

10 Main pre-editing tool – FRedit

N.B. The text below is just a brief description of the idea of *FRedit*. **Please do not try to use *FRedit* without reading the *FRedit* instruction manual.** The manual and some sample files are available from:

www.archivepub.co.uk/documents/FRedit.zip

Also, there are some (helpful?) videos:

FRedit for Beginners Part 0 (4:15) – Find FRedit and install it on your computer: 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) – Here's 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 (15:09) – Reinforcing the basic principles of FRedit: youtu.be/Mt4iuh6SOAM

FRedit revision 2 (5:50) – More about applying attributes, plus another gotcha: youtu.be/aQ_hR_K-INM

FRedit with notes and text boxes (4:58) – FRedit with foot/endnotes and text boxes: youtu.be/bGIMXppJIFM

FRedit and MultiSwitch (9:09) – Differences and similarities between the two macros: youtu.be/yGZHej6vaZ4

New way to run FRedit (5:41) – Run FRedit by selecting a FRedit list from a menu: youtu.be/1bVduGAFrhU

MiniFRedit (4:34) – Subset of FRedit's features, using a quick-and-easy method: youtu.be/DfAGD9RCpNQ

FReditListChecker – Check your FRedit list for some of the 'obvious' errors: youtu.be/Z7cjf446JWM

FRedit – Stretch yourself (31:35) – A look at one of my fairly advanced FRedit lists: <https://youtu.be/doMAH8b94fQ>

Text manipulation example (12.24) – Using FRedit and other tricks to manipulate a problem text:

youtu.be/FTdyBwm4AzY

US UK spelling conversion (12:23) – Using FRedit to speed up US/UK spelling conversion: youtu.be/XkgTTNcmI2I

Alternatively, try *FReditSimple*, below (also *MiniFRedit*).

When I very first started freelance editing, I noticed that each time I opened a new file, ready to edit it, I was doing a number of global find and replaces – double space to single, spaced hyphen to spaced en dash, 'et al' to 'et al.' and so on. I thought to myself, 'Wouldn't it be good if I could create a list of these changes, in a Word file and then run the F&Rs automatically? It would save me typing out the same F&Rs over and over again.' And I realised too that this would be especially useful with a multfile job where I would want to do exactly the same set of F&Rs on each file, and wouldn't want to forget any of them.

A friend who was then 'into' macros said that he could write something that would do just that, and '*FRedit*' was born – **F**ind and **R**eplace **e**dit'.

Then, as I too have learnt how to program macros, *FRedit*'s facilities have got more and more sophisticated, and now I simply couldn't imagine doing an editing job without it. It saves me time, and increases the consistency of my edited output.

Importantly, you can use *FRedit* with Word's wildcard searching facility, and the fact that you don't have to (remember or) type out those arcane strings of symbols (e.g. Find ([0-9]) ^= ([0-9]) and Replace \1^=\2) is a real boon. Where it becomes *really* powerful is that you can do a *series* of linked wildcard F&Rs. But don't panic: you don't have to work these things out for yourself. Other people have done the hard work – you just copy their F&Rs and paste them into your own lists of F&Rs and use them.

You may also be worried that by doing global F&Rs, you might change things that you didn't intend to. **You are absolutely right to worry!** *FRedit* is an extremely dangerous tool if used ill-advisedly, so don't try to do too much with it too soon. As with any dangerous tool – a sharp cooking knife or a circular saw – your skill and confidence builds the more you use it. But, in the same way that many tools have safety features that you can use (or choose not to use as your skill grows), so does *FRedit*. You can use tracked changes (but you may know that it too has its

dangers!) and/or you can choose to add a coloured highlight (any colour you want) to any or all of the changes that *FRedit* makes – so at least you can see what it has changed.

The *FRedit* files, including the instruction manual, are downloadable from my website at (www.archivepub.co.uk/documents/FRedit.zip). These include full instructions as to how to install the *FRedit* macro and how to use it. There's also a library of ideas of things that you can do with *FRedit* plus a section containing hints and tips that will help you to build up your efficiency in using *FRedit*.

The *FRedit* macro listing is *only* given here to make it easier if you want to update to the latest version. **Please do not try to use *FRedit* unless and until you have read the instruction manual** – you wouldn't buy a chainsaw and start to use it without at least reading the instructions, would you?!

N.B. *FRedit* now works on either the whole text, or just the selected text

Sub FRedit()

FRedit's younger brother – MarkIt

You can use FRedit to just mark some items with highlighting or font colouring, but here's a quick and easy way to do it.

You create a list of words, phrases and/or wildcard finds, and onto each, you add an attribute or attributes. Here's a sample list to give you a flavour:

| Find vocative comma errors (use wildcards)

```
[!,] name[.,\?\!]'
[!,] my dear[.,\?\!]'
[!,] father[.,\?\!]'
[!,] mother[.,\?\!]'
```

| Find forbidden dialogue tags. May need to amend for style, and for tense.

```
shrugged, "
smiled, "
nodded, "
shrugged, '
smiled, '
nodded, '
```

| Find speech at end of a para that is not closed with a full point

| Need to use highlight,
| but you can't show highlights of the forum

```
"^p
'^p
```

| Find paras closed by a comma instead of a full point.

```
^p
```

| Find commas followed by a cap (not I)

```
,<[A-Z][a-z]
```

| Find speech that does not close with punctuation (note trailing space)

```
[a-z]"
[a-z]'
```

| Find paras closed without a full point. Search for body text style, wildcards on.

```
[a-z]^13
```

For any of the items of text (or any WC find) in the MarkIt list that has one or more of the following attributes applied, then over in your target document, that attribute(s) will be applied to the text:

- highlight (any colour)
- font colour (any colour)
- underline
- bold
- italic

But any line in the MarkIt list that doesn't have at least one of those attributes will be totally ignored.

Alternatively, if you have a line that you *don't* want to be used, e.g. it's just a comment for your own benefit then you can:

- start the line with a vertical bar '|'
| This is a comment line

- apply a strikethrough
~~This line is ignored~~

The latter is useful because you might decide, one day, that you don't want to mark certain of the items in the list; if so, just strike them through. Then you can remove the strike-through, or just close the file and don't save the changes.

One application for MarkIt would be to use SpellAlyse to create a spelling errors list, then go through it and add highlight or colour to any items that you want marked, then run MarkIt – any unmarked words in the spelling errors list will simply be ignored by MarkIt.

But remember that if you want to mark a word like 'etc', you will need to take care, else you'll get:

'Fetch some etchings and sketches and a bottle of ketchup!

(Be careful what you wish for!)

Rather use the item:

<etc>

which will be activated by MarkIt as a wildcard find.

MarkIt, like FRedit, is case sensitive, but if you want an item to find *any* case, then add a 'bent pipe' in front:

-word

The item above will give you: My word! Microsoft Word is useful.

Sub MarkIt ()