line is removed first.

"DELETE the entire line" means Fixedit will remove any line passing the RED block test.

"Delete any whitespace at the beginning and end of the line" means Fixedit will remove any spaces or tabs from the beginning and from the end of the line. Only lines passing the test from the RED block will be affected. (Good procedure to run before any end of line tests).

"Delete the line segment right of the Test Key Text" means Fixedit will check each line for presence of the Test Key Text. If found, it will delete all characters in the line after the end of the Test Key Text. Only lines passing the test from the RED block will be affected.

Global Changes Tab Features - LOWER (GREEN) BLOCK , continued

"Delete the line segment left of the Test Key Text" means Fixedit will check each line for presence of the Test Key Text. If found, it will delete all characters in the line up to the beginning of the Test Key Text. Only lines passing the test from the RED block will be affected.

"Delete the Test Key Text from the line" means Fixedit will check each line for presence of the complete Test Key Text phrase. If found, it will delete the complete Text Key Text phrase, leaving everything else as it was. Only lines passing the RED block will be affected.

"Delete a specific range of character positions" means Fixedit will read the numbers from the manual entries in "1st Char position" and "Last char position", and for each line, will remove characters starting at 1st char position , and stopping at Last char position. Any parts of the line before and after those position numbers will be left untouched. Only lines passing the test from the RED block will be affected.

"Retain only a specific range of character positions" means Fixedit will read the numbers from the manual entries in "1st Char position" and "Last char position", and for each line, will remove all characters until it gets to the 1st char position, and also will remove all characters in the line starting after the "Last char position". Only lines passing the test from the RED block will be affected.

"Truncate right side of the line to Linelength Num Characters" means Fixedit will read the manual entry for "Linelength Num Chars" and remove all characters in the line after that. An entry of 70 would mean that every line longer than 70 characters would have all characters after the 70th cut off the end of the line, without regard for words, spaces, etc. Only lines passing the test from the RED block will be affected.

"Start new line exactly at Linelength Num Characters" means Fixedit will read the manual entry for "Linelength Num Chars". If it were number 75, a CR-LF would be inserted every 75 characters exactly, regardless of words, spaces, etc. Enough new lines will be generated, each one having 75 chars, until the end of the original line is reached. Only lines passing the test from the RED block will be affected.

"Add Replacement Key text at end of line" means Fixedit will read the manual entry for "Replacement Key Text" and add it on the end of every line Passing the RED block test.

"Attempt to Strip HTML tags, including multi-line" means Fixedit looks for the presence of formatting tags < >. When it sees the first open tag < , it removes all text in the file until it gets to the close tag >. It will remove as many lines as necessary until it finds the close tag. Then starts looking for an open tag again. To avoid unexpected results, this command is best used with the RED block set for the default "PASSES..."

This:

```
<br></td></td></tr></table></TABLE><br>
<a name="Operating_System"></a>
<TABLE WIDTH="100%" BORDER=0 CELLSPACING="0" ALIGN="CENTER">
<a href="#TOC">Top</a>
<CAPTION ALIGN="TOP"><FONT size="5">Operating System</FONT></CAPTION>
<TR ALIGN="LEFT" BGCOLOR="#00FF00"><TH>Property</TH><TH>Value</TH></TR>
<TR BGCOLOR="#F0F0F0"><TD>Name</TD><TD>Windows XP (Personal) Service Pack 2</TD></TR>
<TR><TD></TD><TD>Terminal Services in Remote Admin Mode</TD></TR>
<TR BGCOLOR="#F0F0F0"><TD></TD><TD>Uniprocessor Free</TD></TR>
<TR><TD></TD><TD>Running on the console.</TD></TR>
```

continued next>

Global Changes Tab Features - LOWER (GREEN) BLOCK , continued

Becomes this:

[Top](#)

[Operating System](#)

[PropertyValue](#)

[NameWindows XP \(Personal\) Service Pack 2](#)

[Terminal Services in Remote Admin Mode](#)

[Uniprocessor Free](#)

[Running on the console.](#)

You can frequently make readable plain text out of a clipboard copy of a web page this way, since some pages can't be saved as readable text by the browser.

“Start new line at beginning of each Test Key Text” means Fixedit reads the manual entry for “Test Key Text”, and each time the Test Key Text is found in a line, a new line is started at the beginning of the occurrence. Only lines passing the test from the RED block will be affected.

“Strip Backspaced Overstrikes and Underlines” means Fixedit will look for backspace characters used to generate Underlines, etc by reprinting the same space on the page more than once. This is used most often in Linux text documents. Underlines and Overstrikes created this way will be removed. Only lines passing the RED block will be affected.

“Move (Unwrap) next line to end of test line” means Fixedit will take any line passing the test in the RED block, and tack the NEXT line onto the end of it. It is not recursive. It then goes to the next untouched line for its test results. If all lines PASSED, the resulting file would have half as many lines. Only a line passing the test from the RED block, and the line immediately following will be affected.

EXAMPLE – Get Rid of lines in a Log File containing certain string text
in the illustration here we will use the string text **Cookie**

Use **File, Open** or **Edit, Paste** to load a log file into Fixedit.

Some sample lines from a fictional log:

C:\Documents and Settings\John\Cookies\John@upi.112.2o7[1].txt -> TrackingCookie.2o7 : Cleaned.
C:\Documents and Settings\John\Cookies\usatoday1.112.2o7[1].txt -> TrackingCookie.2o7 : Cleaned.
C:\Documents and Settings\John\Cookies\John@3.adbrite[2].txt -> TrackingCookie.Adbrite : Cleaned.
C:\Documents and Settings\John\Cookies\John@usatoday1.112.2o7[1].txt -> TrackingCookie.2o7 : Cleaned.
HKLM\SOFTWARE\SurfSideKick3 -> Adware.SurfSide : Cleaned with backup
HKLM\SOFTWARE\SurfSideKick3\Internet Explorer -> Adware.SurfSide : Cleaned with backup
C:\Hjt\backups\backup-20060325-182722-342-meowa.exe -> Downloader.Qoologic.bj : Cleaned with backup
C:\Program Files\SurfAccuracy -> Adware.SurfAccuracy : Cleaned with backup
C:\Program Files\SurfAccuracy\License.Ink -> Adware.SurfAccuracy : Error during cleaning
C:\Program Files\SurfAccuracy\SAccU.exe -> Adware.SurfAccuracy : Error during cleaning
C:\Users\John\AppData\Roaming\Microsoft\Windows\Cookies\Low\John@waterfrontmedia.112.2o7[1].txt -> TrackingCookie.2o7 : Cleaned.

To delete all lines containing the text "cookie":

Click "Make Global Changes" tab, and set these entries:

"Contains the Test Key anywhere"

"Test Key text" = Cookie

"Delete the entire line"

Hit **OK**

Text now will look like this:

HKLM\SOFTWARE\SurfSideKick3 -> Adware.SurfSide : Cleaned with backup
HKLM\SOFTWARE\SurfSideKick3\Internet Explorer -> Adware.SurfSide : Cleaned with backup
C:\Hjt\backups\backup-20060325-182722-342-meowa.exe -> Downloader.Qoologic.bj : Cleaned with backup
C:\Program Files\SurfAccuracy -> Adware.SurfAccuracy : Cleaned with backup
C:\Program Files\SurfAccuracy\License.Ink -> Adware.SurfAccuracy : Error during cleaning
C:\Program Files\SurfAccuracy\SAccU.exe -> Adware.SurfAccuracy : Error during cleaning

With other large log files you can do a similar thing, by typing in different **Test Key Text** entries to select lines that will pass a test for removal. These **Test Key Text** entries might be useful:

C:_OTL\
\Qoobox\
\Quarantine\
\System Volume Information_restore

Another Example

Take a large list of pathnames and produce just a list of all folders contained in the list. The sequence would be - import the original file, add a backslash marker to the end of each line, use the last backslash pair to remove all the filenames, then remove duplicated folder lines.

Import the file. A small sample of the lines might look like:

```
C:\WINDOWS\System32\svchost.exe
C:\WINDOWS\system32\Ati2evxx.exe
C:\Program Files\Common Files\Symantec Shared\ccSetMgr.exe
C:\Program Files\Common Files\Symantec Shared\ccEvtMgr.exe
C:\WINDOWS\Explorer.EXE
C:\WINDOWS\system32\spoolsv.exe
C:\Program Files\CyberLink\PowerDVD\PDVDServ.exe
C:\Program Files\Symantec\LiveUpdate\ALUSchedulerSvc.exe
```

Add Settings in the Global tab:

"PASSES- Each Line Passes, will be changed if possible"

"Replacement Key Text" = \

"Add Replacement Key text at end of line"

Hit **OK**. Text will now look like this:

```
C:\WINDOWS\System32\svchost.exe\
C:\WINDOWS\system32\Ati2evxx.exe\
C:\Program Files\Common Files\Symantec Shared\ccSetMgr.exe\
C:\Program Files\Common Files\Symantec Shared\ccEvtMgr.exe\
C:\WINDOWS\Explorer.EXE\
C:\WINDOWS\system32\spoolsv.exe\
C:\Program Files\CyberLink\PowerDVD\PDVDServ.exe\
C:\Program Files\Symantec\LiveUpdate\ALUSchedulerSvc.exe\
```

"Replacement Key Text" = \\

"Treat 1st two characters in Replacement Key as phrase begin/end markers-delete LAST pair of markers in the line and chars in between"

Hit **OK**. Text will now look like this:

```
C:\WINDOWS\System32
C:\WINDOWS\system32
C:\Program Files\Common Files\Symantec Shared
C:\Program Files\Common Files\Symantec Shared
C:\WINDOWS
C:\WINDOWS\system32
C:\Program Files\CyberLink\PowerDVD
C:\Program Files\Symantec\LiveUpdate
```

CHECK box "Output lower case" at the bottom of the page to make sure it sees things like "System32" and "system32" as the same when it removes duplicates.

Hit **OK**.

"Remove all Duplicate Lines [Case Sensitive] – Ignore test"

Hit **OK**. Text will now look like this, All done. Save it as your folder list.

```
c:\windows\system32
c:\program files\common files\symantec shared
c:\windows
c:\program files\cyberlink\powerdvd
c:\program files\symantec\liveupdate
```

A Multi-Action EXAMPLE

An illustrative example of making a blacklist for gambling sites. Most of these sites are already listed in a HOSTS file like HpHosts or mvps. In this case, choose to get a list of all the lines in the HOSTS file which refer to "poker" or "vegas" or "casino". Use **File,Open** to load a HOSTS file into Fixedit. Not wanting to manually edit 20, 000 lines or more, put the FixEdit to work. The sequence is: mark all the lines we want to keep, delete unmarked lines, cleanup the lines.

Actions on the "Make Global Changes" tab :

"Contains the Key text anywhere"

Set "Test Key text" = **casino** "Replacement Key text" = # "Repl Pos" = **1**

"Insert Replacement Key Text at Repl Pos character#"

Hit **OK**. (You can "peek" at the "Current Text" tab now and after any/each time you hit OK).

Set "Test Key text" = **vegas**

Hit **OK**.

Now every line with the word "vegas" in the text will also have a # as the first character.

Set "Test Key text" = **poker**

Hit **OK**.

Now every line with the word "poker" in the text will also have a # as the first character.

We should now have all the lines with "poker", "casino" or "vegas" content marked with one or more # characters at the beginning. Now let's get rid of all the rest of the lines.

New settings:

"Does **NOT** Contain the Test Key starting at position shown in 1st char position#"

Set "Test Key Text" = # "1st Char position" = **1**

"DELETE the entire line"

Hit **OK**.

Now we have a list with only the sites we want.

Let's clean up the list by getting rid of the # characters.

"PASSES – Each line will be changed if possible"

"Delete Text Key Text from the line"

Hit **OK**. (That gets rid of the # signs we used as markers - we don't need them now)

Set "Test Key Text" = **127.0.0.1**

Hit **OK**. (The selected Action is still to delete the test key. That gets rid of the 127.0.0.1 text)

"Delete any whitespace at the beginning and end of the line"

Hit **OK**. (You can "peek" at the "Current Text." tab now and see your file)

You can click "Sort Current text by 1st char position # entry - **Ignore Test**" and hit **OK**

Now use **File,SaveAs** to save the file as name "Gambling Blocklist.txt"

Latest FixEdit Update Note - Version 3.3, June 2015

Fixed bug in Test Key text position sensing.

Updated User Guide.

New form color

Changed to bolder fonts in Global Changes tab for better visibility.