



A refined accent

Tim Nott discovers an accented character shareware utility to alleviate your acute problems without sending you to an early grave. And there's proofing in tongues, plus a neat tiling fix.

We've had a lot of correspondence about inserting accents with various word processors but, until now, haven't found a satisfactory solution that works in any application. You can, of course, memorise the ANSI codes, and holding down the Alt key, type "0233" on the numeric keypad for an é, for instance. The problem with this, I find, is memory. Not the computer's, but mine. I'm fine on the lower-case "e" permutations but with other letters have to resort to the Windows Character Map, and the problem with this is that it suffers from poor design.

It includes all the characters that can be typed normally from the keyboard and the display is too small to distinguish the different accents easily. You have to use the magnification cursor to track them down. Having done this, it's another three clicks to get the wretched thing onto the clipboard. So, let's have a big hand for Paul Herber of High Wycombe, with his Diacrit. This is a tiny shareware utility that allows single-click copying of an accented character to the clipboard, from a set of keys that vary with the chosen language.

You can run multiple instances of Diacrit, to have several languages available at once and make it stay "always on top". It comes with a variety of preset languages and useful symbol sets, like fractions and currencies, and as these are all contained in a simple .INI file, you can easily customise your own sets.

So far, I make this: Diacrit 4, Character Map 0. It's a Windows 3.1 application but will work happily under Windows 95. The bad news is that it is difficult (but not impossible) to access the extended Windows 95 Greek, Cyrillic and Central



Diacrit: Easier accents in any application

European accents, or characters from Symbol or other special fonts. If all you want to do is insert ANSI characters in the current font, anywhere in Windows, it's convenient and well worth the £10 registration. You should be able to find it on our PCW cover disc (Diacrit.zip — 22.1Kb), or download it from Paul's web site at www.pherber.com/share/diacrit.html which also has some riveting links for fans of The Archers.

On a similar subject, David Bivar wrote to me about the DOS word processor, Nota Bene. This is, he says, "An excellent and flexible DOS program which enables the typing of Greek, Russian, Hebrew and all European and Orientalist accents and diacritics: indispensable for academic work and international correspondence."

I must confess that I'd never heard of this before and haven't yet been able to get hold of a review copy. When I do I'll keep you posted, but if you want to find out more, there's a web site at www.notabene.com/index.html.

Duke of URL

If you type a URL or email address (or anything it thinks might be one), Word 97 automatically formats this as a link, with default blue underlined text, and also turns it into a link: move the pointer over it and it will

change to a finger. Click on it, and it will launch your web browser or email client.

Wonderful though this technology is, it has the frustrating side effect that if you mis-type the URL or address, you can't click back on it to correct it, as this just launches the (wrong) link. Even if you creep up on it from behind, with the arrow keys, editing the text won't edit the link.

The solution is to right-click the link, choose Hyperlink, Edit Hyperlink and then correct the link. This doesn't correct the visible text, though, so you'll still have to creep in with the cursor keys to edit that — it's probably quicker to delete the thing and start again. If you want to turn this feature off or on, don't go searching in "Tools/Options" because it isn't there. Instead, go to the "AutoFormat" and "AutoFormat as you type" tabs of "Tools/AutoCorrect". You can also get there from "Format/AutoFormat/Options" if you don't mind one more click.

Per-version

If you're running various versions of Word in the same organisation and want everyone to be able to read each other's files, you have two choices. First, you can save Word 97 files in Word 6/7 format but this, as I noted last month, is rather misleading as it

p278 ➤

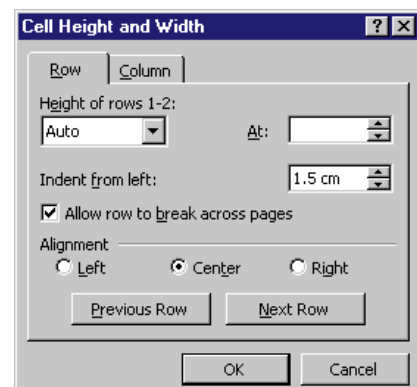
actually saves them as RTF and may result in some loss of formatting. I haven't found this myself, although this column is always sent to the PCW editorial office as RTF (but then, I don't use a lot of formatting).

RTF files also have the advantage of being smaller than native Word 97. If you want to make this the default save format rather than trawl through the "Save as Type" options every time, then "Tools/Options/Save" has the facility to do just this. Microsoft, apparently, has promised a "real" version 6/7 "Save As..." filter for later this year, but don't hold your breath.

The other approach is to tackle the problem from the Word 6/7 end, as there is a convertor available from the Microsoft web site for reading Word 97 files into versions 6 or 7. I've got all three versions of Word here, I've tried it, and it worked for me.

Front and centre

In the May issue I dealt with positioning Word tables on the page by enclosing them in a frame. Richard Guy was the first of many readers to point out that there is a simpler way if all you want to do is centre or align a table: with the insertion point in the table (but with no text selected) go to



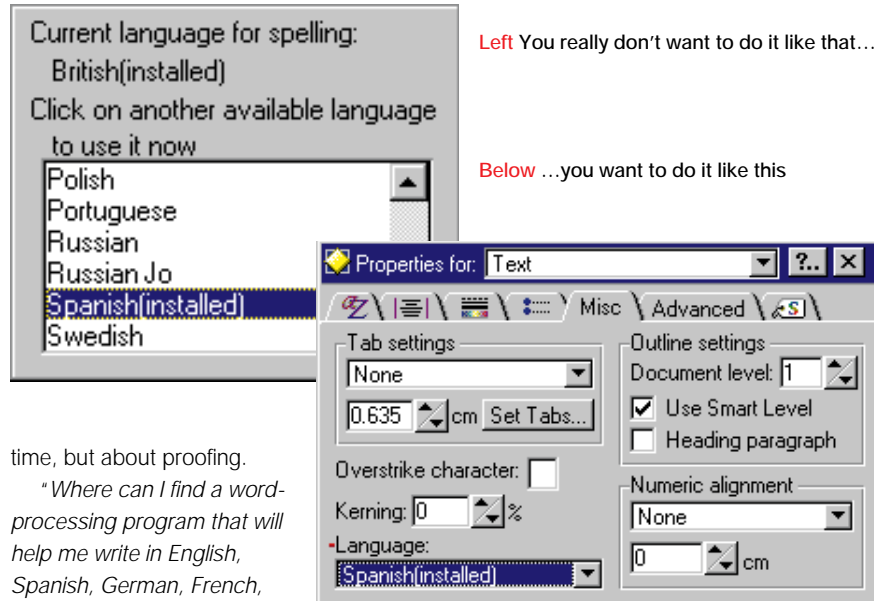
Aaah, there it is!

"Table/Cell Height and Width".

Intuitive? I think not. But here you will find three options for left, centre or right alignment, which work on the table itself rather than the cell contents. If you select a cell or row, you can stagger the table, and if you want finer control, there's also an "Indent" box. Grovelling apologies for not spotting that earlier.

Proof of the pudding

Iktomi Wakahn emailed me from Spain bemoaning the lack of support for those who create documents in more than one language. No, this isn't about accents this



Left You really don't want to do it like that...

Below ...you want to do it like this

time, but about proofing.

"Where can I find a word-processing program that will help me write in English, Spanish, German, French, Italian, Dutch and sometimes Catalan?" he asked.

"I bought WordPro on the strength of its 15 languages, but find I can only use one per document. We're all multilingual here, but even the simple task of proofing a four-language menu for the local café seems beyond the powers of the software."

Don't despair, Iktomi — you can proof a multilingual document in WordPro. In fact it's one of its selling points over Word, where you have to buy extra proofing languages separately. What isn't so good is the way the help file fails to explain how to implement this feature.

First, install the language modules you need. There is no (at least not in my 96 version) installation routine for this. Instead, you should find a README.TXT file in your LOTUS folder that gives instructions on which files to copy where from the CD.

I won't go into all the details, but, for example, if you want to install the Spanish spell checker, go to the /EXTRAS/LANGUAGE/SPANISH directory on the CD and copy LOTUSES2.DIC to the directory on your hard disk where the existing dictionaries are located. For 16-bit components this directory will normally be \WINDOWS\LOTUSAPP\SPELL, for 32-bit applications it will normally be \LOTUS\COMPONENT\SPELL.

If you then start WordPro, open the spell checker and scroll through the list of languages, you'll see that Spanish is flagged as "installed". This list is headed by an invitation to "Click on the language to start using it now". Resist the temptation to do this, as that is the path to great suffering and frustration.

Instead, close the checker, select the

text you want proofed in Spanish, right-click on it and open the "Text Properties Box". Go to the "Misc" tab and there you will find another language list. This is the one to use. With the Properties Box still open, repeat for other selections and other languages.

Now, when you run the check, it will automatically switch to the correct language for each piece of text. And will do so even in Catalan. If you want to set the default language for the entire document, you can do this from "File/Document Properties/Document/Options/Language".

Tile in style

Question: what do WordPro and WordPerfect users have that Word users still lack? Incredibly, after five versions (there wasn't a Word for Windows 3, 4 or 5) you still can't arrange two documents side-by-side without manually dragging and resizing them. Maybe Microsoft thinks this is unimportant, but given that most documents are taller than they are wide, most screens are wider than they are tall, and it's much easier to compare documents side-by-side than top-to-bottom, I find this incredible.

In fairness, I must say I did find a Microsoft macro that attempted to do this in Word 6 LAYOUTS.DOT but it went to pieces if the zoom levels were other than 100 percent. It would also restore all minimised documents, which is somewhat counter to standard practice. So I had a stab at rewriting it. The Wordbasic code for Word 6 and 7 is shown in Fig 1.

What this macro does, for those into trivia, is store the name of the active

Fig 1: My rewritten macro

```
Sub MAIN
mywin$ = WindowName$()
If DocMinimize() Then DocRestore
ScreenUpdating 0
If DocMaximize() Then DocRestore
wordwidth = Val(AppInfo$(6))
wordlength = Val(AppInfo$(7))
Dim dlg As ViewZoom
GetCurValues dlg
zoomperc = Val(dlg.ZoomPercent)
wordwidth = wordwidth * zoomperc / 100
wordlength = wordlength * zoomperc / 100
num = CountWindows()
Dim name$(num)
wincount = 0
gap = 0

If num > 0 Then
For i = 1 To num
Next Window
If DocMinimize() Then
gap = 25
Else
name$(wincount) = WindowName$()
wincount = wincount + 1
EndIf
Next i
For i = 0 To wincount - 1
Activate name$(i)
x = (i * wordwidth) / wincount
DocMove x, 0
DocSize wordwidth / wincount, wordlength - gap
Next i
Activate mywin$
Else
MsgBox "There are no open documents to arrange."
End If
End Sub
```

window, then restore it if minimised, turn off screen updating and restore it if maximised.

These last three actions are necessary to stop the macro getting confused. It then fetches the available window space from the AppInfo\$ settings and compensates for the zoom level of the active window. Then it counts the windows and steps through each one.

If it is minimised, it leaves it alone, but creates a gap at the bottom of the main window to display it (Windows 3.1 users should increase this to 50, as minimised documents are shown as icons). If it isn't minimised, it adds its name to the list and then steps through that list, moving and sizing the window accordingly, and returns

the focus to the original active window. Screen updating is turned on automatically when the macro exits.

I was going to write a Visual Basic equivalent for those of you who are working with Word 97, but by a stroke of extremely good fortune, Jason Ozin got there first with the code shown in Fig 2, saving me a lot of effort and winning himself a book token.

Fig 2: Jason's Visual Basic equivalent

```
Public Sub MAIN()
On Error GoTo skip
Dim NumberOfWindows
Dim MyWindow$
Dim AppWidth
Dim AppLength
Dim WinCount
Dim Gap
Dim i
Dim ss
Dim tt
Dim WinLeft

NumberOfWindows = Windows.Count
If NumberOfWindows = 0 Then GoTo skip
MyWindow$ = ActiveWindow.Caption
If ActiveWindow.WindowState = wdWindowStateMinimize Then
ActiveWindow.WindowState = wdWindowStateNormal
End If
Application.ScreenUpdating = False
If ActiveWindow.WindowState = wdWindowStateMaximize Then
ActiveWindow.WindowState = wdWindowStateNormal
End If
AppWidth = Application.UsableWidth
AppLength = Application.UsableHeight
ReDim name$(NumberOfWindows)
WinCount = 0
Gap = 3

For i = 1 To NumberOfWindows
Windows(i).Activate
If ActiveWindow.WindowState = wdWindowStateMinimize Then
Gap = 20
Else
name$(WinCount) = ActiveWindow.Caption
WinCount = WinCount + 1
End If
Next i
For i = 0 To WinCount - 1
Windows(name$(i)).Activate
WinLeft = i * AppWidth / WinCount
ActiveWindow.Top = 0
ActiveWindow.Left = WinLeft
ActiveWindow.Width = AppWidth / WinCount
ActiveWindow.Height = AppLength - Gap
Next i
Windows(MyWindow$).Activate
skip:
Application.ScreenUpdating = True
End Sub
```

PCW Contact

You can contact **Tim Nott** by post c/o the PCW office or at wp@pcw.co.uk.