Trading places

Stephen Wells shares his stock of spreadsheet knowledge to help you chart your investments

Paulo Freitas Tavares (MD), emails: "*I have a small problem. It extends for dozens of sheets and many parameters but let us suppose it is only one sheet and one parameter.*

"In column B I have weeks; in C I have weights of patients; in column J I have 'alerts' for a quick visualisation if something goes wrong. Suppose that the criteria for 'going wrong' is losing or gaining more than 10 percent weight in one week. The aim is to get the word ALERT in red or the word OK, in blue, in column J — I can't make Excel 7 do it."

As regular readers know, I am not a big fan of unnecessary macros and always try to find a built-in Excel solution before using them. In this case, Dr Tavares may be able to take advantage of the extraordinary flexibility of Excel's Custom Number format.

In Excel 4, you can enter the following as a Custom Number format:

```
[>1000] [Blue] #,##0; [<-1000][Red] #,##0;[Green] #,##0
```

Excel assumes that the first section is for positive numbers, the second for negative ones, and the third is anything else. You can't write IF statements, but you can use a condition value symbol. So in this example, any entry in the cell greater than 1,000 is displayed in blue, less than -1,000 turns red, and anything else is green.

Interestingly, you can enter the same custom format in Excel 7 but it will automatically shuffle the description around to:

```
[Blue][>1000] #,##0;[Red][<-1000] (#,##0);[Green] #,##0
```

In this example I've added parentheses, so numbers less than -1000 have brackets as well as being in red.

You can also make text appear even though you have entered a number, or the cell contains a formula which produces a number. Let's say the cell A9 contains the simple formula:

```
=C9-C10
```

We can format A9 with this Custom Number format:

```
[Blue][>14]"OK";[Red][<10]"ALERT";"Other"
```

If the answer to C9-C10 is 15, the acronym "OK" will display in blue; for 8, it will display "ALERT" in red; for 11, it will display "Other" in black.

Previously, I said that the second section of the Custom Number format is for negative numbers. But there is an exception, as in this example. When the first section is conditional (as it is here because it only applies to numbers bigger than 14) then the second section formats other numbers, whether positive or negative. Here the second section ([Red][<10]"ALERT";) is also conditional. So then the third section applies. In this case, if the number in the cell is between ten and 14, then the word "Other" prints, using the default formatting for the cell. The only problem here is that Dr Tavares says that there are many parameters in his actual workbook. Other readers may also like to have Excel automatically change the font of a warning word, or the background colour of its cell.

So, for those who need it, I'm providing a macro in VBA for Excel on the cover CD in the workbook file, ChangeColour.xls. See also Fig 1 (page 272). In this instance, the word to be emphasised is decided on the worksheet by an IF function, like this one:

```
=IF(C15<10,"ALERT",IF(C15>=14,"OK","Other"))
```

There is a button on the worksheet which runs a macro called Changing( ). This specifies a range, although you could use a Name, and the macro runs through that range looking for words which the IF statement has entered. With a macro, you can have as many keywords to look for as you like. It then uses the IF THEN WITH statement to abbreviate font references.

The ColorIndex statement refers to the standard Excel palette box. If you count colours from left to right, and top to bottom, you'll find that 2 is white, 3 is red, 5 is blue and 27 is an off-yellow. So looking at the listing, you'll see the word OK will appear in white on blue and the word ALERT will be red on off-yellow.

ALERT is in Arial Black Bold Italic. OK is in Roman (not italic). You have to include the instruction

```
Italic.False
```

for the word OK because after the macro has run, the font will be set to italic by the instruction,

```
.FontStyle = "Bold Italic"
```

for the `word ALERT`.

p272 ➢

EXCELlent shortcuts and longshots

1. Worksheets saved in the MSOFFICE\TEMPLATES folder will behave like an XLT file and open as a copy, without having to be saved as a template file.
2. Many useful macros, which you can copy into your workbooks, can be found in the SAMPLES.XLS file located in the MSOFFICE\EXCEL\EXAMPLES folder. They include error trappers and default resetters.
3. You can copy colour palettes between workbooks: open the workbook with the colour palette you want. Switch to the workbook to which you want to copy the colour palette. Choose Tools, Options, Colour tab. In the Copy Colours From box, select the workbook that contains the colour palette you want to copy.
4. Right-click the mouse on the AutoSum feature in the Excel Status bar. Change Sum to Average, Count, Count numbers, or find the Max or Min of a selected range of cells.

Up and down the City road

Chris Pack emails: "*I often need to chart market prices, which involves a long series. Daily prices over two or more years can be some 500 plots. It would be nice to label the months along the x axis but calendar months are not evenly spaced and this seems to make them difficult for Excel to display. I have set a column for these labels, but with so many plots Excel appears to be reserving space for the blanks.*" Chris then described all the tricks he's tried, and ended "*…The whole hit-and-miss process seems so time-consuming. I feel sure there must be a simpler way of formatting such X axis labels. Any suggestions?*"

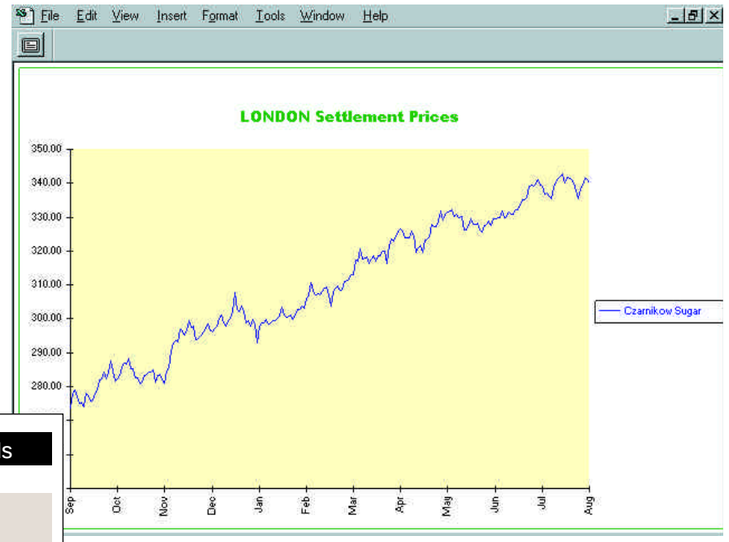
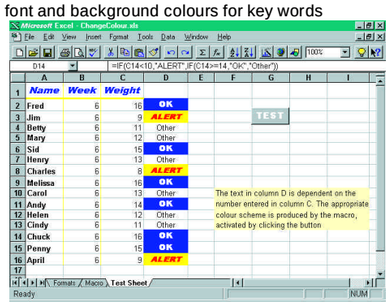


Fig 2 You can format the labelling for axes of an Excel chart on the data source worksheet

Fig 1 Macro — a few colourful words

```
Function Changing()
 Dim Cell As Range
   Worksheets("Test Sheet").Select
 For Each Cell In Range("D2:D16")
  If Cell.Value = "OK" Then
   With Cell.Font
 .Name = "Arial Black"
  .Bold = True
   .Italic = False
 .Size = 10
  .ColorIndex = 2
 Cell.Interior.ColorIndex = 5
   End With
    End If
 If Cell.Value = "ALERT" Then
  With Cell.Font
   .Name = "Arial Black"
     .Size = 10
  .FontStyle = "Bold Italic"
   .ColorIndex = 3
 Cell.Interior.ColorIndex = 27
      End With
       End If
  Next
End Function
```

Fig 1 A macro, started with the button, changes the font and background colours for key words



I returned Chris's workbook with an alternative worksheet and chart. First, I eliminated the extra column. Column A still has all the dates but I used the Custom date format mmm. Column B has all the prices. I let the chart wizard make a new chart. Then I opened the Format Axis dialogue box, for the x axis. In the Scale section of this box you can choose individually the "No. of categories between Tick-Marks" and the "No. of categories between Tick-Mark labels". I experimented with different numbers until there was just one label and one tick mark for each month (as in Fig 2). The number happened to be 21, which is probably the average number of trading days each month. I accepted the defaults for the other four options in this useful box . So now, as required, the x axis is labelled only with the name of each month.

Playing footsie

In my October issue column last year, I encouraged readers to write to me with their experiences of downloading information into spreadsheets and how they used spreadsheets to aid with investments. So I was glad to conduct the above exchange with Chris, as well as to receive an email from Keith Bladon, who downloads share prices via a Teletext card and a package called Updata Teleshares. He uses Excel to analyse the FTSE 100 index. He looks at a 201-day centred average of the FTSE within a channel plus and minus 150.

Investors who use technical analysis, agreeing with Shakespeare that past is prologue, look at historical results. This is opposed to fundamental analysts who keep abreast of things like new products and management changes. Within these two major approaches are multitudes of different theories, often based on the expectations of various cycles.

Keith is a 201-day man. His 1,700-row datasheet records the FTSE for every trading day from 1st January, 1990 to 28th August, 1996. Additional columns make calculations based on percentages and other statistical changes. Another sheet in the workbook file he attached for me has a graph of these results. And then there is a long VBA macro.

His problem is: "*When I want to look at different periods of time, adjusting the graph's normal facilities is time consuming.*" After entering a start and end date, the macro finds the correct cell references and then amends the graph properties.

Keith's question is: "*Because I am using Active-Sheet, the display jumps to the various parts of the graph. I have tried to access the graph's properties without using Active.Properties but have been unable to do this. Is there a way?*"

The objects in Excel spread out much like a tree, going from the trunk to the boughs, to the branches, to the twigs. There is an established hierarchy of the 128 programmable data objects in Excel 5 and the 162 objects in Excel 7. You tie them together using Visual Basic for Applications (VBA), Excel's programming language. A full hierarchical path might read like this:

```
Application.Workbooks(1).Worksheets
(1).Range("A1").Value = 1
```

It is not always necessary to detail the entire object path when setting a property, or calling a method, on a particular object. It depends on the context. To start with, Application refers to Excel, so if you're in Excel you don't need to reference the Application object. But although defaults can often obviate entering any step in a macro, Excel can't get from here to there without traversing the steps between.

However, you don't have to watch all the changes taking place to the objects in your chart, one by one. Do you recall how, in the old days, we used to avoid seeing DOS batch files running on the screen by using ECHO OFF and ECHO ON? There is an equivalent command in VBA. Near the top of your macro just insert:

```
Application.ScreenUpdating = False
```

Your macro will run but the results won't display until it's finished and then they will all show at once.

Inspired by Keith's efforts, I wrote the small Excel 7 application, "Bulls 'n' Bears", which is on this month's cover-mounted CD. But it deploys little VBA. The eight charts it includes were made using Excel's charting wizard.

On our cover CD-ROM…

- A small stock-charting application which will run under Excel 7 or above. It displays charts of FTSE 100 closing daily prices for seven years.
- A shareware program, downloaded from the Excel web page, called Portfolio. The file, Portfo.xlw, will run under Excel 4 or above. A description is in the file Manual.rtf. The program is designed to help you keep track of daily stock quotes that you have downloaded in CSV, WKS or DOS TXT formats. It includes a simple graphing facility. Everything is US-orientated so you would have to adapt it for UK use.
- The macro illustrated in Fig 1, and examples of elaborate Custom Number formats, are in the file ChangeColour.xls.

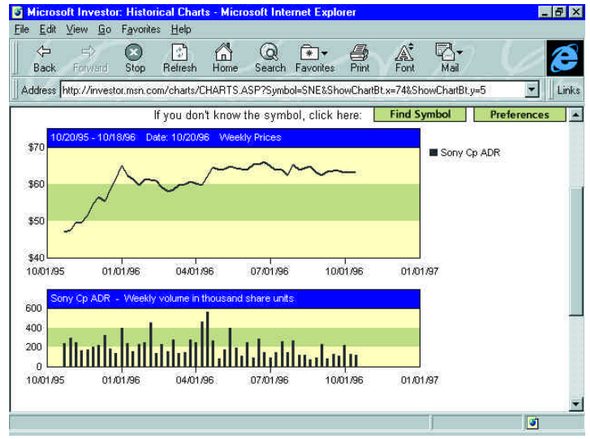


Fig 3 Using Microsoft Investor 2.0, you can download historical data on any listed US stock

Inspired by Keith's efforts, I wrote the small Excel 7 application, "Bulls 'n' Bears", which is on this month's cover-mounted CD. But it deploys little VBA. The eight charts it includes were made using Excel's charting wizard.

Something in the City

Probably the best news for those interested in the stock market is that Microsoft is in the process of localising Microsoft Investor 2.0 for the UK market. You can view this comprehensive product at www.investor.msn.com. It includes a number of related tools.

The Portfolio Manager helps users to create and track multiple stock portfolios. It recognises stock splits and multiple purchase dates and tracks commissions. You can change columns with right-click menus, or double-click columns to "AutoFit" them, just like in Excel. It offers automatic notification when there is news on any stock in your portfolio.

For those into technical analysis, Investor 2.0 supplies historical charts, for any time period, on every listed stock. They can be overlaid with market indices or compared with other securities and downloaded (Fig 3). The product also provides business and financial news from MSNBC, PR Newswire and Business Wire for fundamental analysts.

The Market Summary feature provides up-to-the-minute information on the leading US and foreign markets and currency rates and displays top-ten lists with the market's best and worst performers. Users can look up prices for specific securities by ticker symbol, company name or fund name.

Although Microsoft stresses the integration of Investor 2.0 with Microsoft Money, much of the data can be just as easily imported into Excel. Roll on the day when Investor 2.0 shows the FTSE 100 and other UK listed stocks.

I think we should be told

You may recall the discussion in past columns about calculating the years and fully-completed calendar months between any two dates. It appeared to be easier to find the required solution in Lotus 1-2-3, using the @DATEDIF function, as Excel doesn't offer an equivalent.

But now comes an email from Paul Bloomfield who points out that although it's not listed in the Function Wizard, nor mentioned in the documentation, Excel will indeed accept and correctly use a DATEDIF function — I tried it and he's right. The only possible reason I can think of for this is that Excel is always keen to be able to import 1-2-3 files and so makes allowances.

PCW Contacts

Stephen Wells welcomes comments on spreadsheets, and solutions to be shared, at spreadsheets@pcw.vnu.co.uk

Updata Teleshares from Updata Software, Updata House, Old York Road, London SW18 1TG. Phone 0181 874 4747; email Sales@updtsoft.demon.co.uk

Microsoft Investor 2.0: www.investor.msn.com