

Working With Metadata in Spreadsheets; Tips and Tricks for Excel Users

Copyright© 2008 Wayne Torborg, Director of Digital Collections and Imaging, Hill Museum & Manuscript Library

Many users of CONTENTdm are using external programs such as Microsoft Excel to create metadata spreadsheets. These spreadsheet tables can be converted into tab-delimited ASCII text files which are then used to load objects and metadata as batch operations. This speeds up the creation of digital collections considerably and also allows others to create metadata sets independently, for later import by CONTENTdm administrators.

An important aspect of this is that quite often users already have some or all of their metadata in some sort of database format. Often the data resides in a legacy database system and needs to be re-purposed for CONTENTdm. In these cases, knowing how to reformat existing data without retyping or starting over with data entry can save the user a great deal of time, trouble, and potential errors.

This is a "cheat sheet" containing some of the coding and Excel formula work explained during the workshop titled, "Advanced Data Wrangling" presented at the 2008 Upper Midwest CONTENTdm User Group Meeting. It is intended only as a starting point for users interested in learning some of Excel's text and number functions, and isn't intended to be a complete course in the use of Excel. Using the internet, users can find out others ways to reformat data using Excel.

Combining Data from Two or More Cells into One Cell

Sometimes a user needs to combine the contents of several cells in a spreadsheet into one cell. This can happen if data coming from a legacy database is broken down into discrete categories and needs to be combined to create the more general type of data used in Dublin Core data schemes.

In other cases, the user may want to add some sort of repeating data onto an existing field contents. In HMML's case, this often means taking an image barcode number and adding the file extension, ".jpg" to it to create the field for the object's filename. Here's the Excel function code to do these things:

```
=CONCATENATE ( A2 , B2 )
```

creates field data based on combining contents of cells A2 and B2. Alternatively, if both fields are text entries, the user can simplify the syntax to:

```
=A2&B2
```

There are variations on this them that allow the user to customize the resulting data to a

greater extent:

```
=A2& ". " &B2
```

This would take the contents of cell A2, add a period and a space, and then add the contents of cell B2. NOTE: straight "typewriter" style quote marks are used instead of the "curly" typographically-correct ones.

This formula code:

```
=A2& "<br>" &B2
```

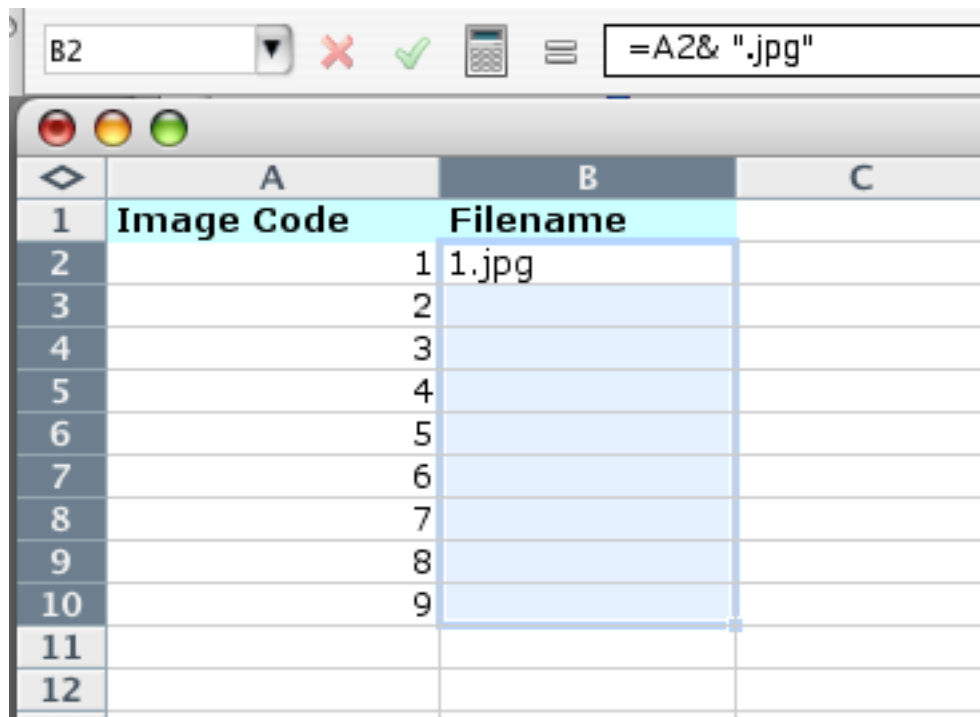
combines the cell contents of A2 and B2 but inserts an HTML break tag in between to create a line break when CONTENTdm displays the metadata online.

This formula:

```
=G2& ".jpg"
```

adds the file extension ".jpg" to the contents of cell G2.

With any of these functions, the user can use the "fill down" action to automatically populate cells with the information from their respective cell data sources.

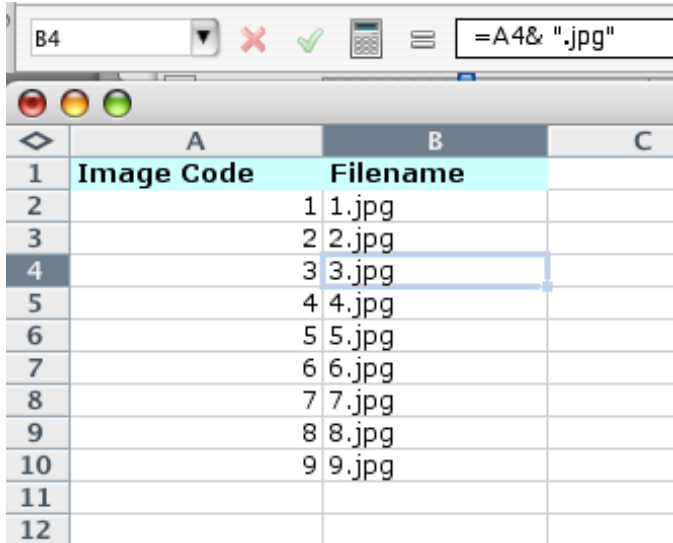


The screenshot shows a spreadsheet application window. The formula bar at the top displays the formula `=A2& ".jpg"`. Below the formula bar is a table with three columns: A, B, and C. Column A is labeled 'Image Code' and column B is labeled 'Filename'. The table contains the following data:

	A	B	C
1	Image Code	Filename	
2		1	1.jpg
3		2	
4		3	
5		4	
6		5	
7		6	
8		7	
9		8	
10		9	
11			
12			

Here, the formula was entered as `=A2& ".jpg"` in cell B2. Then, the selection was extended to cell B10, either by holding the shift key while pressing the "down" arrow key or dragging the selection with the mouse. Choosing the "fill down" action from the "edit"

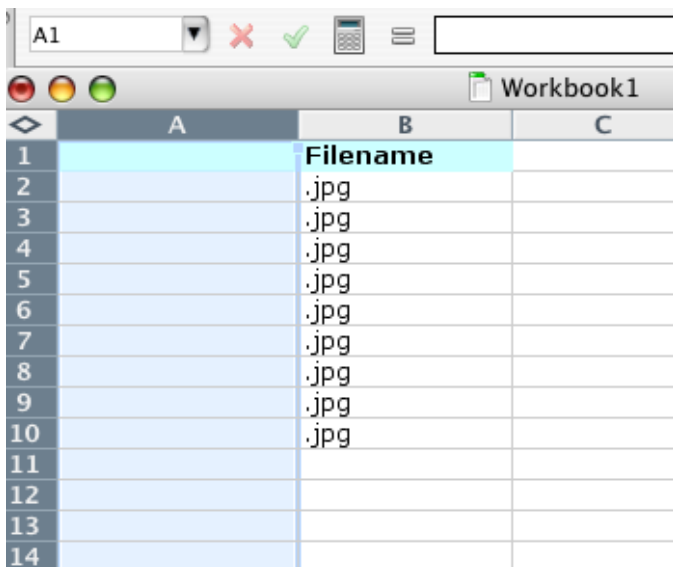
menu fills the formula down, but it's intelligent enough to rewrite the code so that each row "harvests" the source data that corresponds to it:



	A	B	C
1	Image Code	Filename	
2		1 1.jpg	
3		2 2.jpg	
4		3 3.jpg	
5		4 4.jpg	
6		5 5.jpg	
7		6 6.jpg	
8		7 7.jpg	
9		8 8.jpg	
10		9 9.jpg	
11			
12			

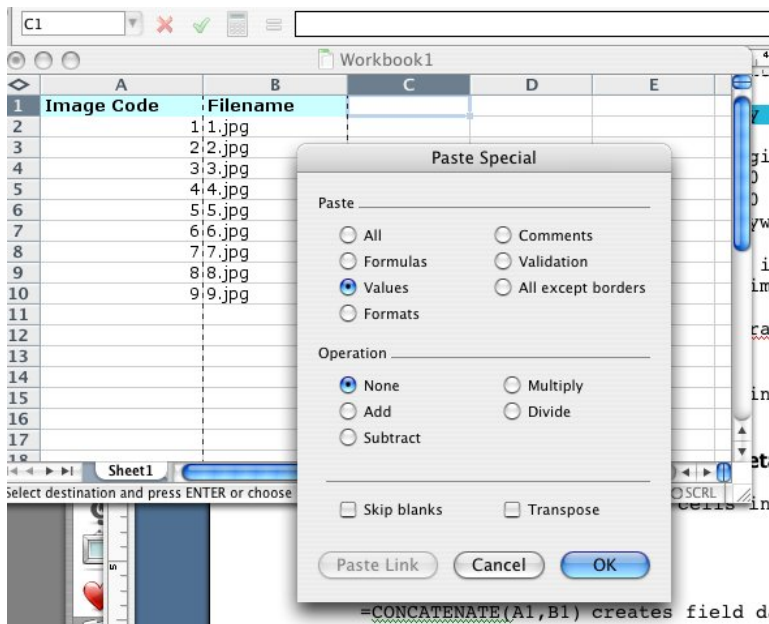
Here it can be seen that although the formula was filled down from the row B2, the formula was changed on row 4 to take the cell contents from A4 to create the new cell contents. This is extremely handy in quickly creating columns of data.

If we intend to keep the "image code" field in our final metadata sheet, this is all the user must do. If the Image Code field is to be removed, a further step must be made. This is because the contents of column B *depend* on what's in column A, and removing the contents of column A will "break" the results in column B as such:

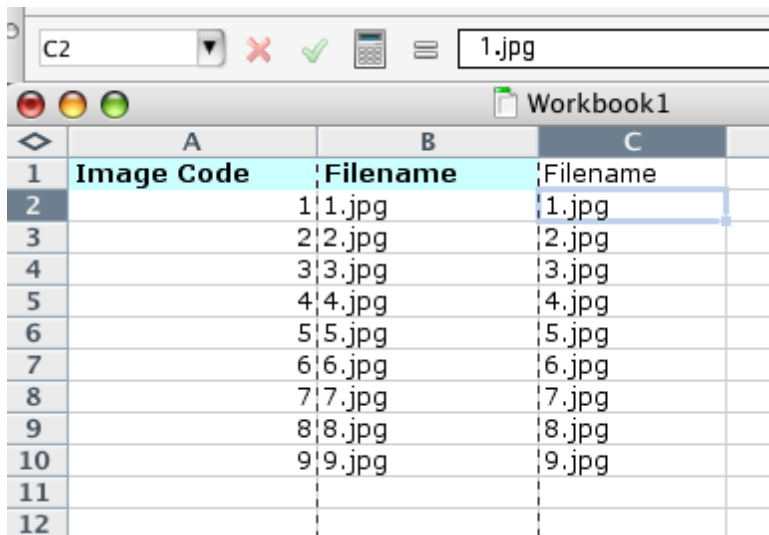


	A	B	C
1		Filename	
2		.jpg	
3		.jpg	
4		.jpg	
5		.jpg	
6		.jpg	
7		.jpg	
8		.jpg	
9		.jpg	
10		.jpg	
11			
12			
13			
14			

With nothing in column A, the formula simply returns ".jpg" as it is written to do. The solution is to copy the formula column and paste it as "values only" in an adjacent column *before* getting rid of the source column A:



This replaces the formula code with the actual result of the expression as such:



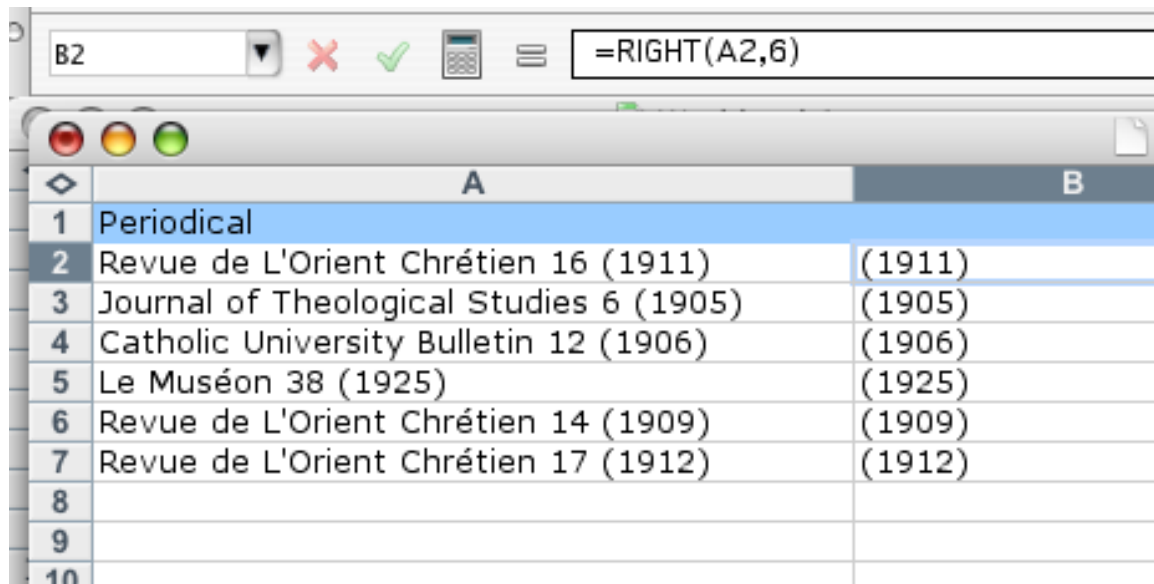
Notice that instead of the formula appearing in the formula bar, the content of the cell C2 is actually what the *value returned by the formula in B2 is*. This breaks the tie between the contents of the cell and its source cells, and the source cells can be deleted if they are no longer needed.

This technique is important. By using the copy and "paste special/values" method, the user can take any complicated Excel formula and convert it into the actual value that the particular formula creates. This can be applied to all the Excel functions outlined in this paper and the workshop session.

Breaking Excel Cell Contents Apart

Sometimes it's helpful to know how to break things apart instead of putting them together. There are a variety of Excel functions that can do this as well.

For example, I was given a spreadsheet with a data field for "Periodical." This data field had the name of the periodical followed by the date in parenthesis. For Dublin Core data mapping this isn't useful; "Date" is a field by itself in standard Dublin Core. How to break apart the periodical from the date? Here's one way:



The screenshot shows an Excel spreadsheet with the following data:

	A	B
1	Periodical	
2	Revue de L'Orient Chrétien 16 (1911)	(1911)
3	Journal of Theological Studies 6 (1905)	(1905)
4	Catholic University Bulletin 12 (1906)	(1906)
5	Le Muséon 38 (1925)	(1925)
6	Revue de L'Orient Chrétien 14 (1909)	(1909)
7	Revue de L'Orient Chrétien 17 (1912)	(1912)
8		
9		
10		

Since the date is at the far right of the data in column A, the user can simply grab it with the formula:

```
=RIGHT(A2,6)
```

which takes six characters from the right side of the contents of cell A2 and displays them. Again, the user would then copy column B, paste special into column C as "values" and could then discard the column with the formula. Then the new column could be selected and a "search and replace" function be done to replace the parenthesis with nothing—this would get rid of them, leaving the user with a clean date field for CONTENTdm.

But now what about the "Periodical" field? The date is still there as redundant information. Look at the formula bar in the next screenshot to see how this could be remedied:

	A	B	C
1	Periodical		Date
2	Revue de L'Orient Chrétien 16 (1911)	Revue de L'Orient Chrétien 16	1911
3	Journal of Theological Studies 6 (1905)	Journal of Theological Studies 6	1905
4	Catholic University Bulletin 12 (1906)	Catholic University Bulletin 12	1906
5	Le Muséon 38 (1925)	Le Muséon 38	1925
6	Revue de L'Orient Chrétien 14 (1909)	Revue de L'Orient Chrétien 14	1909
7	Revue de L'Orient Chrétien 17 (1912)	Revue de L'Orient Chrétien 17	1912
8			
9			
10			

Let's parse this formula a bit:

`=LEFT(A2,LEN(A2)-7)`

What this syntax does is harvest, from the LEFT of cell A2 the number of characters of the LENGTH of A2 minus 7, which is the number of characters that the date with parenthesis and the leading space take up. Again, the user must create a new, blank, column and "copy-paste special-values" the cell contents to replace the formula with the actual data values. Then the source columns can be deleted if need be.

	A	B	C
1	Periodical	Date	
2	Revue de L'Orient Chrétien 16	1911	
3	Journal of Theological Studies 6	1905	
4	Catholic University Bulletin 12	1906	
5	Le Muséon 38	1925	
6	Revue de L'Orient Chrétien 14	1909	
7	Revue de L'Orient Chrétien 17	1912	
8			
9			
10			

Here's the final result. All this formula work may seem like overkill in this example that only has six records, but it really helps if you have hundreds of records to work on!

Number Formatting

Padding Numbers with Leading Zeroes

Working with numbers and their formatting can be confusing and can lead to problems. One thing that many folks have encountered is the improper sorting of numbers in database programs where the number is actually stored as a text value.

A solution that HMML uses is to "pad" the numbers to a fixed number of digits. We use a five-digit system, so the number:

25

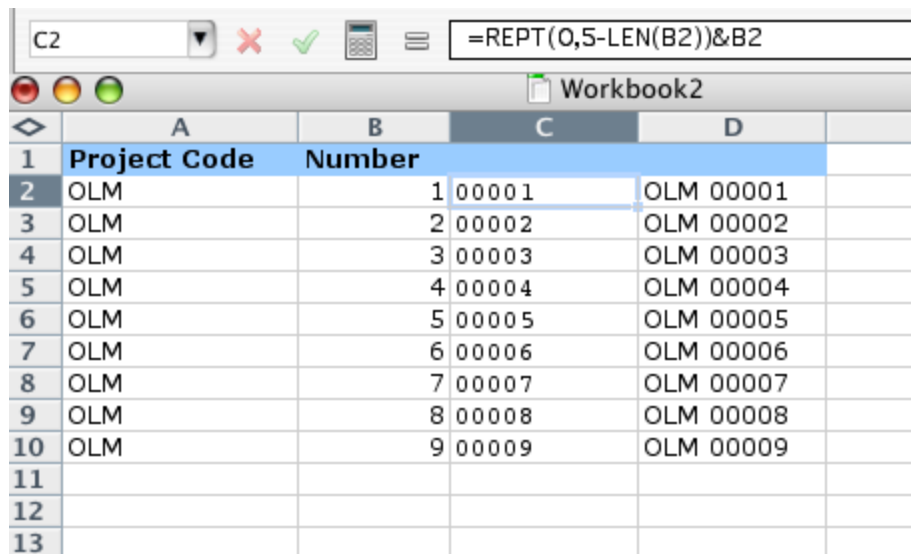
is generally expressed as:

00025

We have a data field titled "Source" that indicates a particular manuscript by using a text prefix combined with this five-digit number, such as:

OLM 00025

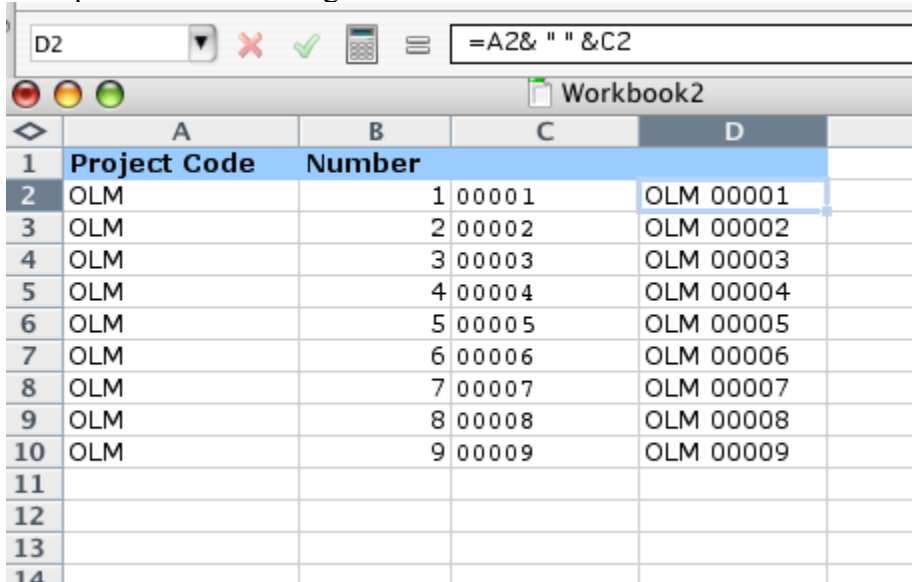
In Excel, the user can write a formula that will pad out a regular number to five digits. Here's an example of using this to create a properly formed HMML source number:



	A	B	C	D
1	Project Code	Number		
2	OLM	1	00001	OLM 00001
3	OLM	2	00002	OLM 00002
4	OLM	3	00003	OLM 00003
5	OLM	4	00004	OLM 00004
6	OLM	5	00005	OLM 00005
7	OLM	6	00006	OLM 00006
8	OLM	7	00007	OLM 00007
9	OLM	8	00008	OLM 00008
10	OLM	9	00009	OLM 00009
11				
12				
13				

Looking at the formula, one can see that the function is set to repeat (REPT) the character "0" five times minus the LENGTH of characters in cell B2, and then display the value of B2. This results in whatever number of zeroes being added to the beginning of the number in column B to create five digits.

The next column uses the concatenation operation discussed earlier to add the project code prefix to the five-digit number:

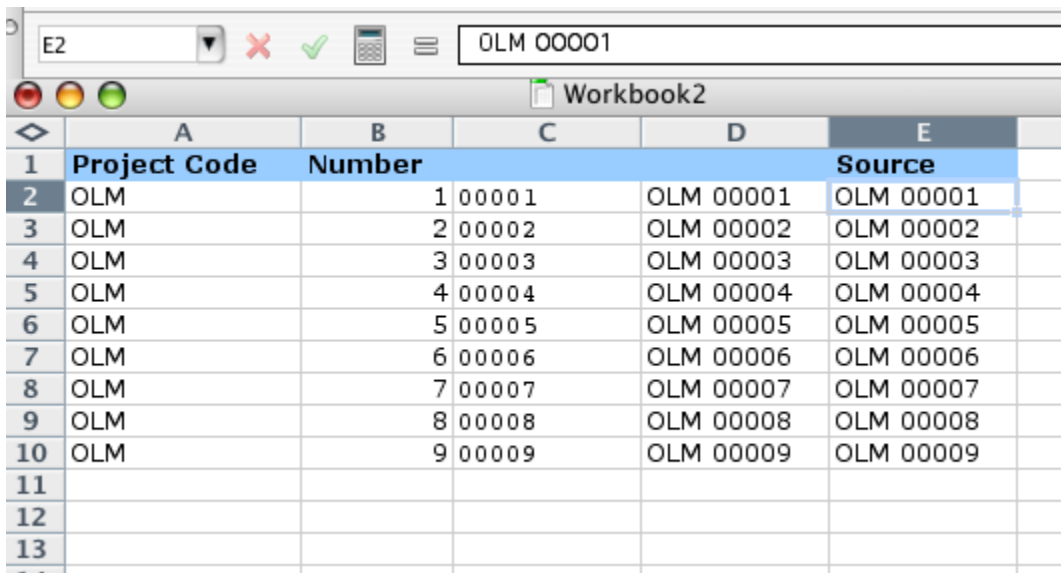


The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D
1	Project Code	Number		
2	OLM	1	00001	OLM 00001
3	OLM	2	00002	OLM 00002
4	OLM	3	00003	OLM 00003
5	OLM	4	00004	OLM 00004
6	OLM	5	00005	OLM 00005
7	OLM	6	00006	OLM 00006
8	OLM	7	00007	OLM 00007
9	OLM	8	00008	OLM 00008
10	OLM	9	00009	OLM 00009
11				
12				
13				
14				

In the formula bar, the code shows that the content of cell A2 is harvested, then a blank space is added, then the content of cell C2, resulting in the properly formatted source code.

The next step is to copy the data in column D and do the "paste special/values" operation in the next column to create a new "source" field that doesn't rely on the formula anymore:



The screenshot shows the same Excel spreadsheet as above, but with a new column E added. The formula bar for cell E2 shows the text "OLM 00001".

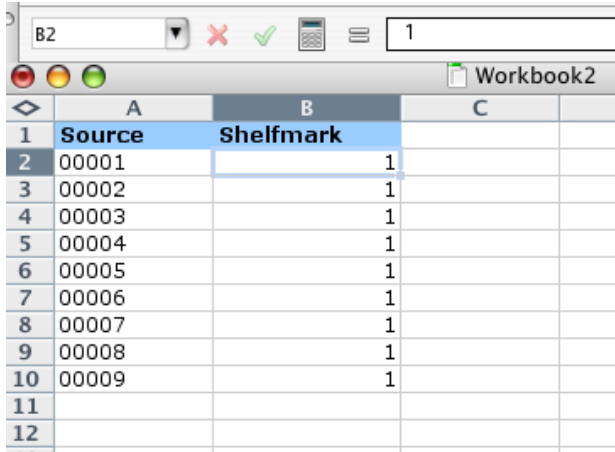
	A	B	C	D	E
1	Project Code	Number			Source
2	OLM	1	00001	OLM 00001	OLM 00001
3	OLM	2	00002	OLM 00002	OLM 00002
4	OLM	3	00003	OLM 00003	OLM 00003
5	OLM	4	00004	OLM 00004	OLM 00004
6	OLM	5	00005	OLM 00005	OLM 00005
7	OLM	6	00006	OLM 00006	OLM 00006
8	OLM	7	00007	OLM 00007	OLM 00007
9	OLM	8	00008	OLM 00008	OLM 00008
10	OLM	9	00009	OLM 00009	OLM 00009
11					
12					
13					
14					

Notice that the formula bar for cell E2 reads, "OLM 00001" instead of the formula code. Now the other columns can be deleted without changing this data and the user has the information formatted as desired.

Getting Rid of Leading Zeroes

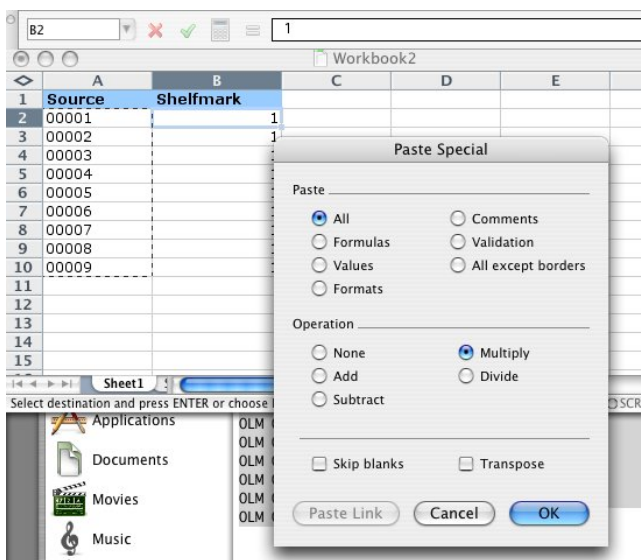
Sometimes a user ends up with some "harvested" numbers with padding zeroes that are rendered in Excel as text and need to be "broken down" back into regular numbers without any padding.

For example, a HMML source number might need to be reduced back to a regular number to serve as a "shelfmark" field entry. In this case, the first thing to do is a search and replace to get rid of the project code, leaving the five-digit number as such:

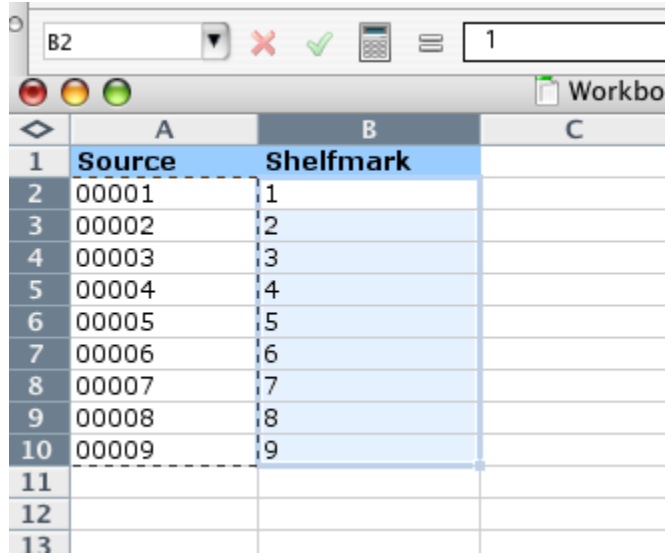


	A	B	C
1	Source	Shelfmark	
2	00001	1	
3	00002	1	
4	00003	1	
5	00004	1	
6	00005	1	
7	00006	1	
8	00007	1	
9	00008	1	
10	00009	1	
11			
12			

Notice the "shelfmark" column; this is created and populated with "1" in every field. Now the user takes the contents of column A, copies it and does another "paste special" operation into column B, only this time instead of using the "values" option, the user chooses to "multiply." Since the value of cells in column B is "1", the cells reformat to regular numbers based on the numeric value of column A (multiplying by 1 doesn't change the value).



The result looks like:



The image shows a screenshot of an Excel spreadsheet window. The window title is 'Workbo'. The active cell is B2, containing the value '1'. The spreadsheet has three columns: A, B, and C. Column A is labeled 'Source' and column B is labeled 'Shelfmark'. The data is as follows:

	A	B	C
1	Source	Shelfmark	
2	00001	1	
3	00002	2	
4	00003	3	
5	00004	4	
6	00005	5	
7	00006	6	
8	00007	7	
9	00008	8	
10	00009	9	
11			
12			
13			

Learning More

This short document scratches the surface of the Excel's formula and function capabilities. Users desiring to learn more can find many books on the subject as well as information in numerous internet forums.