HOW TO BUILD A DOCUMENT REPOSITORY WITH EXCEL

Michael Brothers

In 2016 I built a document repository for a local manufacturing client. The company produces bioreactor assemblies made from a lot of plastic tubing and parts. On a regular basis customers would request a certificate of animal origin, allergens, etc. in the material composition, which necessarily traced back to supplier components. My role was to collect all of the supplier certificates in a central server location, and make them easily retrievable.

Imagine a company that makes fabric dolls. They make all different shapes and sizes of doll, but the raw materials are typically fabric, yarn, thread, stuffing and buttons. So we have five suppliers (FabricCo, YarnCo, etc.) and each supplier offers dozens of variations of their product. Each doll that the company produces will have a Bill of Materials specific to that doll – a particular cut of fabric, a certain color yarn, etc.

When a customer orders a doll, they ask if the doll contains peanuts or latex. Another customer asks about bovine tallow. A third has carbon footprint concerns. In response, our company develops a Material of Composition Certificate (we'll call it a MOCC) that they send to all their suppliers. The MOCC asks a set of questions that address the dozen or so most common customer concerns. Over the years, the MOCC has changed to address newer demands. So in our file system we have three versions of the MOCC form; MOCC1, MOCC2 and MOCC3. They are similar except that the order of the questions is different in each form, and MOCC3 asks two new questions that are not on MOCC1 or MOCC2.

So how do we build a system that allows us to take a doll's Bill of Materials, and automatically answer questions based on supplier MOCC forms? The following sections outline my solution to the problem.

Step 1: File Folders

Each supplier is assigned a folder on the network drive. Supplier folders hold three subfolders; “Forms Sent”, “Forms Current” and “Forms Outdated”. In the data collection phase of the project, all supplier forms are scanned as pdf files and put into the appropriate folder.

- Suppliers
  - FabricCo
    - Forms Current
      - GGparts signed.pdf
    - Forms Outdated
    - Forms Sent
      - GGparts.pdf
  - ThreadCo
  - YarnCo

Step 2: Part Number Worksheet

The next step is to create a spreadsheet to hold individual components, each with a unique part number:

	A	B	C	D
1				
2				
3				
4	**Part Number**	**Supplier**	**Description**	**Material**
5	AB1234	FabricCo	Sunflower pattern, 12" x 12"	linen
6	CD5678	FabricCo	Solid blue, 15" x 18"	terry
7	GG1468	YarnCo	White, light worsted, 4mm	merino wool
8				

The first three rows are intentionally left blank – this will help address the different MOCC versions later on.

Step 3: Hyperlinks

Assume we have MOCC forms for every part. The easy solution is to add a hyperlink to the part number cell that points to the proper file. DON’T! I found out the hard way that hyperlinks (like cell borders) don’t always travel when the spreadsheet is sorted. Instead, use the HYPERLINK function:

```
=HYPERLINK(“\\filepath\filename”, “text to appear in cell”)
```

This provides a permanent link, and some new options.

For example, in my spreadsheet I used the hyperlink to indicate which version of a form we had on file.

If the MOCC for YarnCo’s merino wool products is version 2, I might have

```
=HYPERLINK(“\YarnCo\Forms%20Received\GGparts%20signed.pdf”,“2”)
```

Here, %20 stands in place of a space character – depending on your operating system you may not need it.

This way, I can easily select all parts with older MOCC versions and send requests to the supplier for updated forms.

Step 4: Questionnaire Fields

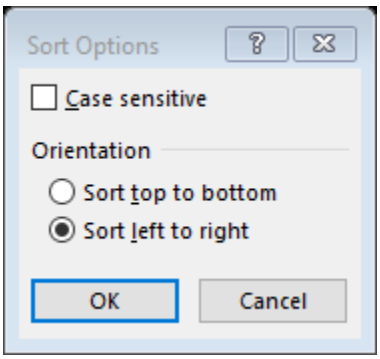
Next, create columns for each question. Then map the questionnaire versions to the questions using the three rows reserved at the top of the sheet.

	A	B	C	D	E	F	G	H	I	J
1					*MOCC1:*	1	3	2		
2					*MOCC2:*	2	1	3		
3					*MOCC3:*	1	2	3	4	5
4	**Part Number**	**Supplier**	**Description**	**Material**	**MOCC**	**Peanuts**	**Latex**	**Tallow**	**Vegetable**	**Green**
5	AB1234	FabricCo	Sunflower pattern, 12" x 12"	linen	3	Yes	No	No	No	No
6	CD5678	FabricCo	Solid blue, 15" x 18"	terry	3	No	Yes	No	No	Yes
7	GG1468	YarnCo	White, light worsted, 4mm	merino wool	2	No	No	Yes	NA	NA
8										

Here, column E contains links to MOCC files, with text indicating which version is on file.

Columns F through J hold answers to survey questions. Note that right now, these columns are sorted by row 3 – that is, they appear in MOCC3 order. Before entering a batch of MOCC2 forms, we could sort by Row 2:

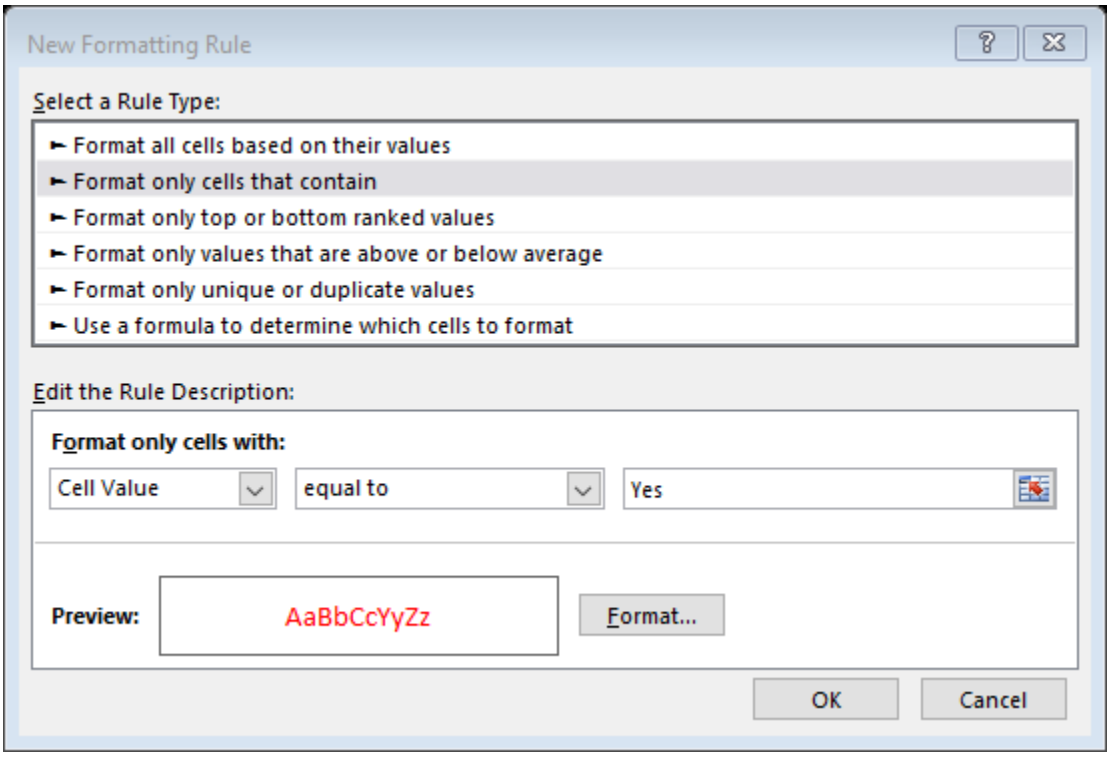
Select columns F:H, open Custom Sort, unselect “My data has headers”, choose Options, select “Sort left to right”:



	A	B	C	D	E	F	G	H	I	J
1					*MOCC1:*	3	1	2		
2					*MOCC2:*	1	2	3		
3					*MOCC3:*	2	1	3	4	5
4	**Part Number**	**Supplier**	**Description**	**Material**	**MOCC**	**Latex**	**Peanuts**	**Tallow**	**Vegetable**	**Green**
5	AB1234	FabricCo	Sunflower pattern, 12" x 12"	linen	3	No	Yes	No	No	No
6	CD5678	FabricCo	Solid blue, 15" x 18"	terry	3	Yes	No	No	No	Yes
7	GG1468	YarnCo	White, light worsted, 4mm	merino wool	2	No	No	Yes	NA	NA
8										

Step 5: Conditional Formatting

Take advantage of conditional formatting to highlight "Yes" responses in certain categories:



	A	B	C	D	E	F	G	H	I	J
1					MOCC1:	1	3	2		
2					MOCC2:	2	1	3		
3					MOCC3:	1	2	3	4	5
4	**Part Number**	**Supplier**	**Description**	**Material**	**MOCC**	**Peanuts**	**Latex**	**Tallow**	**Vegetable**	**Green**
5	AB1234	FabricCo	Sunflower pattern, 12" x 12"	linen	3	Yes	No	No	No	No
6	CD5678	FabricCo	Solid blue, 15" x 18"	terry	3	No	Yes	No	No	Yes
7	GG1468	YarnCo	White, light worsted, 4mm	merino wool	2	No	No	Yes	NA	NA
8										

Note: this is purely cosmetic, and does not affect any of the values.

Step 6: Performing Lookups

Assume that our spreadsheet now contains dozens of suppliers and thousands of parts. It's time to put them to work!

Open a new tab, call it Part Number Lookup, and create columns for the categories you care about.
In the project I completed for the biotech firm, the Bill of Materials was copied from a manufacturing drawing in pdf. As such, the quantity, description and part number all appeared on one line. Fortunately, the part number was always the last word of text. The following line of code let me strip just the part number into its own cell:

```
=RIGHT(A2,LEN(A2)-FIND("|",SUBSTITUTE(A2," ","|",LEN(A2)-LEN(SUBSTITUTE(A2," ",""))))))
```

This counts all the spaces in the string, turns the last one into a vertical pipe, and then grabs text to the right of the pipe.

B2 =RIGHT(A2,LEN(A2)-FIND("|",SUBSTITUTE(A2," ","|",LEN(A2)-LEN(SUBSTITUTE(A2," ",""))))))

	A	B	C	D	E	F	G	H
1	**Paste the Bill of Materials here:**	**Part Number**	**Peanuts**	**Latex**	**Tallow**			
2	8ft White, light worsted yarn, 4mm GG1468	GG1468						
3	2 Fabric swatches CD6250							
4	3 brown upholsterd buttons 65Y123							
5	1 spool denim thread DT876543A							

Using the VLOOKUP function or INDEX/MATCH is your choice. If using VLOOKUP, be sure to add the FALSE argument, while MATCH should include the 0 argument. These arguments prevent false returns if a part number is not listed, and don't require that part numbers be sorted. I prefer INDEX/MATCH:

C2 =INDEX(Sheet1!F:F,MATCH($B2,Sheet1!$A:$A,0))

	A	B	C	D	E	F
1	**Paste the Bill of Materials here:**	**Part Number**	**Peanuts**	**Latex**	**Tallow**	
2	8ft White, light worsted yarn, 4mm GG1468	GG1468	No	No	Yes	
3	2 Fabric swatches CD6250					
4	3 brown upholsterd buttons 65Y123					
5	1 spool denim thread DT876543A					

Sheet1 Column F corresponds to the Peanuts question. *Take care not to rearrange the Sheet1 columns!*
The lookup formula will grab data from column F no matter which question is assigned there.
I got around this by setting aside a small group of columns over to the right of the questionnaire columns. These I devoted to key questions like peanuts and latex. While it added a bit to data entry, it allowed me to sort the questionnaire columns freely. It also gave me a place to record data obtained from sources besides the questionnaire.

Another workaround would be to copy the INDEX/MATCH formula into cell C1 – that is, to pull the column heading from Sheet1 Column F by matching on the phrase "Part Number".

Step 7: Simplify Matters with a Macro

As it stands, performing a lookup requires first pasting the Bill of Materials, then selecting the formula rows from somewhere and finally pasting them into the appropriate spot. This can all be done in one step with a VBA macro.

First, decide where you would like to store the lookup formulas. They should be in a place that is accessible to the macro, and unlikely to be deleted by the user. I kept mine on a separate tab, but here I'll put them into the first row of the lookup tab. After making sure the macro works properly, I'd hide the row from view.

Hit Alt-F11 to open the VB editor in Excel, open a new module, and type the following code:

```
Sub PasteBoM()
' First make sure the ActiveCell is in the right column
   If ActiveCell.column <> 1 Then
      MsgBox("Please select a cell in Column A and try again."
      Exit Sub
   End If
' Then make sure the neighborhood 100 rows down and 4 columns right is clear
   If WorksheetFunction.CountA(Range(ActiveCell, _
   ActiveCell.Offset(100,4))) > 0 Then
      Box = MsgBox("You might be pasting new data on top of old data. _
      Do you wish to continue?", vbYesNo, "DANGER: Potential Loss of Data!")
      Select Case box
         Case vbNo
         Exit Sub
      End Select
   End If
' Paste the BoM from the clipboard, then the formulas
   Selection.PasteSpecial (xlPasteAll)
   Range("b1:e1").Copy Selection.Offset(0,1)
End Sub
```

I assigned "Ctrl-j" to the macro. It should be noted that macros cannot be reversed – that is, Ctrl-z doesn't work. That is why I included the neighborhood check to ensure data is not unintentionally deleted.

Conclusion

That's it! Now we have an orderly filing system, a parts library that points to relevant files, and a means to look up meaningful data from a Bill of Materials.

One final note. In the Lookup section, INDEX/MATCH brought over the cell value – that is, the "Yes" or "No".
If I'd wanted to see the file path instead, I would have to look inside the HYPERLINK function.
Say that the parts spreadsheet had the following entry in column E for part number GG1468:

```
=HYPERLINK("Suppliers\FabricCo\Forms Current\GGparts signed.pdf","2")
```

The file path can be obtained with two formulas:

```
=FORMULATEXT(INDEX(Sheet1!E:E,MATCH($B2,Sheet1!$A:$A,0)))
```

Returns

=HYPERLINK("Suppliers\FabricCo\Forms Current\GGparts signed.pdf","2")

So then

```
=MID(F2,13,LEN(F2)-18)
```

Returns

Suppliers\FabricCo\Forms Current\GGparts signed.pdf

Voila!