Name dropping

Stephen Wells creates a drop-down list of names for automatic addressing, and looks at Add-Ins for Excel.

Following all the exotic applications for spreadsheets described by readers this year, it's refreshing to receive a request for a bread and butter requirement.

Andy French emails: "*I have been asked by a friend to create an invoice spreadsheet. He would like a drop-down list containing his customers' names so that he can click on a name and the address will be automatically entered. Additionally, he would like to be able to enter addresses manually, so I have also created an invoice list for him which stores the invoice details, but I would like to delete entries of a certain age and move the remaining entries up to fill the empty cells.*"

Right, then. Let us begin with the data list. Whether this is off to one side of the invoice on a separate worksheet, stored in another file, or even in another application such as Access, the principle remains the same.

For this example we will enter the company name, address, town, county, postcode, and phone number in columns A through to F respectively from Sheet 1, which we'll rename; Customers.

● The first seven customers will be on rows 7 through to 13 — mark this whole range, plus a row,

```
$A$7:$F$14
```

and name it Customerlist.

● Insert the appropriate labels (Name, Address, etc.) in cells A6 to F6.

● Also mark the range

```
$A$7:$A$14
```

and name it Company. These are the presently listed company names, plus a row.

● Now rename Sheet 2; Invoice. Name the cell L1 on that; Current.

● Refer to *Fig 1* and insert

```
=Customers!A6 in cell B4,
=Customers!B6 in cell B5
```

and so on.

The top of the invoice itself you lay out to match your other stationery. In *Fig 1* I've added a few arbitrary borders and colour patterns to show an example. In use, you would hide the row and column headers but they are shown here for explanatory purposes.

To create the pick list, first right-click on the Standard Toolbar and left-click on Forms, to display the Forms Toolbar. I'm using Excel 7 but there is nothing here which can't be recreated in Excel 4 or 5. Click the Drop-Down button on the Toolbar and drag a rectangle on your worksheet in an area like F3 to H4 (it's not critical because you can resize the box and move it later).

You now have an empty box with an arrow. Right-click on the box and choose, Format Object, Control, Input Range and enter Company. In the box below, labelled Cell Link, enter Current; and next to Drop Down Lines initially enter 3. This means when you click on the name in the box, or the arrow, three names will drop down and appear, as shown in *Fig 1*.

At this point, you now have the drop down list of customer names. The two colour panels behind it are simply Filled Rectangles produced with a click of the right button on the Drawing Toolbar.

The next job is to automatically make the correct address details appear once a name has been selected. My philosophy is never to write macro code when Excel formulas and functions will do the job. They'll always run faster than Visual Basic, anyway. Here, we can do the job with one simple function. No, not the universally loved LOOKUP, but INDEX.

In its basic reference form (there's also an array form but we don't need that here) INDEX has three essential parameters: the reference, a row number and a column number. A fourth, optional parameter is called area but, again, we don't need it here. The way in which we're using the function here is that we're saying the reference is the range covered by the name, Customerlist. The row number is dictated by cell L1 (*see above*) which we've named Current because it names the number of the current customer. The column number on the Customers worksheet is 1 if we want the company name, 2 for the address, 3 for the town, and so on up to 6 for the phone number.

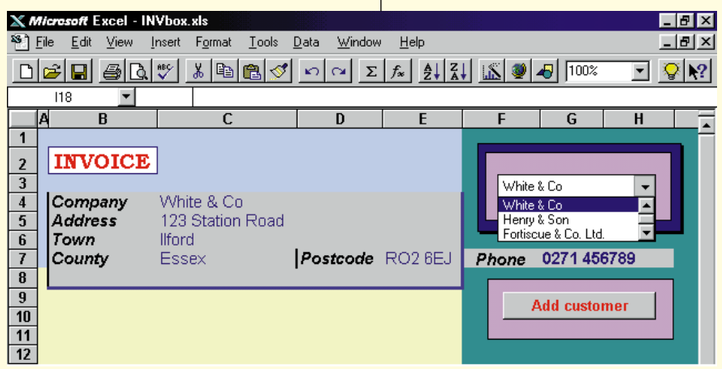


Fig 1 *On an invoice, it's easy to add a Drop-Down text box for selecting existing customers*

Right, now we're in business. To put the selected customer's company name in cell C4 we enter

```
=INDEX(Customerlist,Current,1)
```

In use, this cell simply repeats in the printing area whatever customer name is showing over in the Drop-Down box. You don't have to enter anything in cell L1. The DropDown Object will automatically put in it the number of the selected company. So if you pick the first name in the list, L1 will display 1. If you pick the sixth name in the list, L1 will show 6.

To put the address in cell C5 we enter

```
=INDEX(Customerlist,Current,2)
```

The town goes in C6 using

```
=INDEX(Customerlist,current,3)
```

and so on to

```
=INDEX(Customerlist,Current,6)
```

which puts the phone number in G7.

Once everything's working well, this worksheet can be saved as a template because this is an ordinary worksheet and no macros are involved.

If the user wants to enter an address manually, as per Andy's second request, he can simply click the Address tab, insert a row and type the details in: or choose Data, Form and a ready-labelled dialogue box would be created automatically, (*Fig 2*). It displays the same labels as in the headings row 6.

You could also add a button like the one labelled "Add customer" in *Fig 1*. It can be created automatically from the Forms Toolbar as easily as the DropDown box. If this button starts a macro with the line Worksheets(1) ShowDataForm in it, it would do the same job as turning to the Customers worksheet and choosing Data, Form.

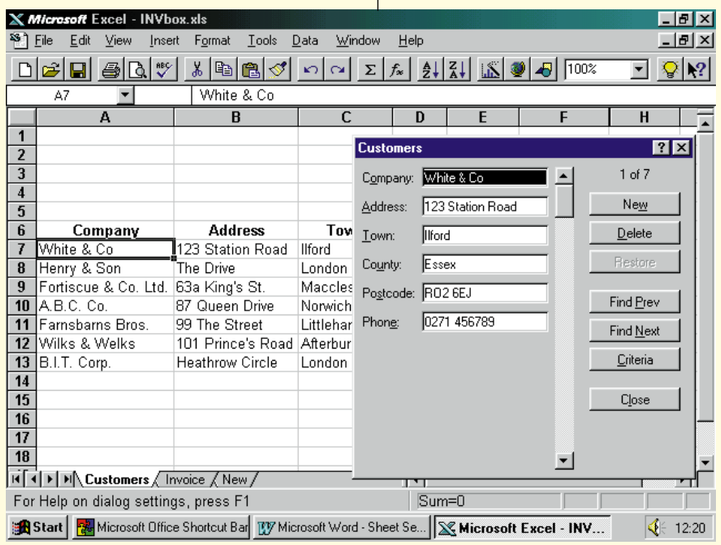
As to Andy's request for a way to clean up a list of recorded invoice details, I'd recommend Excel's AutoFilter. The list must have column labels like company, date of invoice etc. Select a cell in the list. Click Data, Filter, AutoFilter. Click the arrow in the column containing the data you want to filter and then, Custom. In the first box, click the arrow and pick the < (less than) sign. In the second box, choose a date (from the presented list of dates in the column). Once the list has been filtered, delete the rows (records) you wish to eliminate. Or, you could just print the reduced list of invoices and then redisplay the full list if you wanted to keep them on file.

Making it simple

You may recall in the June column that I wrote about a materials resource planning spreadsheet used by a Welsh medical products company. It calculated the component parts which would be needed in a period for assembling the company's products. The key formula was

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

Fig 2 *Choosing Data, Form automatically creates a dialogue box using the labels from the column headers*



I commented in passing: "I can't help thinking there should be some way of multiplying named ranges as an array and simplifying that formula."

During the few days following publication, my email box filled up as suggestions came in. For all of them, note that an array is entered by pressing [Ctrl]+[Shift]+[Enter].

One proposal was from Peter Forty who suggested assigning the name month1_prod to the range $L4:$L$12 and then using two functions in this formula:

```
=SUM(C16:I16*TRANSPOSE(month1_prod))
```

At first, I couldn't get this to work. But then I realised the arrays were of different lengths. When I trimmed the named range down to $L4:$L$12 the formula worked fine. Both arrays now referred to seven cells. If you're not familiar with TRANSPOSE, what it does is shift the orientation of an array from vertical to horizontal and vice versa. Just the job, here.

Paul Bloomfield suggested a variation that eliminates the multiplication sign:

```
=SUMPRODUCT(C16:I16,
TRANSPOSE(L$6:L$12))
```

Paul also suggested

```
=MMULT(L$6:L$12,C16:I16)
```

but I couldn't get it to work properly. Pity, really, as he told me that his name had first appeared in *PCW* in 1981. I was deeply impressed.

Then I received a lengthy email from James Talbut, in Belgium. He firstly pointed out that there was a redundancy in the original. You can either use the plus signs or the SUM function, but you don't need both — quite right. Then he suggested all of the above formulas and his version of the MMLT function worked. He wrote it as

```
=MMULT(C16:I16,L$6:L$12)
```

So did Jim Tavendale of Horndon-on-the-Hill which, if you don't know this charming village, is between Mucking and Ockendon, in Essex.

I couldn't understand why the formula worked one way around yet not the other. The Function Wizard in Excel 7 makes no distinction between the arrays. Then I checked in the indispensible Function Reference book which is Volume 3 of the Excel 4 documentation and found that the first array refers only to columns and the second array to rows.

So, grateful as I am to the other loyal and helpful readers, for accuracy, comprehensiveness and speed of response, the prize must in all fairness go to James Talbut: if you could send me your postal address, James, I'll arrange for a book token, or equivalent, for overseas winners.

Finding a data entry

Microsoft's Excel development team has introduced four new Add-Ins. They are compatible with Excel 7 and the Mac, Windows 3.x, and Windows NT 3.x versions of Excel 5. You can download them from http://www.microsoft.com/msexcel; or Go MSEXCEL on the CompuServe Information Service.

The File Conversion Wizard will be popular, considering the amount of mail I receive on the subject. There is an internet Assistant Wizard which converts Excel data into an HTML table and a Conditional Sum Wizard, which helps you create SUM-IF formulas.

But the Add-In which I suspect will appeal most of all to readers of this column is called the Lookup Wizard. What it does is to create a formula which finds the value at the intersection of a column and a row.

An example is shown in *Fig 3*. Column A has a series of date entries. Rows B through E represent regions with their labels in row 1. The block B2: E10 has simple numerical data entries. After installation, the Lookup Wizard will be found listed at the foot of the Tools menu.

In Step 1 you enter the range to search. In this case it's A1:E10. If you mark this range before starting the Wizard, the "Range to search" box will already be filled in.

Step 2 is shown in the illustration (*Fig 3*). It offers drop down lists of the row and column labels. I've selected the date which is the label for row 8 and the East region which is the label for column D.

Step 3 offers a choice: you can either copy the lookup formula to the worksheet; or you can copy the formula and the values of the lookup parameters. This allows the values of the lookup parameters to be changed on the worksheet without running the Wizard again.

If you take the first option, then in Step 4 you simply specify which cell is to contain the formula. If you plump for the second option, you still pick one cell but the Wizard will also use the next two cells for the lookup parameter values.

EXCELlent shortcuts and longshots

● **New folder:** you don't have to use Explorer or File Manager if you want to open a new folder before you save a file in Excel 7. Just choose File, Save As, and then hover the mouse over the offered buttons until "Create New Folder" appears. Click that button and enter the new folder's name. Click Save.

● **Fancy backgrounds:** you can dress up your displayed worksheets (and your hard copy, too, if you have a colour printer) by choosing Format, Sheet, Background. Select any directory with graphics files in it and pick your preference. Such backgrounds look best if you also choose Tools, Options and empty the Gridlines box under the View tab.

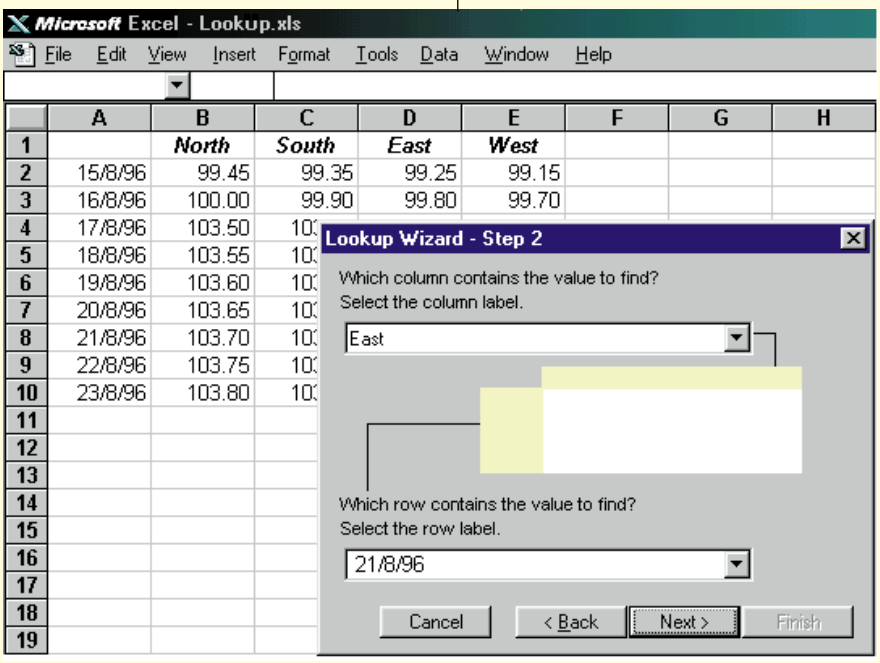
● **Absolutely:** if you press the F4 key before ENTER when entering a relative cell reference it will change automatically to an absolute reference. Example: C4 becomes $C$4.

● **Save memory and disk space:** by initially opening your workbooks with fewer sheets. Choose Tools, Options, General, "Sheets in New Workbook". You can easily add sheets as you need them by right-clicking on a sheet tab and selecting Insert, Worksheet.

● **Writing macros:** as easily as you can insert functions on a worksheet using the Function Wizard, you can insert object names on a module sheet. Just click the Object Browser button on the Visual Basic Toolbar. Then select Excel under "Libraries/Workbooks". Following the selection of an object in the "Objects/Modules" list box you can view all the properties and methods for that object in an adjacent list box. Choose one and click the Paste button — it's immediately copied into your macro.

● **Imported data:** can easily be divided into columns using the Text to Columns Wizard. Whether the data is separated by commas, semicolons, tabs or any other delimiter which you specify, Excel will cut it up into columns for you. Select the cells to convert. Choose Data, Text To Columns. Specify how you want the text divided into columns.

Fig 3 *The new Lookup Wizard Add-In will find a value at the intersection of a column and row*



The Wizard wigs out

Actually, this is how it's *supposed* to work but in fact, after Step 2, I couldn't get past an error message box that stated: "The cell found by the Wizard contains a formula returning an error. Please exit the Wizard and fix your table." The fact is, my table didn't contain any formulas so what the Wizard's problem was, I do not know. Hopefully, by the time you read this and have downloaded the Add-In, Microsoft will have had the bug sorted out.

However, all the Wizard does with choice 1 is to write a formula which you can enter yourself in the cell of your choice. First, make three Names. Call A1:E1 Columns. Call A1:A10 Rows. Call A1:E10 Table. To reproduce the value for East on 21/8/96 in, say, cell F15 then in that cell you enter this formula:

```
=INDEX(Table,MATCH(DATEVALUE
("21/8/96"),Rows,),MATCH("East",
Columns,))
```

And that should solve the problem.

PCW Contacts

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