Main pre-editing tool – FRedit

*0. You can actually **see** how FRedit works*

There are loads of videos showing you how FRedit works:

FRedit for Beginners Part 0 (4:15) – Getting a copy of the macro and installing it: youtu.be/B7ouU3OzWRE
FRedit won't work! (3:20) – Help with trying to get FRedit working for the first time youtu.be/nVcneZZgV2g
FRedit for Beginners Part 1 (4:17) – How to get started: youtu.be/X9e7770QWiY
FRedit for Beginners Part 2 (3:50) – More techniques for file clean-up: youtu.be/4hlnYqyfOQk
FRedit for Beginners Part 3 (4:13) – Learn more things FRedit can do for you: youtu.be/GO8iW0WBfp0
FRedit for Beginners Part 4 (5:08) – Adding wildcards into your FRedit armoury: youtu.be/Fq6p_RdMHb4
FRedit for Beginners Part 5 (5:24) – Adding and removing formatting: youtu.be/MXiPpz0yVQE
FRedit for Beginners Part 6 (5:53) – A few tricks of the trade: youtu.be/ohPwmX1mS00
FRedit for Beginners Part 7 (16:31) – The FRedit Library: youtu.be/EulSXDNJw0k
Advanced FRedit use (10:49) – Some high-power uses of FRedit: youtu.be/wHR7Yl_dBjI
FRedit queries 1 (7:58) – Answers to a couple of queries about FRedit: youtu.be/QYEIVbmIMQA
FRedit revision 1 (15:08) – Reinforcing the basic principles of FRedit: youtu.be/Mt4iuh6SOAM
FRedit revision 2 (5:50) – More on applying attributes, plus another gotcha: youtu.be/aQ_hR_K-INM
FRedit with notes and text boxes (4:58) – Using FRedit when your text has foot/endnotes and/or text boxes
youtu.be/bGIMXppJlFM
FRedit and MultiSwitch (9:09) – Differences and similarities between these two very useful macros
youtu.be/yGZHej6vaZ4
New way to run FRedit (5:41) – Run FRedit by selecting a FRedit list from a menu youtu.be/1bVduGAFrhU
FReditListChecker – Check your FRedit list for some of the 'obvious' errors: youtu.be/Z7cjf446JWM

1. How the macro works

FRedit provides an easy way to do a sequence of global find-and-replaces (F&Rs) on a document that you are working on. Simply create a list of F&Rs in a new Word file (or open an existing one) and run the FRedit macro; it looks through your F&R list file (or 'script') and in a matter of seconds makes all the changes you specify, one by one, to the document you have open.

The important point is that there is only *one* macro and, once it is loaded, you never need to touch it again. Also, you can have any number of different F&R lists for different jobs and/or different clients – just open the list you want to use, and run the macro.

2. Installation and operation

2.1 Setting up

1. Open the file *2_Macro* in Word, press Ctrl-A to select all the text and Ctrl-C to copy it. (*Mac: ⌘-A and ⌘-C*).

2. Click Alt-F8, which brings up the 'Macros' dialogue box (*Mac: Option-F8, or menu Tools—Macro—Macros.*) In the 'Macro name:' slot, type: Dummy and then click 'Create'.

This will load an application called Visual Basic, and create a 'dummy' macro:

```
Sub Dummy()
'
' Dummy Macro
'
```

```
'
End Sub
```

3. Select the whole of this dummy macro, from Sub to End Sub:

```
Sub Dummy()
'
' Dummy Macro
'
'
End Sub
```

and then use Ctrl-V to paste the text of the new macro in its place. Close Visual Basic, by using Alt-Q (for **Q**uit) (*Mac: ⌘-Q*).

4. To assign a keystroke to your macro you need the ‘Customize Keyboard’ dialogue box. This is called up in different ways:

Word 2002/3: Open the menu Tools—Customise, and then click ‘Keyboard’ (near the bottom of the dialogue box).

Word 2004: Open the menu Tools—Customize—Customize Keyboard.

Word 2007: Click on the ‘Customize Quick Access Toolbar’ menu (the little down-arrow at the right-hand end of the quick access toolbar) and choose ‘More Commands’. Find ‘Keyboard Shortcuts’ at the bottom of the box and click on ‘Customise’.

Word 2010: Right-click in a blank piece of the ribbon and click on ‘Customize the Ribbon’, and then underneath the left-hand of the two columns of options is a separate item ‘Keyboard shortcuts: **Customise…**’, so click on ‘**Customise…**’

Word 2011: Open the menu Tools–>Customize Keyboard.

(All versions…) In the ‘Customize Keyboard’ dialogue box that appears, in the left-hand list (Categories), select Macros (right near the bottom of the list), and the right-hand list becomes (not surprisingly) ‘Macros’. Select ‘FRedit’ and click in ‘Press new shortcut key’ *(or ‘Press new keyboard shortcut’)*. So press the key combination you’ve decided to use for running FRedit – something you can remember. Then a line will appear saying, hopefully, ‘Currently assigned to: [unassigned]’ which means that no other function is using that key combination. However, if it says that it is currently assigned to another function, you have to decide either to choose a different key combination, or to assign it to that combination anyway, in which case the original use of that combination will be no longer available. Once you’ve settled on a keystroke to use, click ‘Assign’ and then ‘Close’.

2.2 Testing the macro

1. Close the file *2_Macro* and copy both *3_Sample_text* and *4_Sample_List* into a suitable folder on your computer, and then open both files in Word. (It’s easier if you make sure that you close all other Word files, so you only have these two files open.)

2. Run the macro by pressing the keystroke you selected. You should now see that the contents of the *3_Sample_text* file have changed. Use Ctrl-Z a number of times to undo the changes and run the macro again so that you can see clearly what it is changing.

3. Using the macro

First an important principle that you must get hold of about how FRedit works.

Whenever you use FRedit, there must be (at least) two Word files open: (1) The text that you're trying to edit, and (2) a list of the particular F&Rs that you want FRedit to do for you on that text.

So, there is just the one macro – FRedit – and then two open files: the text to be edited and the FRedit list.

*(If an area of text is selected, FRedit works on **only** that bit of the text. This can be confusing if you accidentally select a bit of text, so make sure that no text is selected when you run FRedit.)*

3.1 What can I change?

FRedit can do most of the things that you can do with the normal F&R dialogue box. For example, you could use it to change double spaces into single spaces, or <space><hyphen><space> into <space><en dash><space>, and you could use it for changing alternative spellings, such as:

focuss|focus
co-operat|cooperat
co-ord|coord

The vertical bar character ('|') is used to separate the 'Find' from the 'Replace'. (The keystroke for '|' is a shifted backslash at the bottom left of many keyboards, although on some keyboards, the keytop is marked as a sort of elongated colon – a vertical bar with a gap in the middle.)

The F&R list can also contain any of the special operators that you normally use with Word's F&R, so you can have items such as:

^p^p|^p
^p^32|^p
^32^p|^p
^t^32|^t

The ^p is a new paragraph (return), and the ^t, a tab (there's a list of Word's '^' codes in Appendix 1 of my free macro book). The ^32 is the ASCII code for a space; you could use an actual space in the list, but using ^32 instead makes it more easily visible.

As well as straightforward find and replace of text, you can use some of the other facilities offered by Word's Find & Replace dialogue, as I'll now explain…

3.2 Wildcard searches

(If you haven't yet learned how to do wildcard F&R, don't worry. I'm just saying that FRedit can do wildcard F&Rs if and when you want them.)

Your list can contain wildcard searches, just as you might use in normal F&R. For example, you can add a full point to every 'etc' by using:

~<etc>|etc.

Let me make it bigger and use highlighting to show up the different elements of the line:

~<etc>|etc.

The Find and Replace elements are the bits I've coloured in turquoise and green, respectively. The vertical bar character between them you've already met, but the tilde at the beginning of the line is to tell FRedit that this line is to be executed as a wildcard F&R, not an ordinary F&R.

3.3 Case sensitivity

Generally, the macro works in a case-sensitive mode; so, for example, you might use:

co-operat|cooperat
Co-operat|Cooperat

This would catch both capitalisations – but it would not catch 'CO-OPERATION'.

However, you can put a '¬' character at the beginning of the line (on most keyboards, the '¬' is the shifted back-tick key to the left of the number-1 key). The 'bent pipe' symbol *(apparently, it's officially called the 'not sign', as in negative)* tells FRedit that you want the F&R to be executed case-insensitively, so the line

¬co-operat|cooperat

will do the same as the two lines above, as well as also changing CO-OPERAT into COOPERAT.

3.4 Highlighting the changes

If **the whole of a line** in the list is highlighted in a particular colour, the changes in the text will be highlighted in that same colour. (Similarly, if the whole line is in a specific text colour, the changes in the text will be in that same text colour.)

For example:

focuss|focus
co-operat|cooperat
favor|favour
humor|humour
~([0-9])-([0-9])|\1^=\2

So each of these different changes will be highlighted or font coloured accordingly.

People ask, "Why highlight the whole line?" The answer is complicated, so if your learning style is "Just tell me what to do; I'll do it" then fine. But if, like me, you're more of an "I learn best by understanding how and why things work", then turn to Appendix 1, below.

3.5 One-off F&Rs

If you just want to do one or two quick F&Rs, you can either open a new blank Word document and type them in there, or you can type them in at the top of an already open list file. If you follow it/them with a hash symbol on a new line, FRedit will ignore the rest of the list. For example, if you wanted to try out the 'etc' wildcard F&R mentioned above, you could just add it at the top of your existing list:

~<etc>|etc.
#
| This is the rest of my list
^p^p|^p
^p^32|^p
^32^p|^p
| etc, etc.

So, in this list, only the wildcard line will be executed when you run FRedit. When it gets to a line starting with ‘#’ it stops.

(Note that the line saying, ‘| This is the rest of my list’, is just a comment, and not a FRedit item. Having the ‘|’ as the very first character of a line tells FRedit that this line is ***not*** an F&R.)

3.6 Ignoring selected parts of the text

If you want to protect parts of the text so that they are not altered when you run the macro (for example, quoted text or reference lists) then all you need to do is to put the font effect ‘strike-through’ on those sections of text that you want to protect. It will look like the following line:
~~“This is a direct quote, so it must ***not*** be changed by FRedit’s find and replace items, on pain of death or worse.”~~

The macro will now *only* operate on those parts of the text that are *not* struck-through.

If you’re likely to use this facility a lot, it’s probably worth having a keystroke that executes the macro:

```
Sub Ignore()
  Selection.Font.StrikeThrough = wdToggle
End Sub
```

so that you can apply and remove the strike-through easily. (There’s a more sophisticated version of this simple macro in my book – called *StrikeSingle* – which stops the adding of the strike-through itself being tracked.)

(Hint: The macro *QuotationMarker* can be used to automatically apply strike-through to all the quotations – single quote marks, double quote marks or displayed quotes – in your text, which then are protected from being changed by FRedit.)

3.6a FRediting ONLY selected parts of the text

A new feature has been added. If, at the beginning of the macro, you change the line:

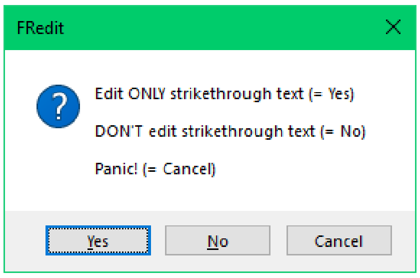
```
useNewStrikeFeature = False
```

to

```
useNewStrikeFeature = True
```

then each time you run FRedit, you can choose whether to (a) (as at present) **not** edit any of the text that is struck through or (b) **only** edit the text that is struck through.

So, if you want to make some global F&R changes to, say, **only** the text that is in style “Body text”, you can select those bits of the text, then apply the single strikethrough attribute, then run FRedit and, when asked:


FRedit

Edit ONLY strikethrough text (= Yes)

DON'T edit strikethrough text (= No)

Panic! (= Cancel)

you press Y.

3.7 F&Rs that add formatting/effects

As with adding highlighting and text colour (see 3.4 above), if you make the **whole of an F&R line**, say, italic, then the resulting text change will be in italic, ditto bold or superscript or... well, here's the full list of attributes:

bold, *italic*, underline, $^{\text{superscript}}$, $_{\text{subscript}}$, ALL CAPS and SMALL CAPS, plus also font colour, font name, font size and highlight colour.

Here are some example FRedit items:

color|colour
favor|favour
Beverley|^&

Can you see what each does? *(Answers at the end of this file.)*

In Section 3.6, I explained that any text that has a single strikethrough is not changed by any of the F&Rs. Well, you can in fact use FRedit to *apply* single strike. However, putting single strike to an F&R means "Do not track this F&R", so you can't use single strike to mean "**apply** a single strike". Instead, I have used *double* strike for this feature (confusing, huh?!). So if any F&R item that has a *double* strike, that means "apply a *single* strike".

But beware, you can't just use such double-strike items at the beginning of your main list, thinking that this will apply single strike and thereby protect that text. That's not how FRedit works, sorry. Instead, you have use a separate FRedit list, to apply the single strike, and then, as a separate action, run FRedit with your main list.

One possible application is:

~~~"*"|^&~~

which is a wildcard F&R to apply single strike to all quoted text. (Unfortunately, you can't do the same for single quotes, because *apostrophes* are also single quote marks. Instead, you could use *QuotationMarker,* which can apply single strike to both single and double.)

This single-strike protection system is used by other macros, such as all three *IS/IZ* macros, *HyphenAlyse*, *ProperNounAlyse*, *SpellingErrorLister*/*Highlighter*, etc.

3.8 Adding styles

As with adding highlighting and text colour (see 3.4 above), if you make the **whole of an F&R line** a particular style, then the change will be made, and that style will be applied. So, for example, to find any line that starts with <number>.<number>, and change it into Heading 2, you would use:

~[1-9].[1-9]*^13|^&

3.9 Changing styles

(Even I find it difficult to get my head around how this works, so maybe leave this until you get more familiar with using FRedit.)
As with formatting changes, to change the style of sections of text, you put the Find text in one style and the Replace text in another. The snag here is that if it's a paragraph style or a list style, you can't have two halves of the same line in different styles! To accommodate this, the macro allows you to put Find and Replace texts on two separate lines. Any given list can contain both single-line and dual-line F&Rs.
So, for example, to find any line in Heading 2 style that, say, starts with <number>.<number>, and change it into Heading 3, you would use:
~~~

~[1-9].[1-9]*^13

^&

To explain how this works, the first line looks (with wildcards on) for any number 1–9, followed by a full point, followed by any number 1–9, followed by any text (*), followed by a paragraph end (^13), and the second line says "replace it with what you have just found", but in Heading 3 style.

If on the other hand, you simply want to force, say, any paragraph starting with '<C>' into Heading 3, you can use a single line:

~\<C\>^13|^&*

(The '\<' is the wildcard way of looking for a '<', and ditto '\>'.) In other words, you only need a two-line F&R if you want to say "find something in this particular style... and do something to it." So, to add a '<B>' in front of every Heading 2 style line, you would use:

Blank

<B>^&

(As explained in Section 3.10, the word 'Blank' is FRedit's way of saying, "Leave the Find box empty".)

3.10 Sample F&Rs

Here are a few examples of the sorts of things you can do with FRedit. To use them, just copy and paste them into your lists.

| Small caps for AD/BC etc
AD|AD
BC|BC
CE|CE
BCE|BCE

|Common chemicals
H2O|H;;2O
CO2|CO;;2
;;2|$_2$

If you look in the FRedit Library, which comes as part of the FRedit package, you will find many, many more examples of things you can do with FRedit! Other people have worked out how to do various clever things with FRedit, and you can just copy and paste them into your own FRedit list.

3.11 Changing everything in a given format

If you want, say, to change *everything* that is in underline into italic, if you do it manually, you leave both the Find box and the Replace box blank and just select the required formats. However, with a FRedit list we can't leave the left half (Find) empty, so we use a special 'code'. So, when FRedit sees that the Find text is just the word 'Blank', it leaves the Find box empty.

So if we want to just change the format of the text then, in the Replace box, we use ^&, which means 'that which you have found'.

Blank|^&

This says, "Find anything that is in underline format and change it into italic format."

Or, if you want to *delete* everything in a particular format, put the word ‘Blank’ in the Replace text too, so when FRedit sees

Blank|Blank

it will find all the text that is in red font and replace it with nothing, i.e. delete it.

However, if you want to *highlight*, say, all the text that was underlined, you must use the ^& version:

Blank|^&

(As mentioned in 3.4, to force something to highlight, you give the whole line that highlight.)

You can also do something similar if you want to change the colour of the highlighting from one colour to another:

Blank|^&

The line above would change all yellow highlighting into turquoise.

3.12 Footnotes and endnotes

If you also want the text of the footnotes to be edited then, at the top of the list, add:

| Footnotes = yes

and similarly for endnotes:

| Endnotes = yes

And if you ***only*** want to edit the footnotes/endnotes but *not* the main text, just add:

| Text = no

3.13 Text boxes

These are a bit of a law unto themselves, but if you find that FRedit doesn’t change the text inside the text boxes, add the following line near the top of the list.

| Textboxes = yes

3.14 Standard clean-up list

I thought it might be helpful to provide a standard list of things that you might want to use in any job. You might not want them all, or you might want to vary them, but it’s a start from which you can vary. (The very first item is born of experience. Some files, especially when converted from other systems, have rogue carriage returns that can cause problems later, so get rid of them first, if they exist.)

```
| multiple returns
~^13{2,}|^p

| multiple spaces
~^32{2,}|^32

| rogue spaces in three places
^32^t|^t
^32^p|^p
```

^p^32|^p

| highlight any tabs at the start of a line
^p^t|^&

| delete tabs on line ends
~^t{1,}^13|^p

| change all ‘space-hyphen-space’ to ‘space-en dash-space’
 - | ^=^32

| change hyphen in number ranges to en dash
~([0-9])-([0-9])|\1^=\2

3.15 Can I use track changes?

If your client wants changes tracked, can you still use FRedit? Well, yes, with care.

As you may know, Word’s F&R can get very confused if you have track changes switched on (even if you just use the normal F&R dialogue box) and especially so with wildcard F&Rs. For example, if you use the ‘dashes in number ranges’ F&R, with track changes on, it changes ‘47-54’ into ‘475–4’! (That’s Word doing that, not FRedit.)

I have therefore added an extra feature into FRedit whereby, if track changes is on, it will track all F&R lines in the FRedit list *except* those that have the strike-through attribute applied to them. So, in the example in section 3.13 above, you might do:

| space-dash-space
 - | ^=^32

| Dash in number ranges
~~~([0-9])-([0-9])|\1^=\2~~

So the ‘space-en dash-space’ change would be tracked, but the ‘dash in number ranges’ would not be tracked (so that it can work properly). And of course you can highlight it so that you know it’s been changed.

However, because FRedit can turn TC on and off, you *can* put TC on some of these difficult F&Rs provided that you have a sufficiently devious mind.

Here’s the example above *with* track changes (explanation below):

~~-|zczc~~

~~~([0-9])zczc([0-9])|\1-\2~~

-|^=

~~zczc|-~~

(Thinking of the above four lines as (a) to (d)...)

With TC off, you (a) change all hyphens to some unique ‘code’, e.g. ‘zczc’, and (b) replace the ‘code’ in number ranges back into real hyphens.

Now, in the clear light of track-change day, (c) you change ‘all’ hyphens (i.e. just those hyphens that are in number ranges) into en dashes.

Turn TC off again, and (d) change all the other code-hyphens back into real hyphens.

I told you it was devious!

So, of those four changes, the only one that the client will actually see tracked is where the hyphens in number ranges being deleted and replaced by en dashes.

Ha! I can now do it with THREE F&Rs:

~~~([0-9])-([0-9])|\1zczc-\2~~
zczc-|zczc^=
~~zczc|~~

3.16 F&R Unicode characters

In the Find box, Word allows you to use, say, ^u8216, where 8216 is the (decimal) Unicode number for an open single curly quote – ‘ – and so you can use this in a FRedit list. However, in Word’s Replace box, you can’t use a ‘^u’ code; you have to use the actual Unicode character. However, FRedit allows the use of Unicodes in Find *and* Replace, but they are in hexadecimal (sorry!). So, for example, to change a spaced em dash (^+) into one that has *thin* spaces (hex 2009) either side, you could use:

^32^+^32|<&H2009>^+<&H2009>

Then if someone has used Greek alphas (α) instead of the proper ‘proportional to’ symbol (∝), you could use:

<&H03B1>|<&H221D>

(Mind you, you could just as well use:

α|∝

I suppose!)

N.B. The 8216 is, as I say, the decimal code for an open curly quote, and if you want to find the decimal value for a character, it’s worth knowing that the *WhatChar* macro gives you the Unicode in both hex *and* decimal.

3.17 Running other macros from within the FRedit list

This feature means that you can run other macros from within the FRedit list. So, in this hypothetical example, various F&Rs are interspersed with running specific macros:

| space-dash-space
~~ – | ^=^32~~

| Dash in number ranges
~~~([0-9])-([0-9])|\1^=\2~~

DoMacro|AutoTagger

~~ ,|,~~
~~ .|.~~
~~ ?|?~~
~~ !|!~~
~~)|)~~
~~ (|(~~

~~DoMacro|SpellAlyse~~
DoMacro|PDFsoftHyphenRestore

Blah blah|blah blah...

As I’m sure you can guess, ‘DoMacro|Whatever’ means ‘At this point in the list, run the macro called “Whatever”’.

But note that if the DoMacro lines is struck-through then, as with any of the F&Rs, this means ‘Do not track the changes while running this particular macro’.

4. Errors

When you get an error from FRedit, there’s usually a good reason – you’ve tried to do something you shouldn’t. However, it might be an error caused by FRedit itself, in which case do let me know, so that I can sort it out and then other people won’t have the same problem.

When an error does occur, see what it says in the title line of the error box. If it says ‘Microsoft Visual Basic’ then it’s probably a programming fault; if so please contact me. However, if it says ‘FRedit’ at the top of the error box then it is FRedit telling you that you’ve done something wrong. Here’s a list of possible error messages and what they might mean.

“No matching replace text”

What is happening is that FRedit has found a line in the FRedit list that doesn’t have a vertical bar character in it. It has therefore assumed that it is the first line of a two-line F&R. However, it can’t find a second line.
Remember that you can have explanatory lines in the list that say, for example:

| This is my list for Bloggs and Co

and because the line starts with a vertical bar character, FRedit knows that this is not an F&R, but there must be a vertical bar character *somewhere* on every line (unless it really ***is*** part of a two-line F&R).

How do you find the rogue line? Well, when FRedit generates the error ‘No matching replace text’, it selects the line containing the rogue text.

“FRedit needs a file to work on, and a list file.”

When you run FRedit, it looks at the files that are open in Word and tries to work out which is the text and which is the FRedit list. However, if you ***don’t*** have both a document containing some text to edit and a FRedit list, then it will rightly complain.

However, to identify a list, FRedit looks to find a vertical bar character within the first few lines of the file. Now, two-line F&Rs don’t have any vertical bar characters, so if your list consists of, say, just a couple of two-line F&Rs, then you need to add a line at the top of the list something like:

| Bloggs’ list

FRedit will then see the vertical bar character and know that this is the FRedit list.

“Please create or load a list.”

It says this if there’s only one file open, i.e. you’ve forgotten to open a FRedit list.

“Too many files are open.”

This error is telling you that FRedit is having difficulty working out which is the FRedit list and which file is the text that you want it to work on. But this error also gives you another sentence to help you see what the problem is...

“Please switch to the text file and rerun the macro.”

FRedit has worked out which is the FRedit list, but there are two or more other files open in Word, and it needs to know which of these files you want it to work on. All you need to do is go to the task bar and select the file that you want FRedit to work on. As long as this file has the ‘input focus’, i.e. it has the cursor in it, FRedit will know that this is the file you want to work on and not any of the others.

“Please switch to the F&R list file and rerun the macro.”

In this case, FRedit has found more than one FRedit list, so it doesn’t know which list to use. So, if you select the list you want, and make sure it has the input focus then FRedit will run that list.

“Please close unused files and rerun the macro.”

FRedit is now totally confused! You’ve given it more than one FRedit list ***and*** more than one non-list file that it could work on. You’ll have to close at least some of the unused list and/or non-list files.

5. Hints

Switch Track Changes On!

It can be very annoying if, when you run FRedit, you forget to switch on track changes before running it.

So FRedit has a facility such that you can add a line at the top of the FRedit list:

| Track = yes

If, when you run FRedit, track changes is switched off, it opens a window to warn you that it’s off, and it won’t let FRedit go ahead until track changes is switched on.

Paragraph problems

It’s important to realise that Word is ***very*** fussy about the use of ^p and ^13 in find and replace. Here are the rules that you must obey:

1) Never use ^p in a ***wildcard Find*** – only ever use ^13. (In fact, FRedit will see this and will warn you that what you have done is wrong.)

2) Never, ***ever*** use ^13 in any ***Replace*** – only ever use ^p.

The reason for (2) is that the ‘^p’ is the thing that holds the formatting information about the paragraph. If you use ‘^13’, you’re saying, ‘Create the start of a new paragraph, but give the paragraph the same style/format as the paragraph that follows.’

Unwanted font size changes

Look at the following FRedit item. Can you spot the problem?

cat|dog

No? Well, I’m not surprised! Try this one:

cat|dog

Now can you see? I have deliberately changed the font size (it’s normally 12pt in this file, but the first is actually 13pt, and the second is 16pt).

So what? Well, if you use either of those F&R items, they will change the font size of the resulting ‘dog’s.

This sort of thing tends to happen where you see an item of text that you want in your FRedit list and, instead of just typing it into the list, you copy and paste it from the other file, bringing with it the font/size/style etc.

The rule is that if FRedit sees that the document’s ‘Normal’ style is, say, 12 point Times New Roman, and the item in the list is, say, 10 point Verdana, then the text will be changed **and** it will be forced into 10 point Verdana.

So, FRedit looks at each line in the list, and if it’s in the ‘Normal’ font size/name then no change is made to the font size/name.

Find and replace (non-)curly quotes

If you want to force all quotes into their curly variety, ‘, ’, “ and ”, then as long as you have the Word’s *AutoFormatCurlyQuotes* switched on, you can just use:

```
'|'
"|"
```

The other way round, if you want all curly quotes back as straight quotes, you have to use the ASCII codes:

```
'|^39
"|^34
```

Adding one effect on top of another, e.g. underlining all italic text

The problem is that if you want to do something with ‘all italic text’, you have to make it not italic. Therefore you have to do something like:

1) Make all italic text (a) not italic (b) a unique font colour

2) Make all text in that unique font colour (a) not coloured (b) italic and underline. Thus...

Blank|^&
Blank|^&

A common misunderstanding

(N.B. This is really, really difficult to understand. Over the years, I’ve tried different ways to explain this, but each time I’ve tried, it’s been difficult for people to comprehend. Work through the following example and see if it helps. Then after the practical example, I’ll try to explain the theory behind it. Good luck!)

STOP PRESS *New idea: Is this explanation any help?!*

The only two things you can do with a FRedit F&R, as regards to any given attribute, is to apply it or to remove it.

This means that you can’t say “Do this F&R, but only to (say) italic text.” You can do so in ‘manual’ F&R, but not in FRedit. So, to make a change to only italic text, you have to make the change and, at the same time remove the italic!

The trick therefore is to make the change, remove the italic, but at the same time apply another different attribute such as underlining (assuming it isn’t used anywhere else in the text!) and then, in a second F&R, find all the underlined text, remove the underlining and add the italic attribute back as before.

Devious, but it works.

In this example the change wanted is: “highlight all superscripted numbers”.

~([0-9]{1,})|\1

~(*)|\1

The first line uses a wildcard expression to find “any superscript number”, remove the superscript and add underlining.

The second line finds any underlined text, removes the underlining, adds superscript, and adds the required highlight.

(Now back to what I wrote a few years before...)

What do you think this F&R should do?

> et al^32|*et al.^32*

You would think it should add a full point, and make the ‘et al.’ italic. OK, try it on this text:

> Jones et al 1996, Brown *et al* 1992, Harrison et al 1997

What you will get is this:

> Jones *et al.* 1996, Brown *et al* 1992, Harrison *et al.* 1997

Can you work out why that has missed the middle one altogether?

Answer: The original F&R (et al^32|*et al.^32*) has roman on the left and italic on the right, and that says “Find any ‘et al’ that is **not** italic and make it ‘*et al.*’ (with the full point added) in italic. So because the middle one was already italic, it ignored it.

You should have used:

> *et al^32|et al.^32*

The whole line is in blue, which means “Make the TEXT change and then colour it blue”; similarly, as the whole line is in italic, it means “Make the TEXT change and then make it italic”.

It doesn’t matter that ‘Brown *et al*’ was already italic; it does the text change anyway, adding the full point.

Now here’s the theory behind what’s going on there:
The problem we have with FRedit is that each side of the F&R (the F and the R) can only either be italic or roman, so that means there are only four possible combinations (roman/roman, italic/roman, roman/italic and italic/italic), and my program has to assign a meaning to each combination. Here they are, as applied to a silly example F&R:

| Option 1 – roman/roman
cat|dog
This means (obviously): Only change the text, don’t change the italic/roman status.

| Option 2 – *italic*/roman
cat|dog
This means: Find each **italic** ‘*cat*’ and make it a roman ‘dog’, i.e. remove the italic. But each **roman** ‘cat’ will remain untouched.

| Option 3 – roman/*italic*
cat|*dog*
This means: Find each **roman** ‘cat’ and make it an italic ‘*dog*’. So it will only find each **roman** ‘cat’, and will ignore any ‘*cat*’ that is already italic. This is not a terribly helpful option!

| Option 4 – *italic/italic*
cat|dog
This means: Find **any** ‘cat’ (roman or italic) and make it an italic ‘*dog*’, i.e. it finds **every single** ‘cat’ and makes them all into italic ‘*dog*’s.

5. Customising FRedit

At the beginning of the macro, there are a few things you could change, if you wanted to. Here’s the code:

```
doFinalBeep = True
showTime = False

' debugging = True
debugging = False

promptForSelectedText = True

funnyCode = "Blank"
' used as the code to mean "Leave the Find/Replace box blank"

isMacro = "DoMacro"
' used as the code to mean "Do the following macro"

caseCode = ChrW(172)
' This is the 'bent pipe' ('¬') character (horizontal line
' with bent end)
```

The `doFinalBeep` decides whether FRedit beeps at you to indicate that it has finished its work.

If you make `showTime = true` then FRedit tells you how long it took to do its job. (This is more of a tool for when I was developing FRedit.)

The debugging line is only of use if you’re using FRedit and get something strange due to a programming error. I can then try to solve the problem by getting you to change to `debugging = True`.

The next three are just the two ‘code words’ that FRedit uses ‘Blank’ and ‘DoMacro’ plus the special character used to mean ‘case insensitive’. You could change any of these, if you wanted to.

[THE END]

Well, that’s it. Have fun, and I hope that you find this macro useful.

Paul Beverley, paul@archivepub.co.uk
13 June 2019

~~~~~~

Answers to the question in 3.7:

color|colour

This changes all occurrences of 'color' to 'colour', but at the same time makes them bold and highlighted. But note that 'coloring' and 'colorful' will become '**colour**ing' and '**colour**ful'.

favor|favour

This changes all occurrences of 'favor' to 'favour', but at the same time makes them green type and underlined.

Beverley|^&

This doesn't actually **change** the word 'Beverley', but it makes it bold, and a larger font size.

~~~~~~

Appendix 1 – Understanding the font change mechanism in FRedit

To understand why and how FRedit does its font-changing activities, it's important *not* to think in the same way as you have to with manual F&R – they are very different. Here's why, using bold as the exemplar.

With manual F&R, if you click in the Find box and click Ctrl-B, it says, 'Font: Bold'. If you click again, it now says, 'Font: Not Bold'. Click a third time and you're back to where you started. So there are three 'states' for the 'boldness' of the Find; ditto for the Replace. This means that there are nine different combinations (well, eight if you ignore them both having no mention of 'Font: (Not) Bold'.

By contrast, with a FRedit item, each of the find and replace sides (left and right of the '|') can be *either* bold or not. So that's only four combinations – three, really, as 'not bold' with 'not bold' is, well, nothing to do with 'boldness'.

Below is a summary of the nine (eight) manual F&R options and what they do, plus the four (three) options for FRedit.

The important 'states', and how they compare, are highlighted

Find	**Replace**	**Change what?**	**Change text + ?**	
—	—	all occurrences	—	Nothing to do with bold
—	Bold	all occurrences	make it bold	
—	Not Bold	all occurrences	make it non-bold	
Bold	—	only bold items	no font change	
Bold	Bold	only bold items	make it non-bold	
Bold	Not Bold	only bold items	make it non-bold	
Not Bold	—	only non-bold items	no font change	
Not Bold	Bold	only non-bold items	make it bold	
Not Bold	Not Bold	only non-bold items	no font change	*A bit pointless*

Find	**Replace**	**Change what?**	**Change text + ?**	**Comment**
bold	**bold**	all occurrences	make it bold	
bold	plain	only bold items	make it non-bold	
plain	plain	all occurrences	no font change	Nothing to do with bold
plain	**bold**	only non-bold items	force to bold	*A bit pointless*

This will need some study and thinking, but it's the highlighted bits that are the significant ones. I hope this makes it less clear than mud!