Mountain of **macros**

Tim Phillips didn't have to go there — the mountain came to him. A mass of macros for your delight, plus tips on spell-checking and accents.

There are several macros this month, so I'm limited for space to tell you about what is going on in the industry. One major development, though, is the distressing spread of the "macro virus" to Word Pro (formerly Ami Pro).

According to Andy Campbell, managing director of anti-virus software supplier Reflex Magnetics, the recently-discovered Green Stripe virus is the first to affect documents produced on any other word processor except Word.

To recap: a macro virus attaches itself to a word processor document as a macro stored with the document template. When it's opened, it attaches the macro to other documents it finds. What the macro actually does when it is opened, is the payload, which can potentially be quite destructive. The problem is that documents are commonly shared across networks.

Green Stripe is a classic example of this for Word Pro users. The price you pay is fairly innocuous — it changes all the "its" in your documents into "it's". Reflex picked up the source code in an underground magazine: no, we aren't going to tell you which one. You can spot it because you will find macros called InfectFile.smm (not very subtle, that) attached to your document — and because your files keep being opened and changed.

Reflex's anti-virus software, DiskNet, has a novel solution to the macro virus problem — it searches for auto-executing macros and warns you. Phone Reflex for more information (look at the *PCW* Contacts box for details).

If it's any consolation, at the rate Word for Windows sells these days, the Word Pro virus is limited by market forces — not, perhaps, a comforting thought for Word Pro users.

Internet assistants

Over the last few months I have had a few replies to my search for a good HTML editor. This is becoming increasingly important, and equally difficult to find. Many of you recommended HotDog, which I found far too complex for my tiny mind. My main horror is still the multiple searching and replacing needed for upkeep of a word processor document in HTML format.

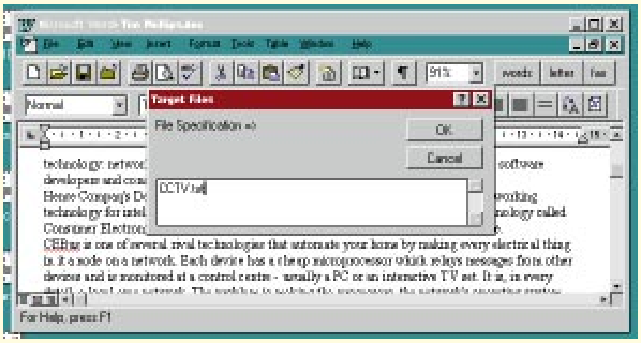
davcefai (*sic*), from sunworld.mt, offers a long but glorious search-and-replace macro by email (see *Fig 1*). The macro searches across directories and subdirectories. It has several uses, but I include it here because it is excellent for searching and replacing links across a collection of Web pages, so if you are struggling to maintain a whole set of documents, this is your macro. It's ideal for contracts and reports.

"The attached file, MACRO.TXT, is a Word 6.0 macro to perform a find and replace on an entire directory and its subdirectories," Mr davcefai explains. *"Basically this macro first reads the files into a file named PDIR.OUT, and then loads each of these files in turn in order to perform the find and replace operation… When you run the macro you are first asked for the filespec of the files to change. You are then asked for the top level directory from which to start searching."*

Now please mail me, Mr davcefai, to claim your prize, as I don't know where you live.

A similar shareware program can be found in ftp.demon.co.ukpub/ibmpc/win3/apps/wchange.

Called WinChange, it will handle GRP, INI, PIF and HTM files and works over a network. Thanks to David Agbamu, of Manchester, for the upload.



File Specification => CCTV.txt

CEBus is one of several rival technologies that automate your home by making every electrical thing in it a node on a network. Each device has a cheap microprocessor which relays messages from other devices and is monitored at a control centre - usually a PC or an interactive TV set. It is, in every

The global search-and-replace macro in action

Tim's Macro Club

Don't be shy of telling me if you haven't been rewarded for your macros yet and I declared you a winner in previous months. "A fair day's work for a fair day's pay" is what I say, and some of your efforts imply more than a day's work.

Please be aware that I can rarely print long macros because of space limitations. On the other hand, I've already included a long macro this month, because it's a good one. You'll just have to take care copying it out.

If you are online, you can mail me to

Fig 1 Search-and-replace macro

```
'----------------------------------------------------------------
'         Declare Global Vars. Filespec$ is target files, path$ is the
'         target directory, tcnt is running total of files found
'----------------------------------------------------------------
Dim Shared filespec$, path$, tcnt
Sub MAIN
         filespec$ = InputBox$("File Specification => ", "Target Files")
         Search                                    ' Create a File List
         If tcnt < 1 Then
                  x = MsgBox("Warning: No files found", "Warning", 0)
                  Stop
         End If
'-----------------------------------
' Get the FIND and REPLACE strings
'-----------------------------------
         is$ = InputBox$("String to replace >", "Replacing")
         willbe$ = InputBox$("New String >", "Replacing")
'---------------------------------------------------
' Confirm operation as an error would be disastrous
'---------------------------------------------------
         x = MsgBox("Replacing " + is$ + " by " + willbe$ + " in " +
filespec$ + " (" + Str$(tcnt) + " files)", "Confirm", 1)
         If x <> - 1 Then Stop                   ' User pressed CANCEL
'-------------------------------------------------------------------
' Now we open the text file with the list of files to operate on
' The first line of this file is the directory name so we read this
' into PTH$. The next lines are filenames
'-------------------------------------------------------------------
         Open "pdir.out" For Input As 1
         Input #1, pth$
'--------------------------------------------
' Sequentially open the files and do the job<None>
'--------------------------------------------
         While Not Eof(1)
                  Input #1, fname$
                  fto$ = fname$
                  FileOpen .Name = fto$
                  Doit(is$, willbe$)                        ' Call Subroutine
                  FileSave
                  FileClose
         Wend
End Sub
'=======================================
' Subroutine to do the Find and Replace
'=======================================
Sub DoIt(is$, willbe$)
         EditReplace .Find = is$, .Replace = willbe$, .Direction = 0,
.MatchCase = 0, .WholeWord = 0, .PatternMatch = 0, .SoundsLike =
0, .ReplaceAll, .Format = 0, .Wrap = 1
End Sub
' =================================================
' These are the routines that create the file list in PDIR.OUT
' =================================================
```

CONTINUES OVER

request copies of macros if you can't get one to work. I might take a couple of days to reply to your messages though, as I'm deluged by your emails — the information superhighway has left me lying in a hole in the ground covered in tyre-marks.

My plea for macros that can use a counter, such as an invoice counter, prompted many interesting and well thought out macros. But all were for Word; so any other contributors with experience on other platforms will, as ever, win by default.

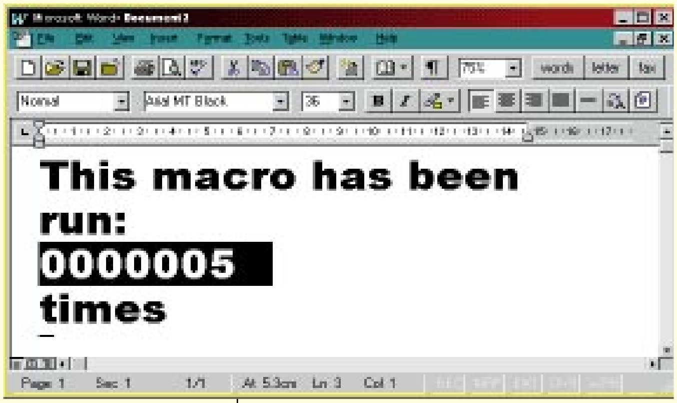
Fig 1 (continued)

```
'========================================
' Search the directory for matching files
'========================================
Sub Search
   path$ = InputBox$("Enter Directory > ")          ' Get Directory
   If Right$(path$, 1) <> "\" Then path$ = path$ + "\"
   Open "pdir.out" For Output As 1                  ' This is the file list
         Print #1, path$                    ' Write path to file
   listfiles(path$)              ' Call recursive subroutine
'        listdirs(path$)
   Close 1
' Close Output File
End Sub
'================================================
' Recursive Subroutine to list the files in the directory
'================================================
Sub listfiles(p$)
   Dim cnt
         Dim f$(100)             ' Wordbasic does not have a REDIM
PRESERVE
                       ' Statement, hence the arbitrary number
         Dim i, fname$
   fname$ = Files$(p$ + filespec$)' Get first filename.
'--------------------------------------------
' Read all the files found into the array f$()
'--------------------------------------------
         While fname$ <> ""
                       f$(cnt) = LCase$(fname$)
      cnt = cnt + 1                           ' Increment counter.
      fname$ = Files$()
   Wend
'-------------------------------
' Write the filenames to the file
'-------------------------------
         tcnt = tcnt + cnt       ' running total of found files
         cnt = cnt - 1
         For j = 0 To cnt
                       Print #1, f$(j)
         Next
         listdirs(p$)
End Sub
'---------------------------------------------------------------
   Get subdirectories in a directory. You need to first
get a count, then read each one. Don't blame me, there
is no DIR$ function in WordBasic!
'---------------------------------------------------------------
Sub Listdirs(p$)
         x = CountDirectories(p$)
         For i = 1 To x
                       d$ = GetDirectory$(p$, x)
                       dpath$ = p$ + d$ + "\"
                       listfiles(dpath$)        ' get the files
         Next
End Sub
```

● Most contributors used the solution outlined by Mark Heaton on the republic of Microsoft Network: *"By far the easiest way to tackle the problem is to store the last used invoice number in the INI file (or Registry) to do this function:*

`GetPrivateProfile$()`
and
`SetPrivateProfile$()`
are used. The AutoNew macro is called each time the template is opened, reads in the number of the last invoice and increments it.

The only task still remaining is to store the new number in the INI file. To do this, use the AutoClose macro and write the number only if it is greater than the existing number (this allows the invoice to be reopened at a later date without having to destroy the last number)."



This macro has been run:
0000005
times

Microsoft Word - Document2
File Edit View Insert Format Tools Table Window Help
Normal
Arial MT Black
36
Page 1 Sec 1 1/1 At 5.3cm Ln 3 Col 1

The kludge macro for generating invoice numbers. It simply inserts the contents of a file at the insertion point. Not pretty, but quick

● Vital Debroey and Jacques Thoorens both mailed macros from Belgium, solving the problem in this fashion. M Debroey's macro code and explanation is shown in *Fig 2* (it was shorter).

Fig 2 Debroey's macro

1. In the template, insert a bookmark at the position where the invoice number will be inserted. Give it a name, say "InvoiceNumberGoesHere".
2. Add the following AutoNew macro to the template, or insert the lines between "Sub MAIN" and "End Sub" into the existing AutoNew macro:

```
Sub MAIN
  INIFile$ = "WINWORD6.INI"
  Section$ = "My Private Settings"
  ProfileString$ = "NextInvoice"
  BookmarkName$ = "InvoiceNumberGoesHere" 'this bookmark MUST exist!
  'get next invoice number (as a string) from the INI file
  NI$ = GetPrivateProfileString$(Section$, ProfileString$, INIFile$)
  'if nothing found, assume "1" as next number
  if NI$ = "" then NI$ = "1"
  'insert at the appropriate bookmark
  EditGoTo .Destination = BookmarkName$
  Insert NI$
 'save the next invoice number in the INI file
  NI$ = LTrim$(Str$(Val(NI$) + 1))
  SetPrivateProfileString Section$, ProfileString$, NI$, INIFile$
End Sub
```

Hints & Tips

Spell checking
A Mr or Ms aspiers (that's an email name) added to my discussion of spell checking in the last issue.

"A useful feature when using Microsoft applications is the spell checker. This can highlight unknown words and add them to a custom dictionary. This is normally called 'CUSTOM.DIC'. A problem can occur as each application can create its own CUSTOM.DIC, meaning that you have to teach each application your own variety of unusual words. The way around this is to point each application to the same dictionary. Edit WIN.INI using SYSEDIT, then amend the entries for the dictionaries under [MS Proofing Tools] to point to the same file."

Accent on WordPerfect
Mr P Kahrel saves the day for WordPerfect users with a routine to get accented characters by typing two-letter combinations as discussed in the hints section a couple of months ago: "You said that you felt a macro coming on to enter accented characters, and you were right. So here you are. Chris Collins' accent key is easily implemented in WordPerfect. Here is the macro:

```
CHAR( Accent ; "Accent: " )
Accent := NTOC( Accent )
CHAR( Letter ; "Letter: " + Accent )
Letter := NTOC( Letter )
```

```
SWITCH( Accent+Letter )
  CASEOF "ga" : Type( "" )
  CASEOF "ge" : Type( "" )
  CASEOF "gi" : Type( "" )
  CASEOF "go" : Type( "" )
  CASEOF "gu" : Type( "" )
  CASEOF "gy" : Type( "" )
  CASEOF "aa" : Type( "" )
//more combinations...
ENDSWITCH
```

"In the commands Type(" ") the relevant WP characters need to be inserted. To enter the a-grave in WP, press Ctrl+A and type `a in the Compose window. The e-grave; Ctrl+A `e.

"To assign this macro to a hot-key, let's say the Insert key is in the separate coloured key island, save the macro in the WP macro directory (as, say, ACCENT.WPM) and edit the keyboard so that the Insert key calls the ACCENT macro. Shift+F1, 4, 3 brings you to the Keyboard edit screen. There choose Create, press the Insert key in the Key field (WP inserts "en Ins" for enhanced Insert key), then choose 5 and press Enter to create the macro.

"In the macro edit screen type CHAIN("ACCENT.WPM") and press F7 until you are back at the edit screen. To try the macro: press the Insert key and type ga. If all is well, the a-grave should be printed on the screen. (You can also enter the whole macro in the macro edit screen, rather than calling it from disk, but the method described here makes it easier to change the macro.)"

Note that he uses a section called "My Private Settings" in WINWORD6.INI, which will be created the first time the macro runs. You can use this macro technique for all kinds of settings that need to be inserted into new documents.

Fig 3 Using a dummy file

```
Sub MAIN
f$="c:\invoice.num"
a=1
On Error Resume Next
Open f$ For Input As #1
Read #1, a
        Close
        a$=Str$(a)
a$=String$(8-Lan(a$),
"0")+Right$(a$, (Len(a$) - 1)
Insert a$
a=a+1
Open f$ For Output As #1
        Write #1, a
Close
End Sub
```

● A less elegant but nevertheless effective solution, from Rob West in Cheshire, uses a dummy file called "invoice.num" in the root directory. (See *Fig 3*.)

Jason Whyte agrees with this approach and suggests a refinement:

"It's not the most elegant way of doing things, but it does work. Perhaps a better, and much more useful way of doing things would be to have a file with a table of numbers, filenames and other useful information on the document. When creating a new invoice, Phil could have a dialogue to get the desired filename and info, then read the control file to get the last number and add a new row for the new invoice. That way, he not only numbers each one automatically, but builds an index document for them."

Comments on both of the above, please — and maybe you could look at problem number two, below.

● This month, I have a couple of simpler problems, suggested by Alison Walley at the American University of Beirut:

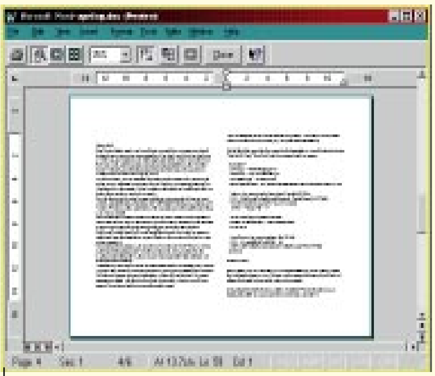
1. *"Converting the 'new document' button to bring up a list of templates, instead of one based on the normal template. In Word 2, which I still use on my trusty Dell 486P/50, I reconfigured the standard button as FileNew instead of FileNew-Default. Is there any way of doing the equivalent in Word 6, because I can't seem to find one?*

"My solution up to now has been to design a new button altogether, which seems a bit of an effort. Also, why does the Word 7 Ctrl+N command always bring up a blank document? It should bring up the list of templates.

2. *"I want to print A5 brochures. It's possible to get these the right size by setting page orientation to landscape and putting the document into two columns. The problem is the page numbers.*

"WordPerfect v6 (for DOS anyway, and I suppose for Windows) had an option to print brochures like this which actually sorted out the pages in the right order for printing. I rather hoped Microsoft would put something like it in Word 7.

"It could be done with a counter of some sort set at the left and right margins of each page, but how do I get such a counter? I see that someone else (in your February issue) asked about counters for a Word 6 macro, so hopefully someone will produce some ideas for this."



Printing an A5 booklet using landscape orientation and two columns. Any improvements on this for a more professional job?

PCW Contacts

Contact **Tim Phillips** by surface or airmail to **PCW**, otherwise email him at **wong@cix.compulink.co.uk** and **CompuServe 100436,3616**

Reflex 0171 372 6666