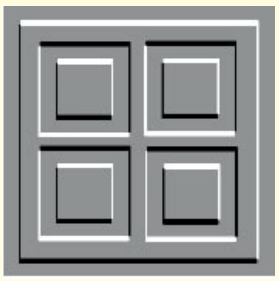


There's **life** in the old dog yet

Tim Nott highlights areas in which dynamic data exchange still wins out over OLE.

Those of you with long memories, or long bookshelves, may recall that some time ago I wrote that "DDE will be covered comprehensibly in a future *Hands On* session." I can't remember whether I really meant "comprehensively" but I think I'll try and stick with making it comprehensible for now — sorry about the wait.

Dynamic Data Exchange (DDE) dates back to pre-Windows 3.0 days and provides a way of transferring information or commands between applications without using the clipboard. To a large extent, it's been superseded by Object Linking and Embedding (OLE) but there are things DDE can do that OLE can't, at least not in its present incarnation.

One example of DDE at work is the way installation routines add icons to Program Manager, automatically. For end-user access, the applications involved must either have routines built in to the menus, or a macro language with which to construct them.

Let's look first, however, at a very simple example of one application, Word for Windows, controlling another, Cardfile, without using DDE proper.

The object of the exercise is to prompt the user to enter a word, then start Cardfile with a data file loaded and look for an entry whose title contains that word. An obvious example would be to look up an address when writing a letter. It assumes you have a Cardfile data file named ADDRESS.CRD in the directory C:\LETTERS and that CARDFILE.EXE is on your path.

The macro in *Fig 1* works with version 2.0 of Word for Windows and later. If you use a different word processor, then I'm sorry, but it shouldn't be too hard to translate. Anything preceded by an apostrophe is a comment — Word ignores these statements which are simply there to explain to the reader what's going on. There's all sorts of room for improvement, but I've deliberately kept it simple and you should see Cardfile leap into action and fill in its own Go To… box when you run it.

● *Hint: click the OK button rather than press Enter after typing the name.*

Swimming the channel

Proper DDE is far more functional — it uses "channels" to communicate rather

Look, no hands — Cardfile fills in its own Goto box via a Word macro

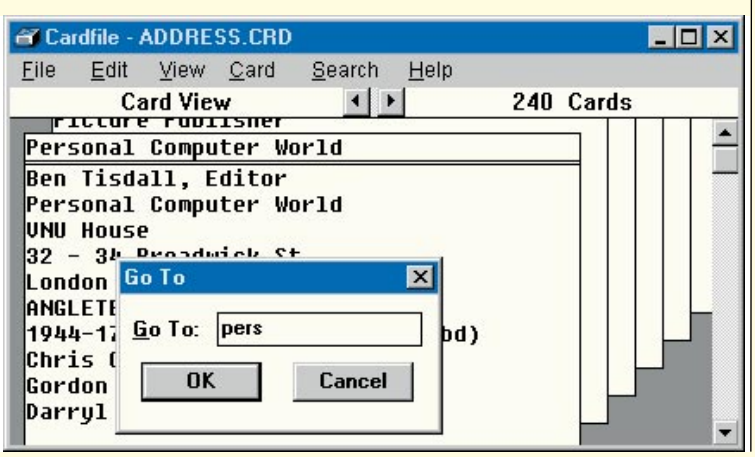


Fig 1 Word Cardfile macro

```
Sub MAIN
name$ = InputBox$("Type in a name")
        'prompts for a name
restore$ = "%( )" + "r"
SendKeys restore$
        'stores keystrokes to restore
        'cardfile if minimised
find$ = "{f4}" + name$ + "{enter}"
SendKeys find$
        'stores the go to command
On Error Goto whoops
        'if the cardfile isn't open
        'we can't activate it
        'so macro jumps to whoops label
AppActivate "cardfile - address.crd", 0
        'activate cardfile
        'and carry out the keystrokes
Goto skip
        'we don't need the rest if
        'cardfile was running so
        'jump to the skip label
whoops:
Shell "cardfile.exe c:\letters\address.crd", 1
        'address.crd wasn't loaded, so load it
        'and send the stored keystrokes
Err = 0
        'resets the error handler
skip:
End Sub
```

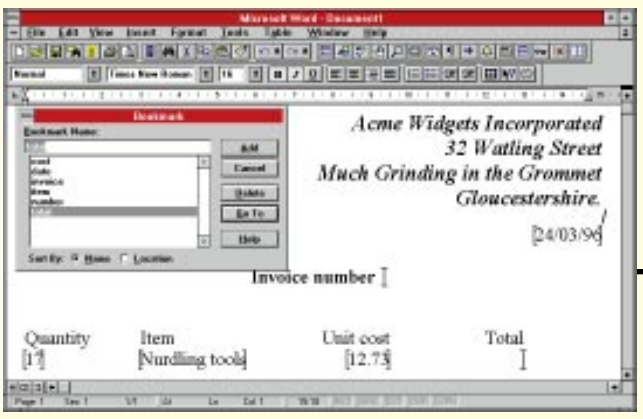
Fig 2 Word-to-Excel DDE macro

```
Sub MAIN
Dim data$(5)
channel = DDEInitiate("Excel", "list.xls")
        'opens channel 1 for DDE
StartOfDocument
lastinv$ = DDERequest$(channel, "R2C7")
thisinv$ = Str$(Val(lastinv$) + 1)
        'gets the last invoice number stored in G2
        'and adds one to create a new invoice number
firstinv$ = DDERequest$(channel, "R2C1")
currentrow = 2 + Val(thisinv$) - Val(firstinv$)
currow$ = Mid$(Str$(currentrow), 2)
        'gets the first invoice number
        'and works out which row to go to
data$(1) = thisinv$
EditGoTo "date"
data$(2) = Selection$()
EditGoTo "number"
data$(3) = Selection$()
EditGoTo "item"
data$(4) = Selection$()
EditGoTo "cost"
data$(5) = Selection$()
'reads the current invoice number and
        'bookmark contents into the array data$()
For count = 1 To 5
        cell$ = "R" + currow$ + "C" + Mid$(Str$(count), 2)
        DDEPoke channel, cell$, data$(count)
Next count
        'fills in the spreadsheet row
cell$ = "R" + currow$ + "C6"
total$ = DDERequest$(channel, cell$)
        'gets the total in col F from Excel
EditGoTo "invoice"
Insert thisinv$
EditGoTo "total"
Insert total$
        'inserts Excel data into empty bookmarks
DDEPoke channel, "R2C7", thisinv$
        'updates last invoice number
DDEExecute channel, "[SAVE()]"
DDETerminate channel
        'saves spreadsheet and hangs up
End Sub
```

than depend on the Sendkeys function.

There are only five basic DDE commands in Word Basic: DDEInitiate and DDETerminate are used to open and close a connection; DDERequest fetches data from the remote application; and DDEPoke sends data to it. Finally, DDEExecute runs a command or a macro in the remote application.

So, let's get our anoraks really muddy with the example in *Fig 2.* This takes data from an invoice created in Word, puts it in an Excel spreadsheet, picks up other items from there, places those back in



A sample Word document showing the bookmarks used in the DDE exercise

continued on page 262

A quick tip from Reg

In March's column I looked at reader Dejan Stojnic's strange case of the disappearing associations — double-clicking a data file in File Manager would no longer launch the relevant application as the Registry was corrupt.

My suggestion of moving REG.DAT and letting Windows re-create it didn't quite go far enough. You can usually save yourself the trouble of redefining associations or re-installing applications by using the File Manager Search… command to find all *.REG files — these contain third-party registration information. Double-click on each of these and the information contained will be added to the registry. If you get an error message saying there's "no association", you'll have to re-associate .REG files with REGEDIT.EXE.

the Word document and then saves the spreadsheet. I've kept it very simple and it works with Word 2.0 and Excel 4.0 upwards.

Before you run the macro, you need to do a little groundwork. First do the easy bit — start a new worksheet in Excel and enter the headings shown here in columns A-G in the first row:

```
Invoice number / Date / Quantity / Description / Unit Cost / Total / Last invoice
```

In row 2, enter any integer under "Invoice number" and the formula

```
"=C2*E2"
```

under "Total". Use the Fill command to replicate this formula down column F a dozen or so cells. Under "Last invoice" enter the same number that you put under "Invoice number". Save this as LIST.XLS.

Keep LIST.XLS open and start a new document in Word. Type in a letterhead and an address if you like, then on a new line the date, say 24/3/96. Highlight the date and create a "bookmark" (Edit menu in Word 6, Insert menu in Word 2) named "date".

On a new line, type "Invoice number", space, then insert another empty bookmark at the cursor position and name it "invoice".

On another new line type "20", select it and bookmark it as "number". Tab and type "Universal grommets", select that and bookmark as "item"; tab again and type "1.50", select and bookmark as "cost", tab again and insert another empty bookmark named "total". Save the document as BILL.DOC.

If all this seems too much like hard work, then all the files and macros can be found in DDESTUFF.ZIP on our free, cover-mounted CD-ROM.

See how they run

Still in Word, run the macro listed in *Fig 2* (page 259). If all has gone well, the date, quantity, item description and unit price will be inserted into the third row of the spreadsheet and the document will itself be updated with an invoice number and the total. The new invoice number will replace the original "Last Invoice" in Excel.

DDERequest expects an absolute cell reference; it can't find the end of the spreadsheet on its own. So this method means the macro always knows where to look for the last invoice number, and by comparing it with the first knows where to poke the next lot of data. It's quick and dirty but means we don't need any Excel macros. And note that the DDE commands identify a cell by row number/column number, so "A4" in Excel becomes "R4C1".

Try changing the contents of the bookmarks and running the macro again — you'll get a new row in the spreadsheet. Take care not to delete the bookmarks themselves.

● *Hint: type in the new text before deleting the old.*

In Word 6 you can view bookmarks (Tools/Options/View) as square brackets. In Word 2 you can't, and you're also limited to 20 characters.

This is a simple example, and not a particularly robust or sophisticated piece of programming, so please don't use it for any real work.

However, DDE can be an extremely powerful tool. There's an example bundled with Autocad that takes an Excel spreadsheet containing the calculations on a shaft, taking into account load, material properties, RPM and so on, to calculate the minimum dimensions. This is linked to an Autocad drawing of the same shaft and as you alter the figures in the spreadsheet — increasing the load, for example — the dimensions in the drawing change to suit and vice-versa. This needs a custom DLL as well as hefty Autocad and Excel macros, but shows just what can be done.

Dealing with DOS

DOS users please note: In future issues, Tim Nott will be covering DOS topics in this column on an occasional basis.

PCW Contacts

Tim Nott can be contacted either by post c/o *PCW* or by email at **timn@cix.compulink.co.uk**