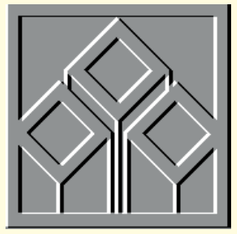

Just in **time**

Without a big computerised system, how do you keep track of your business? Stephen Wells has a suggestion. Plus, Excel's Camera gets cute.

Excess warehousing is anathema to manufacturing companies. Having raw materials, parts or finished product sitting around is not an efficient use of assets. Storing, insuring, and handling depreciating stock just adds to costs. That's why the theory of Just In Time became popular. In large companies, MRP (Materials Resource Planning) is considered a vital element of manufacturing. They want new supplies to arrive in the nick of time, but not before.

But what of small companies? They can't afford massive computerised systems. No problem, while there are resourceful managers like Steve Evans who works for a company near Pontypridd, Mid Glamorgan, that makes a highly specialised medical product. I won't tell you what it is or you'll feel queasy. But the relevant point is that the component parts are expensive and date quickly, so he's devised an Excel worksheet which tells him when to order them *(Fig 1)*.

The original covers six months and includes a long list of component parts. I've just shown a snippet or you wouldn't be able to read it.

Cell A7 has the TODAY() function. The rest of column A is for the part numbers that require tracking. Column B displays the parts descriptions.

C14 to I14 are the codes for the products to be built. C16 to I22, in this brief example, contain the number of each part contained within each of the products — effectively a parts list. Thus products A, V and O each need one of the first four listed parts.

J16 to J22 contain the current stock levels for each part. These are typed in by hand. M4 contains a date within the first month of production being monitored. The following five months run on out to the right in a similar fashion.

K6 to 12 contain the products to be built again (with a fuller description) and L6 to L12 the quantity planned to be built. This is repeated in the following periods.

K16 calculates the total number of the part on row 16 required during the period. The formula is

```
=SUM((L$6*C16)+(L$7*D16)+(L$8*E16)
+(L$9*F16)+(L$10*G16)+(L$11*H16)+
(L$12*I16))
```

The quantity to be built is given in absolute terms so the formula can be dragged down the column.

Incidentally, I can't help thinking there should be some way of multiplying named ranges as an array and simplifying that formula. If any reader has an idea on that I'd be glad to hear from them.

L16 is a simple calculation of the current stock levels minus the quantity required that period:

```
=SUM($J16-$K16)
```

The formatting is in red to show up when to order, and in this case they are short of 427 parts.

In the next column, M16 looks at the value in L16 by the function

```
=IF(L16 < 1,M$4-56," ")
```

If it is less than 1 (negative) then this part needs to be ordered. The date the order should be placed allows for a delivery time from the supplier of 8 weeks. So, 56 days are subtracted from the date in M4. If the value in L16 is greater than 1, then no date is displayed. This is repeated for the remainder of the parts for that period and subsequent periods.

Microsoft Excel - Production Plan.xls

File Edit View Insert Format Tools Data Window Help

M22 =IF(L22 < 1,M$4-56," ")

	A	B	C	D	E	F	G	H	I	J	K	L	M
4											Month 1		6/6/96
5											Prod Qty		
6	Run on										Audio	50	
7	30/3/96										2KHz Pr	25	
8											5KHz Pr	25	
9											Vert	50	
10											Obscure	400	
11											FM10	30	
12											FM60	30	
13	Parts	Description	Qty per product							Stock	Prod Qty	Stock	Date to
14	Req'd		A	2	5	V	O	10	60			left	Order
15													
16	1020-1003	Front Cover Stainless	1	0	0	1	1	0	0	73	500	-427	11/4/96
17	1020-1004	Rear Cover Stainless	1	0	0	1	1	0	0	11	500	-489	11/4/96
18	1020-1005	Blanking Plate	1	0	0	1	1	0	0	1911	500	1,411	
19	1020-1100	Pocket clip	1	0	0	1	1	0	0	436	500	-64	11/4/96
20	1020-1300	Vascular faceplate	0	0	1	1	0	0	0	631	75	556	
21	1020-1310	Vascular nose cone	0	0	1	1	0	0	0	486	75	411	
22	1020-3276	Probe Cable clamp	0	1	1	1	1	1	1	1,428	560	868	

Staffing up

The second spreadsheet from Steve Evans reminded me of a song from The Pyjama Game, the musical about factory unions: "Seven and a half cents doesn't seem a helluva lot..."

They were fighting for higher rates paid per hour. Today, the negotiation is over the number of hours worked. Fortunately for manufacturers, there are more people today who want to work part-time. In this factory, Steve tells me, about 20% of the employees do not work full time, so he also uses a spreadsheet to calculate staffing requirements *(Fig 2)*. I've had to eliminate a lot of his attractive spacing and formatting to capture all the essential parts in one screen.

Calculating the number of parts to order and when to order them for a specialised medical manufacturing company

Calculating the number of part-time workers who will be needed during specified periods in the same company

Microsoft Excel - Staffing.xls

File Edit View Insert Format Tools Data Window Help

A4

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Production Plan												
2	Run on												
3	4/2/96	and orders as at 10/06/96											
5	Product	T'ducer	PCB	Assy	Test		Period 1	June				Period 2	July
6		Time	Time	Time	Time		Qty	T'ducer	PCB	Assy	Test	Qty	T'ducer	PCB
7	OB	0.1125	0.525	1.15	0.25		200	22.5	105	230	50	450	50.625	236.3
8	VAS	0.225	1.125	1.313	0.25		0	0	0	0	0	0	0	0
11	Audio		0.9	0.625	0.25		40	0	36	25	10	25	0	22.5
13	2MHz Pr	0.1125	0.275	0.65	0.1		40	4.5	11	26	4	15	1.6875	4.125
15	4.5MHz Pr	0.225	0.225	0.51	0.1		25	5.625	5.625	12.75	2.5	0	0	0
17	6MHz Pr	0.25	0.225	0.61	0.2		0	0	0	0	0	0	0	0
18														
19	T'ducer Total							32.625					52.313	
20	PCB Total								157.6					262.9
21	Assy Total									293.8				
22	Test Total										66.5			
23														
24	Total Hrs										550.5			
25														
26	Losses													
27	Sickness									75				
28	Holidays									100				
29	Tea-breaks									90				
30														
31	People Req'd										5.4367			
32	In each section							0.7	1.5	2.4	0.9		0.7	2.2

Staffing

Setup will usually take the same amount of time regardless of how many products are required to be made, whether 1 or 1,000. Run time is directly related to how many are required; therefore, changing the batch size has a direct relationship to the productivity of the manufacturing unit.

In the block B7:E17 is recorded the time, in parts of an hour, it takes to make, assemble and test each part or sub-assembly. The quantity of each assembly needed in the first period is recorded in column F. In columns G to J are the total hours to be taken for each type of work. The same calculations for further periods continue in columns out to the right.

In rows 19 to 22 it is therefore possible to total the number of hours of work required in the period for each speciality. The total hours to be worked is given in row 24.

To this has to be added a weighting for typical sickness, holidays and the traditional tea-breaks here entered in rows 27 to 29.

The sum of all these hours is now divided by 150 in row 31. This is the figure that they use as total working hours in the month. (They must spend Friday afternoons at choir practice.)

And so it is that row 31 indicates that, arithmetically, 5.4367 people are required for these duties in this month. This is then broken down into the number of people required in each production section and displayed in cells G33 to J33. The formula in G32, for instance, is

```
=SUM((G19+(I27+I28+I29)/4)/150)
```

The system has also been used to calculate the effect of changing the batch size of various products. This has required the expansion of each of the sub-assembly times to include setup time and run time.

I've included these two spreadsheets this month not because they make any startling breakthroughs in application development, but, their very existence is indicative of the advances which have been made in the last decade. Now we have managers in small companies showing initiative and using spreadsheets to improve their efficiency.

Meanwhile, back at the hospital

Speaking of things medical, I've received a follow-up email from Dr H Baillie-Johnson whom you met in the March column. He had asked for a macro for entering and shifting data, and I gave him an auto-recorded one as it's the fastest way to program in VBA (Visual Basic for Applications). I was grateful at the time to the smart sub-editor who italicised my comment that, medically speaking, the example I gave was nonsense.

I'm even more grateful now when I find out what Dr B-J does with his spreadsheet. After praising my "elegant solution", he says he uses it to help calculate doses for ten different drug cocktails used in cancer chemotherapy. I tell you, I went cold. I didn't know whether to be honoured to have made a minuscule contribution to medical science, or scared witless in case I'd made a mistake. There are times, dear readers, when I'd rather not know what you're doing.

A reader who has concealed his application well is Mark Campbell, an NHS unit manager in Newcastle. He emailed a problem with an attached file to illustrate it.

Mark's using Excel 5. His nose is out of joint because the Apply-to-all checkbox option in Excel 4's Format, Patterns dialogue box has disappeared in version 5. He wants to be able to create a multiple-series line chart with all the markers the same size, shape and colour. Goodness knows why, but with these medical chaps I don't ask.

"I can Format, Patterns as many times as needed, but this is tedious when we often have more than 60 series," he says.

Well, what I'd do is select Tools, Record New Macro, then go through it once. Apply a shortcut key to the macro. Then, in future, you just have to double-

click the chart to select it, and press Ctrl+f or whatever you've chosen.

That's how I initially produced the following macro to do the job. Then I tightened it up by putting the main routine in a GoSub statement and substituting the variable X for the number of the series.

When viewing the module sheet, you can see all the possible colour reference numbers, from 1 to 56, by clicking on the keyword,

```
MarkerBackgroundColorIndex
```

and pressing F1. Similarly, you can see the descriptions of all the shapes available for Markers by clicking on MarkerStyle and pressing F1.

To use this macro, just choose Insert, Macro, Module then type in the listing. The new sheet will appear as the last tab but you can move it up front easily with a right-click. And you can rename it with a double-click.

The listing is shown in *Fig 3*.

Photo fit

Here's the cutest idea I've seen in a long time. I downloaded the gist of it from the Excel Forum, though I've refined the idea.

I've always known that if I made an entry into an Excel spreadsheet cell, any existing content would be deleted. Also, a function statement always has to refer to any other cell(s) than the one it is in. I was wrong on both counts.

Supposing you want a form just to display capitals. If the user enters n, for no, or y for yes, then the cell would display N or Y.

You could write a macro but it's not necessary. What you need is the Camera tool. If you can't find it, go to View, Toolbars, Customise, Utility. Then drag and drop it on to any toolbar.

Let's assume you want the entry cell to be D7. Go to some distant cell, say AZ90, and enter =UPPER(D7). Then select cell AZ90 and click the Camera button, hold down the Alt key and click on cell D7. What you're doing is creating a picture of the formula.

Now, while the object is still selected, choose Format, Object, Border - None; and Fill - None. This conceals the object.

Click elsewhere on the sheet to deselect the object, and then select cell D7 and choose Format, Font, Colour - White. This hides the new entry so if you enter any lower-case letters, they only display as capitals.

A variation is to use another formula. For instance,

```
=IF(D7="y","Y",IF(D7="n","N",D7))
```

would only capitalise those two letters and not any others. You could as easily turn an entered "n" into "NO" or any word.

Ain't misbehaving

Ask not why your spreadsheet is messing you up. Ask if you're messing up your spreadsheet.

The cursor moves to the right after you've made an entry and pressed Enter. In Excel, you can clear the Move Selection After checkbox under Tools, Options, Edit in Excel. In practically any spreadsheet, you can press the arrow key for the direction of your choice after entry. It will also finalise the entry.

Your printout is not as you intended. Always mark the Print Area before you start; and go through the many options under File, Page Set-up carefully. Check the orientation, margins, scaling, headers and footers. Also, check the Black and White option if you don't have a colour printer but are displaying colour formatting. It can speed things up.

Fig 3 Medical checkbox macro

```
Option Explicit
' Formatting Macro
' Keyboard Shortcut: Ctrl+f
Sub Formatting()
Dim X
X = 0
For X = 1 To 60
    ActiveChart.SeriesCollection(X).Select
    GoSub Routine
    Next
    Exit Sub
Routine:
       With Selection
        .MarkerBackgroundColorIndex = 5
        .MarkerForegroundColorIndex = 5
        .MarkerStyle = xlCircle
            End With
   Return
End Sub
```

Converting from 1-2-3 to Excel

In the March column, Denzyl Pereira requested a macro to make a bulk conversion of *.wks files to *.xls and frankly I wasn't that helpful. Now to the rescue comes Chris McCarthy of Birmingham, who has emailed a solution that I've tried and it works for me. The macro found the right files and did the business, as they say in some parts of the capital. Here's the listing:

```
Option Explicit

Dim FileToConvert As String
Dim FileConverted As String

Sub MultipleConvert()
ChDir "C:\windows\temp"
FileToConvert = Dir("*.wks")
While FileToConvert <> ""
    Workbooks.Open FileToConvert
    FileConverted = Left(FileToConvert, Len(FileToConvert) - 3) & "xls"
    ActiveWorkbook.SaveAs Filename:=FileConverted, FileFormat:=xlNormal
    ActiveWorkbook.Close saveChanges:=False
    Kill FileToConvert
    FileToConvert = Dir("*.wks")
Wend
MsgBox ("All *.wks files in temp directory converted")
End Sub
```

You can either adjust the instruction to change directories, or copy the files you want to convert in to that directory. I would add that the conversion of the files is done automatically when the macro opens the Lotus 1-2-3 file and uses the SaveAs function to save it in the Excel format. It works just the same as if you did it manually.

As Excel will automatically open files in numerous formats, there are many other options for you than the .wks extension; and you can save the file in a myriad formats by changing the .xlNormal extension.

PCW Contacts

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