

EXCEL



By Donald Reynolds, CMA, CPA

Printing Multiple Forms from a List

Setting up and printing a form in Excel typically isn't a difficult task. When updating and printing a form with a large set of data, however, the process becomes tedious and time consuming. By creating a print macro with a data update loop, this process can be done quickly and error free.

In the following example, a Materials Requisition Form needs to be updated and printed for a list of sales orders that need to be produced in a job order system. Figure 1 shows the Materials Requisition Form and the Order List.

The worksheet in Figure 1 is designed to update the part number, usage rate, and requisition quantity (cells B9:D13) on the Material Requisition Form using lookup values based on the item number and quantity input into cells C4 and C5. The Order List includes the data needed for cells C4 and C5 in the Materials Requisition Form as well as the order number in cell C3.

In order to accomplish the update and print task, the total requisitions to print was added in cell H22 using the COUNTA function: =COUNTA(H5:H21). This counts the number of cells that aren't empty in a specified range. We'll

Figure 1

Materials Requisition Form				Order List		
Order Number	SO13-0038			Order Number	Item	Quantity
Item	A1 - Standard			SO13-0034	D4 - Ultra	4
Quantity	3			SO13-0036	E5 - Gold	2
				SO13-0038	A1 - Standard	3
				SO13-0040	F6 - Platinum	2
				SO13-0042	E5 - Gold	4
Part Number	Usage Rate	Requisition Quantity		SO13-0044	C3 - Superior	1
1 X-3047-C	3	9		SO13-0046	B2 - Premium	5
2 V-5469-MM	5	15		SO13-0048	D4 - Ultra	1
3 G-0031-G	1	3				
4 H-8264-C1	2	6				
5	0	0				
Issued By _____						
Received By _____						
Total Requisitions to Print						8

use it as a variable to determine how many requisitions to print.

Writing the macro code for this

process can be started by recording a simple print macro. Before you begin, however, be sure that your file is saved

Figure 2

Sub Macro1()

ActiveSheet.PageSetup.PrintArea = "\$A\$1:\$D\$18"

ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True, _
IgnorePrintAreas:=False

EndSub

Figure 3

```
Sub Macro1()
```

```
Dim PageCount As Integer
```

```
For PageCount = 5 To Range("H22")+4
```

```
Range("C3") = Cells(PageCount, 6)
```

```
Range("C4") = Cells(PageCount, 7)
```

```
Range("C5") = Cells(PageCount, 8)
```

```
ActiveSheet.PageSetup.PrintArea = "$A$1:$D$18"
```

```
ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True, _
```

```
IgnorePrintAreas:=False
```

```
Next PageCount
```

```
End Sub
```

A loop counter to track the number of pages printed

Using the cell "row/column" format and the PageCount variable, this updates the Materials Requisition Form from the data in the order list.

Adds a value of 1 to the page counter and loops back to the "For" line

with an XLSM, XLSB, or XLS extension, as the default XLSX extension won't allow macros to be saved in the file.

To record the print macro, begin by selecting the first cell in the form. In this example, it's cell A1. From the View tab on the ribbon, click on the Macros dropdown and select Record Macro. The Record Macro dialog box will appear. Make note of the macro name (typically Macro1), and click OK. Then go through the normal steps to select and print the form. After the form prints, go to View, Macros, Stop Recording.

To see the list of macros, press ALT+F8. Select Macro1, and press the Edit button. The Microsoft Visual Basic for Applications window will open, showing you the code for the macro. It should be similar to Figure 2 (depending on the steps you used to select and print the form, it may differ slightly).

To complete the update and print loop, we need to make a few changes to the code. There are a number of different variable types and ways to complete a looping process. Figure 3 shows one solution using a "For/Next" loop. The text in blue represents the updates to the code.

In this example, one variable is added—declared using a Dim statement—to keep track of the row of data to use from the Order List: Dim PageCount As Integer. Then the loop process is added. The first record to be printed is in row 5, so the loop starts at 5: For PageCount = 5. The loop should go to the row that is four more than the calculated row count in H22: To Range ("H22")+4. Then three lines are added to update the Materials Requisition Form from the data in the Order List.

Once these changes are saved, this

macro will now update and print a Material Requisition Form for each sales order listed on the Order List. This macro can be attached to a shape or form button to allow the code to be run with one click of the mouse. On the Insert tab, click on Shapes and select a rectangle. Draw the shape on the worksheet, then add text such as "Print All." Right-click the shape, select Assign Macro, and select Macro1. **SF**

Donald Reynolds, CMA, CPA, DBA, is chief financial officer at Festida Foods, Inc. He is also a member of IMA's Grand Rapids Chapter. You can contact him at donsalt@hotmail.com.

Bill Jelen is the author of Excel 2013 In Depth and 39 other books. Send questions for future articles to IMA@MrExcel.com.