The need for speed

Stephen Wells looks at a new twist on the common problem of achieving raw performance with Excel and how to optimise spreadsheets. And help is at hand for oil men and pilots.

An extreme example of a common problem arrived this month by email from John Ferguson.

"*I need to supply an actuarial team with raw performance on Excel. I could ask them to rewrite their spreadsheets in a better fashion (I am sure their spreadsheets are inefficient) but it would be difficult to persuade them to do this. The spreadsheets are about 7Mb in size. Loading takes 30 minutes on a Pentium P75 with 32Mb memory, Windows 3.11 and Excel 5.0c. Changing to NT 3.51 and Excel for NT does not speed up the load or recalc time. I can either supply them with increasingly higher-spec PCs or an alternative system. I could let them use Visual Basic 5, but I do not expect to see any great improvement with this because they expect to take their VBA Excel macros directly to Visual Basic. Essentially, the spreadsheets test profitability.*"

I have to assume these spreadsheets take 30 minutes to load from an intranet, not a disk. But my reply zeroed in on some key points: developments in Excel and VBA, inefficiency of design, and use of macros.

The calculation engine in Excel 95 was rewritten and is much faster than Excel 5, and the same engine is used in Excel 97. VBA Version 5, which comes with Office 97, is faster than the Excel 4 macro language and earlier versions of VBA.

For one thing, the Visual Basic for Applications object library is no longer a standalone file. It is integrated into the dynamic-link library (DLL). And there are many other efficiencies: for example, in Excel 5, the Names collection is accessed through the Workbook object. Excel 95/7 provides a new Names collection for the Worksheet object as well. I suspect the root problem for John is one of available expertise. A DTP package does not an art director make, nor Excel an application developer. Optimisation of an Excel spreadsheet starts with better formatting, then more use of Excel functions before macros and then more efficient macros.

Here are some examples. Say cell E1 displays the date of the first day of next month in a normal date format of d/m/yy. In cell D1 you want to show the name of that day. It's not uncommon to use a LOOKUP table. You could have the numbers of the days of the week, 1 to 7, in the range F2 to F8, and the names in G2 to G8. So, in D1, you could have the array formula

```
{=LOOKUP(WEEKDAY(E1),F2:G8)}
```

You use Ctrl+Shift+Enter and Excel adds the curly brackets. WEEKDAY provides the number of the day of the week and the LOOKUP function provides the name from the table. But it is far more efficient of space and memory, and you display the same result, if you simply enter =E1 in cell D1 and custom format the cell as dddd.

Moving back to cell E1: many users would create a function to find the first day of next month like this —

```
Function FirstOfNextMonth()
 FirstOfNextMonth = _
  DateSerial(Year(Now), Month(Now)
 + 1, 1)
End Function
```

Then they would write a macro to display it in the cell, like this:

```
Private Sub Sheet1()
Range("E1").Value =
FirstOfNextMonth()
End Sub
```

Again, it would be more efficient to simply enter in cell E1:

```
=DATE(YEAR(TODAY()),MONTH(TODAY())+
1,1)
```

Obviously there are going to be times when an application needs macros. Microsoft itself makes a number of suggestions for speeding up Visual Basic for Excel. These tips can also save memory.

The first is to streamline any code produced by the macro recorder which is a wonderful device for beginners because it automatically produces code. But it is there for convenience rather than economy. It blindly duplicates the keystrokes you make, but it's not psychic. For one thing, it can't tell which options you've changed in a dialog box so it sets all of them. You might change FontStyle to Italic, but the macro recorder will list every formatting variation, with all the others individually set to False. You can shorten what might be a 13-line macro to one line by eliminating the unnecessary instructions.

Also, following your keystrokes, the macro rdecorder will activate or select objects before it specifies any action. Again, if you edit the resulting code you can remove all the Select method calls and use a With statement instead.

When you write your own code, it's important to remember that every dot, or full stop, you use is an OLE call for a method or property. Reduce the number of dots and you expedite the code. You can do this not only through With statements, but by setting an object variable or using a For Each…Next loop. Another good tip is to use Excel functions within a macro. Functions that can take a range as an argument, like SUM, MATCH and LOOKUP, are faster than VBA code equivalents.

Getting sorted

In the post this month arrived a letter from G Eames, of Ramsbottom, which I suspect is more about formatting than sorting.

"*Can you please help? The problem is sorting. Using Excel, when preceded by a word, numbers are treated as separate digits so 100 is placed before 4, and so on. I have enclosed a listing to illustrate the effect.*" (Fig 1)

	A	B	C	D	E
1	HAYDN	SYM 101	VPO	Bern'ein	T
2	HAYDN	SYM 104	BPO	Karajan	BB
3	HAYDN	SYM 104	NYPO	Toscanini	BB
4	HAYDN	SYM 27	A-H HAY	Fischer	I
5	HAYDN	SYM 4	L.PHIL	Beecham	
6	HAYDN	SYM 45	VSO	Moralt	I
7	HAYDN	SYM 48	PH HUNG	Dorati	I
8	HAYDN	SYM 49	St JOHN	Lubbock	I
9	HAYDN	SYM 59	AOSMITF	Marriner	Z
10	HAYDN	SYM 6	PH HUNG	Dorati	I
11	HAYDN	SYM 88	VPO	F'wang'er	AA
12	HAYDN	SYM 92	PARIS CO	Walter	BB

Fig 1 How can you sort on column B which has a mix of text and numbers? Format as a number and include the text in a custom format

There are ways around this. You could put the SYM, which I presume is short for symphony, in a separate column from the numbers. But I think you may prefer the idea of formatting this range of cells with the custom format, "SYM" 0. All you enter are the numbers but they will display preceded by a SYM. Then you can sort the whole block, based on Column B, and it will descend from SYM 4 correctly to SYM 104. If, later in the listing, you need, say, CON, for concerto, you just change the formatting for that range of cells.

As this looks like a music collection, you might be interested in the Music Collection Database template which comes with Microsoft Access 95 and 97. Excel is fine for small databases but Access is better designed for the job. Incidentally, until the latest version, Excel displayed dates alphabetically rather than numerically. The new Excel 97 helps solve sorting and formatting by displaying dates as numbers rather than text. When dates are sorted in ascending or descending order, they are listed in appropriate chronological order. It doesn't matter whether the dates are formatted as 6/9/97 or September-97.

Flying high

Here's that old problem again of subtracting times. Ron Whytock emailed me from Singapore, *en route* to Manchester.

"*I run quite a large spreadsheet in Excel 7 for my flying logbook. I need to be able to subtract two times (24-hour clock), one from one another.*"

Regardless of the format used, Excel stores any date as a serial number and it stores any time as a decimal fraction. For instance, 23/6/97 22:20 is stored as 35604.93.

The way that a time or date is displayed on a worksheet depends on the format applied to the cell. When you type a date or time that Excel recognises, the cell's format changes from the General number format to a built-in date or time format. By default, dates and times are right-aligned in a cell. If Excel cannot recognise the date or time format, the date or time is entered as text, which is left-aligned in the cell. Options you can select in the Regional Settings of Control Panel determine the default format for the date and time, and the characters used as date and time separators. To type a date and time in the same cell, separate the date and time with a space.

Excel separates date elements with a slash (/) and time elements with a colon (:). By default, Excel bases the time on the 24-hour clock. If you type 3:00 instead of 3:00PM, the time is stored as 3:00AM. To type a time based on the 12-hour clock, type a space followed by A or P after the time.

To see the underlying serial number which is stored, select the cell, choose Format, Cells, Number tab, General. Times and dates can be added, subtracted, and included in other calculations. To use a date or time in a formula, enter the date or time as text enclosed in quotation marks.

So when I replied to Ron, I suggested that he simply enter the correct date and time when he went on duty, and again when he went off duty, then in a third cell subtract one from the other. So A1 might display 23/6/97 22:20 and B1, 24/6/97 03:25. C1 would have the formula B1 — A1. It would be formatted as h:mm and the result here would display correctly as 5:05.

Easing into oil

The problem posed by oil consultant Gordon Smith seemed simple enough:

"*We use Excel 7 to analyse data for a number of different wells and we want to chart rock porosity against the amplitude of permeability.*

"*Say we have an X Y data chart made from porosity values listed in column B and the amplitude in column C. What causes a problem is that having created the chart we want to label the points, not with the porosity or amplitude but with the equivalent well names, listed in column A. Typically this would be an alphanumeric field looking like 30/17b-A21Z. How can you easily add this information to the chart? At the moment, we edit each label manually and, faced with 30 or so wells, this is a problem. Is there an easier way?*"

Gordon attached a sample Excel file for me to play with. He also said that when his office used Lotus 1-2-3 there had been no problem. So first I imported the file into 1-2-3 Version 5 and, sure enough, there is a dialog box for specifying the range of the worksheet to refer to for the data point labels. Then I opened the file in the feature-packed Excel 97 and, to my surprise, there is no equivalent. You can format data labels every which way from Sunday but to change them you have to do it one label at a time.

"I must be doing something wrong," I thought. Time to consult my friendly guru, Michael Rickard. As usual, he's rescued my rashers. First of all he knocked up the short VBA macro (Fig 2) which worked for me in Gordon's worksheet, first time. But he also

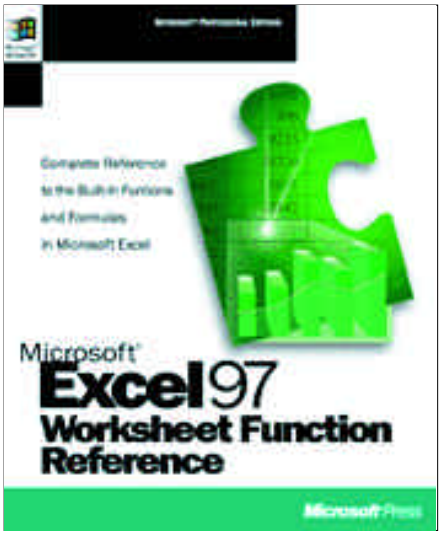
p268 ➢

Fig 2: A macro for changing data point labels on an Excel chart

```
 Sub addlabels()
'assumes labels are in range A2 to A28
'select the appropriate worksheet first
Set LabelRange = Range("a2:a28")
With ActiveSheet.ChartObjects(1).Chart.SeriesCollection(1)
    .ApplyDataLabels
    For i = 1 To .Points.Count
        .DataLabels(i).Text = "=" & LabelRange.Cells(i).Address _
            (ReferenceStyle:=xlR1C1, external:=True)
    Next
End With
End Sub
```

Quick looks at new books

Microsoft Excel 97 Worksheet Function Reference



With version 4.0 of Excel you get a two-volume User's Guide and a separate 580-page Function Reference book. For version 8.0, or Excel 97, the book has shrunk to 308 pages and it now costs £22.99. What's missing?

Well, the Excel 4 macro language functions became redundant, replaced by the properties in VBA v5.0 which comes with Office 97. And, the short lists of related functions which appeared after each function have been dropped. Related to INT, for instance, is CEILING, FLOOR, MOD, MROUND, ROUND and TRUNC. These prompts were useful because they often suggested other ways of doing things. However, the other four main sections for each worksheet function (definition, syntax, remarks and examples) are still there. All the most recently added functions are included, too. All the worksheet functions are grouped by category at the front of the new book, as before, but the previous list of changed functions has been dropped. As has the book's index.

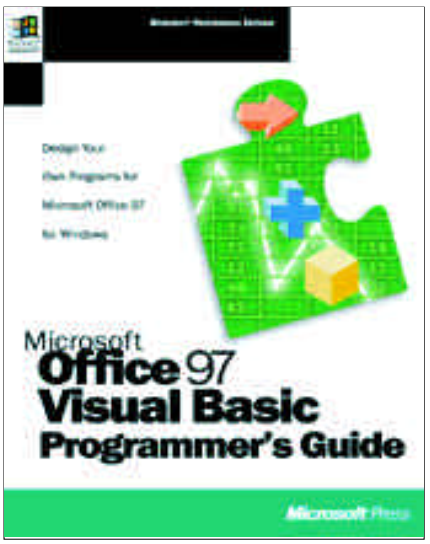
Of course, if you don't have the Excel 4 version on your shelf and you like to sit down with a book instead of working through the on-line function list and help files, this up-to-date version of the *Excel Function Reference* book will be right for you.

Microsoft Office 97 Visual Basic Programmer's Guide

With no visible module sheets and a bewildering multi-window opening screen, experienced and neophyte programmers alike may be initially confused by the Visual Basic Editor in Excel 97. This book is an excellent introduction. It clearly describes, with a mass of examples, how to write, edit, store, run, optimise and debug VBA code. Right at the front there is a detailed illustration, with call-outs, of all the Editor windows and associated boxes and toolbars, which gets you off to a good start.

With VBA Version 5, Microsoft has reduced the differences between VBA for Excel, Word, PowerPoint and Access so the programming environment is now all integrated, including Outlook (the information management program) and Office Assistant (an advanced help system). The sections on creating custom commands, menus, dialog boxes, messages, and buttons, as well as the online help, apply to all these applications. But there are also chapters on the specific objects required in each application. The Excel section details the Workbook and Range objects and their properties as well as support for event-driven programming.

There is everything from programming basics to dealing with the drawing layer, ActiveX Controls and developing applications for the internet and the World Wide Web. A comprehensive appendix details how VBA 5 differs from the Excel 4 macro language.



This book, priced at £32.49, is a first-class introduction to VBA 5 for anyone developing Office 97 under Windows 95, Windows NT or for the Macintosh.

Both these books are published by Microsoft Press and are in the *Microsoft Professional Editions* series. They are available from Computer Manuals (see the "PCW Contacts" panel, below).

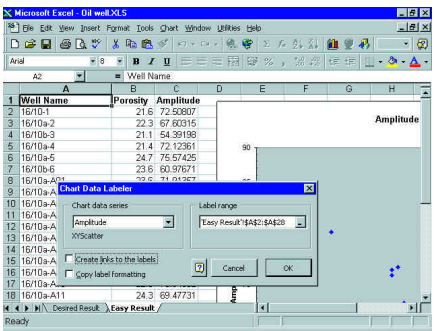


Fig 3 The Chart Data Labeller utility in the registered version of John Walkenbach's Power Utility Pak for Excel 97

recommended Rob Bovey's equivalent utility, a Rolls-Royce job with frills and full documentation. You'll find the Excel 97 version of this on our *PCW* CD-ROM, in the packed file, label_97.exe. If you want versions designed for Excel 5 and 95, download them from www.baarns.com/. There is a comparable utility in the registered version of John Walkenbach's Power Utility Pak 97 (Fig 3). The shareware, unregistered version is at www.j-walk.com/ss/pup97.htm. I've put an earlier version of this Pak, which works with Excel 95, on our CD-ROM. Although this version doesn't include the labelling utility, it offers 21 general-purpose utilities, 23 new worksheet functions, and enhanced shortcut menus.

On the *PCW* CD-ROM

1. In the Software Library, Hands On Spreadsheets section this month is John Walkenbach's Power Utility Pak 2.0a. Click on the file xlpowr2a.exe to unpack it in the directory of your choice.
2. Many loans allow for the early prepayment of the principal at any time. This results in savings of interest and a shorter loan period. Roy Murphy's file, prepay.xls, for Excel 4 and above, is a worksheet which examines the effect on the loan of paying different amounts in each period.
3. The Excel 97 version of Rob Bovey's XY Chart Labeller is in file label_97.exe. Click to unpack. Email addresses for John, Roy and Rob are included in the respective files.

PCW Contacts

Stephen Wells welcomes problems or solutions relating to spreadsheets. Write to him at the usual *PCW* address or email spreadsheets@pcw.co.uk

Computer Manuals 0121 706 6000; www.compman.co.uk