Au revoir, CR

Unwanted carriage returns can be a real pain to handle. Tim Phillips takes you through the ins and outs of successful replacement, and goes on to tackle spacial equality and smart quotes, and offer some neat macros.

Having received a good deal of correspondence following my advice to a reader on finding and replacing carriage returns (CRs), I realise that it would be of benefit to cover the subject in more depth.

A standard problem of imported, pasted text is: you get a carriage return at the end of each line so it's a pain to edit and format. You want to keep the paragraphs but strip out the hard CRs, so it's a job that lends itself well to a simple macro.

The following method works for the text you find on bulletin boards, text copied from the Internet, news feeds, or for "repurposing" old ASCII text, and it saves you hours of work manually deleting the carriage returns.

First, you prepare your text by inserting a blank line between each paragraph. Then you search for and replace all the double carriage returns with a set of symbols not normally occurring in the document (in these macros I've used &@#). Then you find all the remaining (single) CRs and replace them with nothing. Finally, you find all the occurrences of the symbols and replace them with carriage returns to restore your paragraphs — it's a standard trick, and I claim no originality here. I have provided examples of the various commands *[next column]*.

There are caveats, however: it wrecks tables, so cut and paste them into another document first; a good practice is to save under another filename before carrying out a search and replace, but Ami Pro 3.0 won't do this (the forthcoming Word Pro will) — it will only search for one special character at a time, so you can't search for two CRs and therefore need to use a different trick. Thanks to reader John Cowin (who mailed me first), and others.

- WinWord version:

```
Sub MAIN
EditReplace .Find = “^p^p”, .Replace =
“&@#”, .ReplaceAll, .Wrap = 1
EditReplace .Find = “^p”, .Replace =
“^s”, .ReplaceAll, .Wrap = 1
EditReplace .Find = “&@#”, .Replace =
“^p”, .ReplaceAll, .Wrap = 1
End Sub
```

- WordPerfect 6.1 version:

```
Application (A1; “WordPerfect”;
Default; “UK”)
SearchString (“”)
ReplaceString (“&@#”)
ReplaceForward (Extended!)
PosDocTop ()
SearchString (“”)
ReplaceString (“”)
ReplaceForward (Extended!)
PosDocTop ()
SearchString (“&@#”)
ReplaceString (“”)
ReplaceForward (Extended!)
```

Finally, Ami Pro. You should replace every carriage return first, then search on the double occurrences of &@#. Next, replace those with a return, then delete all the single &@#s.

This is for a macro called crstrip.smm:

```
FUNCTION CRSTRIP1()
Replace(0 0 1024 “¶”  “&@#”   )
Replace(0 0 1024 “&@#&@#”  “¶”   )
Replace(0 0 1024 “&@#”  “”   )
END FUNCTION
```

For Wordstar for DOS users, there's also a freeware executable which strips CRs (but it's a 33Kb .EXE standalone file, so I can't include it here) available from the Wordstar conference on CompuServe. As it is freeware, I can pass it on to any readers who want it — but only by email.

Equal spacing

John Cowin has a useful technique for equalising the spacing after a full stop:

"As a documentation consultant, I am frequently faced with problems when integrating text from multiple contributors. Invariably, the contributor has decamped to another continent by the time I am ready to hurl the disk back at him. When this happens, I use similar techniques. For example, to replace all single and treble sentence spacing with double:

1. Search on .<space><space> and replace with $%^&.
2. Search on $%^&<space> and replace with $%^&. Repeat this step to ensure there were no strings of four or more spaces (yes, it does happen).
3. Search on .<space> and replace with $%^& (or .<space><space> — it doesn't matter at this stage).
4. Search on $%^&, and replace with .<space><space> to return the text to its intended appearance.

There must be other uses for this approach. Basically, once you have protected or masked what you want to keep, you can wreak havoc around variations and still return to sensible formatting."

Text search tip

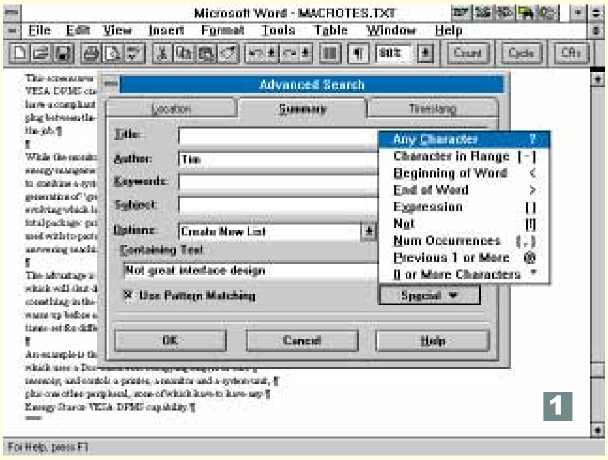
Can you carry out full text searching in Word? Too right you can. It can search across directories for any text string in any word file.

To find this magic function, select File, Find File and click on Search. This brings up a horribly designed dialogue box. The Location section allows you to set the directories you want to search and includes a check box to allow you to search subdirectories. Timestamp allows you to set a date filter, or specify the Author name. The Summary section has a little box at the bottom called Containing Text. Enter the text you want to find, including wildcards if you check the Pattern Matching box (wildcards are in a pull-down menu on the right). Searching isn't fast, but you get back a list of the documents with their place in the directory tree and a preview screen.

Why Microsoft buried the ability to search for text in a document four levels down the menu system defeats me. If anyone can supply a quick text search macro that throws up a dialogue box with the text to search for, in a named directory, I will be extremely grateful. One point to note is that the dialogue box settings are persistent, which isn't obvious, so take care to remove any unwanted author names or date filters from the dialogue before you complete the next search.

Wordstar 7.0 patch

If you feel that you were born to hack code, then try this if you use Wordstar 7.0: I found this patch while optimising a friend's Wordstar installation. The problem is that when you have a mouse driver enabled, the page review function is glacially slow. *Fig 1 (page 300)* shows the solution, as recommended by Wordstar technical support. I can't stress too strongly the importance of making backups of both these files before you try to hack them around.



Title:

Author: Tim

Keywords:

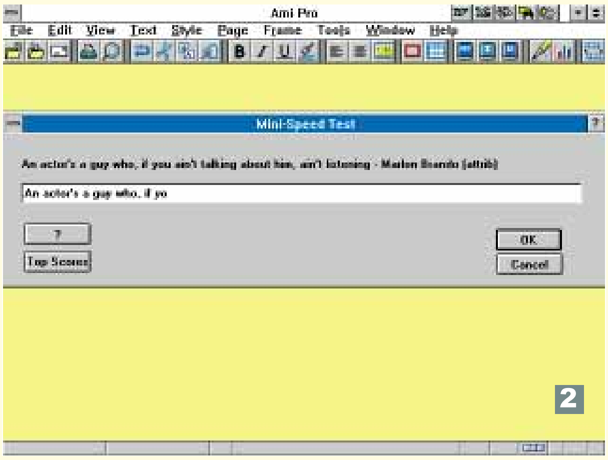
Subject:

Options: Create New List

Containing Text

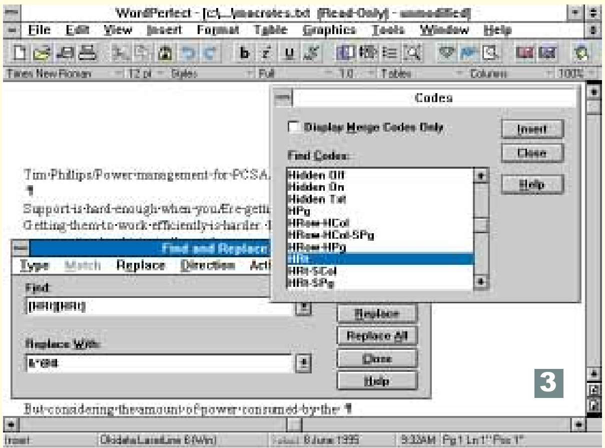
Not great interface design

☒ Use Pattern Matching



An actor's a guy who, if you ain't talking about him, ain't listening - Marlon Brando (attrib)

An actor's a guy who, if yo


WordPerfect - [c:\...\macrotes.txt (Read-Only) - unmodified]
File Edit View Insert Format Table Graphics Tools Window Help
Times New Roman 12 pt Styles Full 1.0 Tables Columns 100%
Codes
Insert
Close
Help
Find and Replace
Type Match Replace Direction Acti
Replace
Replace All
Close
Help
Insert OkidataLaserLine 6(Win) 8 June 1995 9:33AM Pg 1 Ln 1" Pos 1"
3

Tim·Philips/Power·management·for·PCSA.
¶
Support·is·hard·enough·when·you'Ere·getti
Getting·them·to·work·efficiently·is·harder
But·considering·the·amount·of·power·consumed·by·the ¶

☐ Display Merge Codes Only

Find Codes:

Hidden Off
Hidden On
Hidden Txt
HPg
HRow-HCol
HRow-HCol-SPg
HRow-HPg
HRt
HRt-SCol
HRt-SPg

Find: [HRt][HRt]

Replace With: [*@#]

1 *Searching, using free text in Word for Windows*

2 *The typing tester in Ami Pro — by email only, I'm afraid*

3 *Stripping carriage returns in WP*

Neat macros

Undaunted by our fix for his dayfinder macro — remember that we eventually distinguished between the Julian and Gregorian calendars (*PCW,* May '95)? — Graham Brown leaps back into the fray with some neat speed test macros. Type your selection and the macro returns your speed and accuracy, and if your accuracy

Fig 1 The Wordstar 7.0 “solution”

You need to patch two files: WS.EXE and PREVIEW.OVR

● Firstly, the WS.EXE file:

```
What you see                What you type
C:\WS>                              REN  WS.EXE  WS <Enter>
C:\WS>                              DEBUG  WS <Enter>
-                           E A0D2 <Enter>
xxxx:A0D2 00.               21 <Enter>
-                           W <Enter>
Writing xxxxx bytes
-                           Q <Enter>
C:\WS>                              REN  WS  WS.EXE <Enter>
C:\WS>
```

● Now, the PREVIEW.OVR file:

```
What you see                What you type
C:\WS>                              DEBUG PREVIEW.OVR <Enter>
-                           E AC36 <Enter>
xxxx:AC36 CD.                       EB <SpaceBar>
33.                         08 <Enter>
-                           E ACD8 <Enter>
xxxx:ACD8 CD.                       90 <SpaceBar>
33.                         90 <Enter>
-                           W <Enter>
Writing xxxxx bytes
-                           Q <Enter>
C:\WS>
```

(Note: with debug you might see a “D” instead of a “-”.)

is 100 percent, you might qualify for the hall of fame. Meanwhile, if you want to delete the high score table, just type c, then Enter.

He has supplied Ami Pro and Word versions. More would be welcome. The catch is that they are rather long macros; too long to reproduce here. But if anyone wants to email me I’ll pass them on (via email only, I’m afraid).

Mr Brown also supplied an ingenious anagram master based on Word’s ToolsGetSpelling command: “No programming involved, which, judging from my calendar program, is just as well,” he adds modestly. Again, if you email me with a request for the macro, I’ll pass it on.

See *Fig 2* for this neat macro. Quick and easy to use, its only limitation is that it can only do one-word anagrams, so I’m confident that you won’t be able to tell me what NERDS POWER PLUM CAR TOOL spells...

Smart quote solution

On the topic of WinWord AutoCorrect, Shane Devenshire of Walnut Creek, California, contacted me with a problem in replacing smart quotes. To be honest, he solved it himself, so I can’t claim the credit for this one: “As you type, Word changes an ASCII apostrophe to a ‘smart apostrophe’. It does this in the document but not in the AutoCorrect dialogue box,” he says. The problem comes when you try to AutoCorrect any word with an apostrophe in it, with smart quotes enabled (there’s a check box in the AutoCorrect dialogue box); you can’t, because AutoCorrect sees a smart quote.

The answer, as Mr Devonshire discovered, was to “force paste” the word with smart quotes, into AutoCorrect. First, type the word erroneously, then on the next line correctly, like this:

TIm's (with an ASCII quote)
Tim’s (with a smart quote)

Highlight the incorrect one and copy it to the clipboard. Then highlight the correct one. Open the AutoCorrect dialogue by selecting Tools, AutoCorrect. You’ll see the correct version in its place but the radio button above it will have Formatted Text enabled. Click on Plain Text instead. Then click in the Replace: box and manually paste the incorrect version from the clipboard by typing Ctrl+V. Note that you can’t paste from the Edit menu in this situation. You’ll now have an AutoCorrect entry that recognises words containing smart quotes.

Export problem

David Hurren emailed me to ask how to get WordPerfect 5.1 tables into a database. He

Fig 2 An anagram master based on Word’s Tools GetSpelling command

```
Sub main
Dim x$(60)
st:
Redim x$(60)
Begin Dialog UserDialog 344, 104, “Anagram Master”
        Text 19, 8, 168, 18, “Please Enter Anagram”, .Text1
        TextBox 17, 29, 310, 21, .TextBox1, 1
        PushButton 240, 60, 87, 15, “Solve”, .Push1
        CancelButton 240, 79, 88, 16
End Dialog
Dim dlg As UserDialog
C = Dialog dlg
If C = 0 Then Goto fin
a$ = dlg.TextBox1
g = InStr(a$, “ ”)
On Error Goto err
If g > 0 Then MsgBox(“Please enter your anagram with no spaces”)
If g > 0 Then Goto st
h = InStr(a$, “?”)
hh = InStr(a$, “*”)
n = 2
If h > 0 Or hh > 0 Then n = 1
ToolsGetSpelling x$(), a$, “”, “”, n
Begin Dialog UserDialog 301, 238, “Anagram Master”
        Text 26, 25, 97, 13, “Words found”, .Text1
        ListBox 26, 46, 253, 122, x$(), .MyListBox
        OKButton 190, 180, 88, 21
        CancelButton 190, 204, 88, 21
End Dialog
Dim dlg As UserDialog
N = Dialog(dlg)
If N = 0 Then Goto fin
Goto st
err:
MsgBox(“Please enter only letters or ? and *”)
Goto st
fin:
End Sub
```

receives a bunch of reports every month in this format and wants to export the data in them to Lotus Approach under Windows.

Provided you have a copy of WordPerfect 5.1 under which to open your file, it’s not a problem. This is one of the times when WordPerfect’s Reveal Codes feature is extremely useful. Call up the Reveal Codes function. At the start of the table, there’s a code called TableDef. Just delete this code and WordPerfect has no idea that the data ever was a table, and it puts tab stops in there instead. You now need to put quotes around each item of data, so search and replace tabs with “,” (quote, comma, quote). Search and replace returns with “[HRt]” (that’s quote, return, quote) and add a quote at the beginning of the first record. Cut out the table text and save it in its own document as a text file. This can then be imported into any database as a comma-delimited ASCII file.

Breaking up is hard to do

We end this month’s column with a rare Wordstar 5.5 query: Bill Bentley of Belton — try saying that after a night in the pub — reports that when he tries to break up a document into small chunks of text, they all save as plain text without formatting.

I’ve found a list of bugs in Wordstar 5.5 and my guess is that you’re breaking up the document by selecting a block and saving it. This doesn’t save the block style information. Instead, open a new document, copy the block into it and save it. It takes longer but you keep your text style, and as Wordstar supports multiple documents, you don’t have to close the original.

PCW Contacts

And that’s that for this month. Surface or airmail to *PCW*, otherwise I’m on email at **wong@cix.compulink.co.uk** and **CompuServe 100436,3616**