**Counting Cells Containing a Formula**

There is a way to count the number of cells containing formulas in a row or column. The answer is quite simple, using the Go To feature of Excel. Follow these steps:

1. Display the worksheet for which you want a count.
2. Select the row or column in which you want to count formulas.
3. Press **F5** or **Ctrl+G**. Excel displays the Go To dialog box.
4. Click the Special button. Excel displays the Go To Special dialog box. (See Figure 1.)



**Figure 1.** The Go To Special dialog box.

1. Make sure the Formulas radio button is selected.
2. Click OK.

That's it. Excel selects all the cells in the row or column that contain formulas. (If you skip step 2, Excel selects all the formulas in the entire worksheet.) At the bottom of the screen, in the status bar, you can see a count of the number of cells selected. (See Figure 2.)



**Figure 2.** The status bar shows a count of selected cells.

If, for some reason, you don't see a count in the status bar, you should check to make sure you have your status bar configured to show counts. Just right-click any blank spot on the status bar and choose Count form the resulting options.

You could also use a formula to figure out how many formulas are in a range of cells, as shown here:

=SUMPRODUCT(--ISFORMULA(A:A))

This example returns the count of all the formulas in column A; you could just as easily substitute a different range of cells in the formula. Whatever range you specify, it should not include the cell where you place this particular formula—that would result in a circular reference and a probable error.

You could also, if desired, use a macro to determine the count. The following example uses the same approach to determining a count as was manually described in the earlier steps:

Sub CountFormulas1()

 Dim Source As Range

 Dim iCount As Integer

 Set Source = ActiveSheet.Range("A:A")

 iCount = Source.SpecialCells(xlCellTypeFormulas, 23).Count

 ActiveSheet.Range("D1") = iCount

End Sub

This subroutine returns, very quickly, a count of all the formula-containing cells in column A and stuffs that value into cell D1.

It would be very helpful if this approach could be turned into a user-defined function, such as this:

Function CountFormulas2(Source As Range)

 CountFormulas2 = Source.SpecialCells(xlCellTypeFormulas, 23).Count

End Function

This won't work, however. The function always returns the count of the cells in the Source range, not the count of the cells containing formulas. It is an esoteric bug in Excel's VBA that SpecialCells doesn't work in functions; it only works in subroutines. Microsoft hasn't even documented this one (that I can find), thus my reference to it as a "bug" instead of as a "limitation."

There is an actual limitation to what the SpecialCells method can do, however: It can only contain a range of up to 8,192 cells. You can analyze a range that is much larger (as is the case when you have it look at an entire column), but the resulting subset—the resulting range—can only contain up to 8,192 cells. If it contains more, then SpecialCells will "fail" and return a range (and therefore a count) that is equal to the number of cells in the original range.

If you want to create a user-defined function to determine the count, you'll need to rely on something other than the SpecialCells method. Here's an approach that uses the HasFormula property:

Function CountFormulas3(Source As Range)

 Dim c As Range

 Dim iCount As Integer

 iCount = 0

 For Each c In Source

 If c.HasFormula Then iCount = iCount + 1

 Next

 CountFormulas3 = iCount

End Function

If you choose to have this macro evaluate an entire column or an entire row, then be prepared to wait a bit—it can take a while for the macro to step through each cell in a column or row. The SpecialCells method is much faster at deriving results than stepping through each cell.