A **bed** for the **night**

Stephen Wells **examines a solution for calculating student bed nights, puts error reporting into apple pie order, and springs up with a custom-designed dialogue box.**

You've probably played the party game "Chinese Whispers" where half a dozen people sit on chairs in the middle of a room and the first person tells the second a short story. The second then tells the third and so on, along the line. Everyone is amused by how much the story changes: the fifth version is quite different from the first — and often better.

Chinese whispers

A similar sort of thing to this game happens in our forum within these pages: in the January issue, I passed on an Excel User Defined Function for calculating occupied bed days in a hospital (sent in to me by Neil Bain). However, he wrote it in VBA (Visual Basic for Applications) and I translated it into an Excel 4 macro because that can be used by Excel 4, 5 and 7 users. In fact, there is even an "MS Excel 4.0 Macro" option on the Excel 7 Insert, Macro menu.

This motivated Dave Parry to email me, saying that they've been calculating bed nights for students in residence at Imperial College for so many years that they devised their formula in Lotus 1-2-3 1A!

From the brief description Dave has provided, I've recreated his solution in 1-2-3 Release 4 for Windows. A typical result is shown in *Fig 1*: Column A shows the date the student moved in; column B, the date they moved out; C1, D1 etc, are named Month1, Month2 and so on. See the formula in *Fig 2*.

Fig 2

The basic formula is in cell C2:

```
@MAX(0,@MIN(Month2,$B2)-@MAX(Month1,$A2))
```

This is then replicated across the worksheet

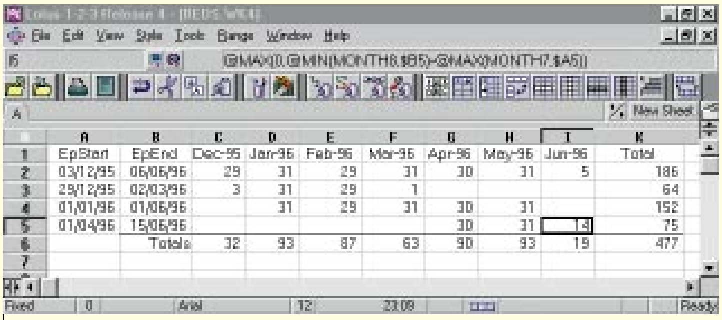
```
    D2= @MAX(0,@MIN(MONTH3,$B2)-@MAX(Month2,$A2))
and C3= @MAX(0,@MIN(Month2,$B3)-@MAX(Month1,$A3))
and D3= @MAX(0,@MIN(MONTH3,$B3)-@MAX(Month2,$A3))
```

and so on.

You always need one more column of formulas than your highest date. In this example, we need column J. There are the same formulas in it although I've hidden the column in the display.

Dave says that it only works when the Ep(isode)Start >=EpEnd but nobody's going to leave before they've arrived, so I don't see that as a problem.

Also, it is assumed that the end date is actually the morning after the day the student vacates the bed. You can easily drop that extra day by having the Month2, for example, as Month2 - 1.


Lotus 1-2-3 Release 4 - [BEDS.WK4]
File Edit View Style Tools Range Window Help
New Sheet
Fixed
0
Arial
12
23:08
Ready

I5 @MAX(0,@MIN(MONTH8,$B5)-@MAX(MONTH7,$A5))

	A	B	C	D	E	F	G	H	I	K
1	EpStart	EpEnd	Dec-95	Jan-96	Feb-96	Mar-96	Apr-96	May-96	Jun-96	Total
2	03/12/95	06/06/96	29	31	29	31	30	31	5	186
3	29/12/95	02/03/96	3	31	29	1				64
4	01/01/96	01/06/96		31	29	31	30	31		152
5	01/04/96	15/06/96					30	31	14	75
6		Totals	32	93	87	63	90	93	19	477
7										

Fig 1 *Calculating student bed nights in residence, using Lotus 1-2-3 R4*

Dave points out that: *"The interesting thing is the different approach to the problem. We were thinking graphically and used the MIN and MAX functions to discard the nights outside the period in question."*

I would add a couple of comments. If you have a solution that has stood the test of time for you, there is no point in changing it. However, creating a function (as previously shown in the hospital example) uses far less memory. So if the version of your spreadsheet allows you to custom-design a function and you expect to have a large worksheet, then you'd be better off following that road.

If you're counting, Dave provided version three. Then I received this email from Len Harvey: *"Great article in this month's mag (January). Can you provide the Visual Basic module for the Bed Sheet Days Function?"*

So I thought I'd offer Dave's solution, written as a VBA macro which creates an Excel function. The EpStart and EpEnd date input ranges have been renamed Start and Ends respectively.

Creating the function in Excel 7 is automatic once you've entered the macro to your workbook. To do that just choose Insert, Macro, Module and enter the

Fig 3

```
Function Beddays(Start, Ends, Month1, Month2)
          Dim Start As Date
          Dim Ends As Date
          Dim Month1 As Date
          Dim Month2 As Date
    ActiveCell.Formula = _
      "=MAX(0,MIN(Month2,Ends)-MAX(Month1,Start)"
     End Function
```

details shown in *Fig 3*.

When you return to your worksheet, you select the first cell where it's needed then choose Insert, Function, User Defined, and then the new function, Beddays.

If the macro finds the Names on your worksheet, it will use them automatically. Otherwise it will run the Function Wizard and ask you to insert data or the cell where the data can be found.

Other readers will recognise that this method of dividing up periodic events across a calendar has many more applications than for slumbering patients, students or hotel guests. If you adapt it for planning staff holidays, recording sunny days, or part-timers' schedules, let me know. I look forward to seeing version five.

Quick tip

Incidentally, if you ever get stuck while you're writing a VBA macro in Excel 7, here's a tip which is worth the price of this magazine and more.

Click the place in the macro where you want to insert the new code. Then choose Tools, Record Macro, Mark Position For Recording.

Click the tab for your worksheet. If the Visual Basic toolbar is not showing, right click on the standard toolbar, then select Visual Basic.

Go to the starting cell. On this Tools menu, choose Record Macro, Record At Mark. Then, using the keyboard (not the mouse) do what you want the macro to do. To finish, click the button with a black square on the VB toolbar (Stop Macro).

Now click the Module 1 tab (or whatever you've renamed it) and you'll find Excel has written the code for you. Bless you, Bill!

To err is...

And so to a question about error reporting from Nick van Terheyden who emailed me from the Saudi Arabian office of a "Big Eight" international accounting firm:

"I have a largish spreadsheet that picks up a figure and places it in a column based on the title of the column. This is for dividing expenses into different categories.

"The formula is:

```
=IF($H6=J$4,$D6,"")
```

"My difficulty is that the worksheet is too wide to fit on the screen so I cannot always see if the amount is put into a column. I want to add a validation column that shows an asterisk, or something, if I enter an invalid code. This would save scrolling across to check whether the amount has been categorised correctly. The best I have done is:

```
=IF(ISERROR(LOOKUP(H6,Valid_Values_
Dec95)),"*","")
```

"But this only reports an error if you enter a number. Any text, and it leaves the cell blank."

With Excel 4 and better, you have a number of options. First, instead of

```
=IF($H6=J$4,$D6,"")
```

if you use

```
=IF($H6=J$4,$D6,ERROR)
```

then when your cell with ISERROR in it

EXCELlent shortcuts and longshots

● **Watermarks.** Like to have a watermark behind your worksheet printouts? You can with Excel 7. Drag aside a small block of your worksheet. Click the Drawing button to display the Drawing Toolbar. On this, click the Text Box button. Drag on your worksheet to produce a box the size of the block. Replace the original cell contents. Outside your worksheet area, produce a duplicate-sized Text Box. Inside, type text (such as CONFIDENTIAL, or your name) or insert a picture (such as your company logo).

To format just the text, select it and a Font dialogue box only will appear. If you select the whole Text Box, the Format dialogue will offer five tabs including Font and Patterns. If your printer doesn't recognise colours, such as a light grey, you can format the type as white and the box in a light pattern.

● **Macro Buttons.** It's easy to produce a new button on any Excel 7 toolbar which will run the macro of your choice. First, right-click on any toolbar, then choose Customize. Drop down the Categories of toolbars to Custom. Drag the blank button (or any design you like) out of this box and onto any toolbar in your workspace. An Assign Macro dialogue box will then appear with the available macros listed. Click the macro you want to assign to this button. Then close the Customize dialogue box.

Now, whenever you click that button, your macro will run.

Down the wire

Having signed on with the Microsoft Network (MSN) I discovered that the Microsoft Exchange options, to request a receipt when a message is delivered (and also when read), currently only work with a correspondent who is on MSN. Then again, it would appear that emails with attached files won't cross networks, although heaven knows I initially received enough of them (uninvited) from MSN sales people.

One day there will be greater compatibility. In the meantime, I was contemplating how I could distribute my templates down the wire. What I needed was to find someone who is interested in financial analysis and who is also on MSN.

Then out of the blue came an email from John Woollam. The name may mean little to you now. But, believe me, when the history of this column is written John will feature as a major contributor to its progress. If I could aspire to the roll of that great journalist, Sir Henry Morton Stanley, then John would be David Livingstone.

His first email read: "*Is it possible, please, to send me by email the Excel spreadsheet templates for service companies and those that carry stock? This is the first time I've found someone in the mags using MSN.*"

I sent him the templates. He emailed them back so I could open them and check that everything was okay. Then we embarked on an orgy of file exchanges: in colour, sound, photos, fancy-formatting, spreadsheets, charts, and more. Apart from font files, every file extension there has arrived in working order, just like it does with my cc:Mail Mobile entry to the *PCW* office network.

So the moral is: if you're on MSN and you want to send or receive spreadsheets, just send me an email. And think of John.

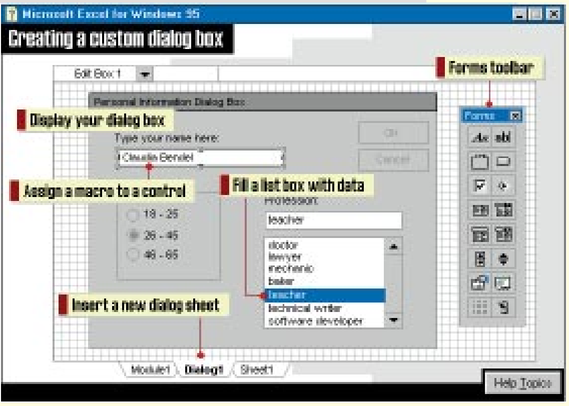


Fig 4 *(right) The comprehensive Excel help screen for creating dialogue boxes*

looks at that formula it will give you your " * " response, whether the error is caused by a number or text.

Also, there are a lot more IS functions to choose from including ISTEXT, ISNONTEXT, ISNUMBER, ISERR (which refers to any error except #N/A) and ISNA (which refers only to #N/A). You can also stack these IS functions in multiple IF statements to get what you need.

Another solution would be to use the ERROR.TYPE function. This returns a number corresponding to one of Microsoft Excel's error values.

A #NULL! Error would return a 1; #DIV/0! 2; #VALUE! 3; #REF! 4; #NAME? 5; #NUM! 6; and #N/A 7.

If you Name the appropriate cell, you can have a macro run automatically which might give you a tailored error message or take some other action. If you called the cell Account, and the macro Fixit, then you would use the following formula:

```
IF(ERROR.TYPE(Account)=6,Fixit)
```

The formula checks the cell named Account to see if it contains a #NUM! error value. If it does, the macro named Fixit is run.

Neat trick

For those readers who wish to go the other way and hide error values, here's a neat trick. Change the font colour of the cell that contains the formula to white (assuming your background colour is white). Then go back and assign a custom numeric format to the cell. Choose, Format, Cells, Custom, and type

```
[Black]General
```

in the Type box. The first action hides all entries. The second action displays entries other than error values. I know this seems odd, but it works in Excel 4, 5 and 7.

If your worksheet is more than a screenful, you can also instantly zoom out to view many more columns. In Excel 4, there are Zoom In and Zoom Out buttons on the Utility Toolbar. In Excel 7 there is a Zoom Control box, which offers several percentages, on the Standard Toolbar.

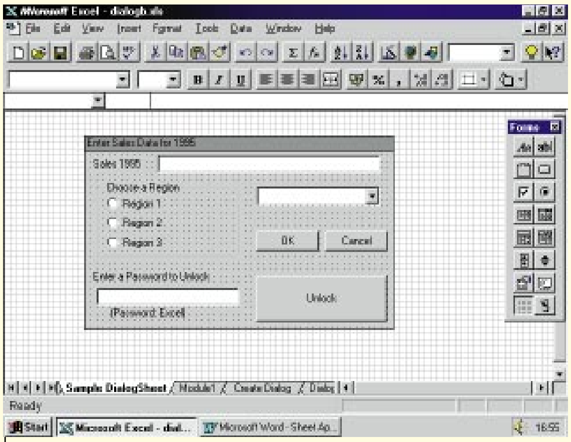


Fig 5 *The template for a dialogue box created by the Create Dialog macro*

Dialogue boxes

To add your data, you may find it more convenient to have a custom-designed dialogue box.

There are several ways of creating one, offered in Excel 7. For the first, choose Help, Index. Type "dialogue box", then pick "Creating custom". This offers the screen, shown in *Fig 4,* which you can click in five different places to open boxes of explanations.

Alternatively you can choose Insert, Macro, Dialogue and be offered a template with an OK and Cancel button in place and the Forms toolbar ready for inserting labels, buttons, list boxes, scroll bars and so on.

An even easier method would be to open up the file called Samples.xls, which should be in the C:\Msoffice\Excel\Examples directory, and copy the listing for the CreateDialog() macro. Then, in your workbook choose Insert, Macro, Module and paste it in. To run this macro, merely click within the listing and then press F5.

It will create a more complete template (*Fig 5*) than the previous option, on a sheet with a tab titled "Sample DialogSheet". You can either modify it graphically there, or change the text back in the listing which will be automatically added on another new sheet in the workbook with a tab titled Module 1.

If you haven't yet started a new workbook, you can also run this macro by opening the Samples.xls file and then choosing Tools, Macro. You'll find CreateDialog among the 30 listed available macros.

PCW *Contacts*

Stephen Wells welcomes comments on spreadsheets and solutions to be shared. Send them to *PCW* Editorial at the usual address or
Stephen_Wells@msn.com
Files can be attached if you are on MSN.