

## **Automating Construction Scope Functions**

### **Executive Summary**

The company that this project work was done for requested to remain anonymous, and will be referred to as "Construction Co." throughout this document. Any information that could reveal the client has been removed, and any real data given has also been removed, leaving a blank template of what would generally include much more information. Construction Co. is a construction company that builds large-scale, specialty buildings around the world, and have a large presence in the United States and in several South American countries. In managing the creation of these specialty buildings, Construction Co. goes through the process of splitting the building project into several smaller projects. Using either their own resources or reaching out third party vendors to do the work, they manage the smaller projects to build high quality buildings, often used by tens of thousands of people each year.

The goal of this project for Construction Co. was to make the creation of project documentation more streamlined for their management. Construction Co. uses a very specific format for their project documentation and requires each of their project managers to adhere to this format. Prior to the creation of this project, each project manager created and formatted new project documentation by hand, or through creating copies of previous projects and re-formatting them for a new project. To help streamline the project documentation process, sub procedures were created to automate three main functions that are used when creating project documentation at Construction Co. These three main functions include adding a new "scope sheet" to a project, adding a new "bidder" to a scope sheet, and properly recording new scope sheets in the project summary sheet. The scope sheet is used when a piece of a construction project is being hired out to a third party. As bids for the scope are received from third party vendors, a new bidder is added to the scope sheet, along with their forecasted costs. Each new scope is added to a summary sheet and is placed according to whether it is a new scope, meaning that it is already approved for the project, or as an alternate scope, meaning that it is a possible change to the overall scope of the entire project being managed. With these sub procedures in place, the creation time of project documentation for managers at Construction Co. has been drastically reduced.

### **Implementation Documentation**

#### **General Information**

Construction Co. uses a very specific format for their project documentation, and per their request, this project focuses on automating three main functions that are frequently used by project managers when creating project documentation for construction projects. These three main functions are:

- Adding a new "scope sheet" to the project file at the desired location.
- Adding a new "bidder" to a given scope sheet, along with sum calculations.
- Adding the new scope information to a summary sheet in the correct "approved" or "alternate" location.

These three main functions are implemented in this project through three buttons added to the ribbon menu of the spreadsheet, and are located under a custom tab called “Spreadsheet Functions”. The three main functions requested by Construction Co. do not correspond directly to the buttons, however, because the function of adding a scope sheet and adding the information to the summary sheet are to be done at the same time. Instead, the three buttons are called Add Scope, Add Alternate, and Add Bidder as requested by Construction Co. Further detail on these buttons and the functionality behind them is described below.

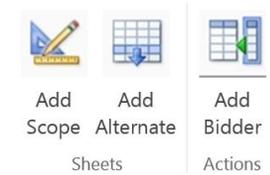


Figure 1 - Custom Ribbon Buttons

### Add Scope Button

The ‘add scope’ button is the first button located under the custom “Spreadsheet Functions” tab. The purpose of the ‘add scope’ button is to create a new scope sheet, formatted according to the specifications given by Construction Co., as well as formatting and adding the new scope sheet’s name and tab location to the summary sheet. According to Construction Co., the ‘add scope’ button is to be used when adding scope to the project that contributes to completing the previously approved project requirements. The scope information on the summary sheet is added to the upper section of the sheet, where scope that fits these requirements goes.

To accomplish this task, the code was split into three separate sub procedures. These sub procedures are called `newScopeSheet`, `addToSummary`, and `addNormalScope`. The first of the sub procedures called `newScopeSheet` works by creating a new sheet in the workbook that has been formatted according to the specifications of the scope sheet template provided by Construction Co. This sub procedure calls for three parameters when it is called, which are provided by a custom user form created to collect the proper information. The user form, called `newScopeForm`, asks for three things from the user: the name of the new scope sheet, whether it should be positioned before or after a sheet, and which sheet it should be placed before or after. After collecting these three pieces of information, they are passed into the `newScopeSheet` function.

The image shows a user form titled 'Add New Scope'. It has a blue header bar with a close button (X) on the right. The form contains the following elements: a text input field labeled 'Scope Name:'; two radio buttons labeled 'Before' and 'After', with 'After' selected; a dropdown menu labeled 'Choose Sheet:' with 'Alternate' selected; and two buttons at the bottom labeled 'Cancel' and 'Add'.

Figure 2 - User Form for Scope Sheet Creation

The `addToSummary` sub procedure takes two parameters which are the name of the scope and the tab number of the sheet that the scope is on. This sub procedure works by going to the pricing summary sheet in the project and finding the last entry in the top section of the summary sheet, and adding formatted cells with the scope name and tab number. This is achieved by simply looking for the line included in the pricing summary template that contains the subtotal information for the scopes in the top section and then inserting cells above and properly formatting them.

The final sub procedure used by the 'add scope' button is `addNormalScope`. This sub procedure is more of an organizational sub procedure, as it calls each of the other procedures in the proper order. It begins by calling the user form, which collects and passes the proper parameters to create a new scope sheet with `newScopeSheet`, and finally adds the new scope sheet information into the pricing summary by calling `addToSummary`. This sub procedure includes some checks so that the functions work properly, or end properly if the cancel button is pressed on the user form.

	A	B	C	D	E	F
1	<b>Example Project</b>					
2	Example Description			Example		
3	JOB # - 123456789					
4	DESCRIPTION					
5						
6						
7						
8						
9						
10						
11						
12	Base Bid Total >>>>>					
13						
14						
15						
16						
17						
18	Bid Total >>>>>					
19						
20						
21						
22						
23						

Figure 3 - Newly Added Scope Sheet

TAB	Scope/Contractor	Sub Quotes / Estimates	Approved Contract Value	GMP Budget	Taxable Material	Tax on Taxable Material at x.x0%
6	Example					
	Bid 1	4,500.00	4,500.00	x	x	x
	Bid 2	4,550.00	4,550.00	x	x	x
	Bid 3	4,375.00	4,375.00	x	x	x
	<b>Sub-Total Cost of Work</b>	4,375.00	#REF!	#REF!	#REF!	#REF!
	Demo		\$1,000.00	x	x	x
	Demo		\$2,000.00	x	x	x
	Demo		\$3,000.00	x	x	x
	<b>Subtotal</b>					
	Demo					
	Demo					
	<b>Subtotal</b>	4,375.00	\$0.00	#REF!		

Figure 4 - Example Scope Added to Pricing Summary

### Add Alternate Button

The 'add alternate' button is very similar to the 'add scope' button in functionality. It functions by again creating a new scope sheet, formatted according to the specifications given by Construction Co., however instead of adding the new scope information to the top section of the pricing summary sheet, it is added to the bottom section of the summary sheet. According to Construction Co., alternate scopes are added when the new scope is not defined by or does not contribute to the original project requirements. Therefore, the scope information for alternate scopes is added to the bottom section of the scope sheet. The bottom section of the pricing summary sheet is formatted differently than the top section, due to extra information being needed for each alternate scope.

The code to complete this functionality was again split into three sub procedures, including the `newScopeSheet` sub procedure that was described under the section above for the 'add scope' button. The other sub procedures involved are `addAlternate` and `addAlternateScope`. These two sub procedures function very similarly to the `addToSummary` and `addNormalScope` sub procedures described above. The real difference lies in the `addAlternate` sub procedure which adds scope information to the pricing sheet in a different way. Instead of simply adding the alternate scope formatting and scope information to the bottom the worksheet, the sub procedure searches to find whether or not it is the first alternate procedure, and if it is to react accordingly. If it is not, it will add itself at the bottom, below the others.

The `addAlternateScope` sub procedure is again, more of an organizational sub procedure. It works in the exact same way as `addNormalScope` as described above, except that it calls for an alternate scope instead of a normal scope. It calls the same user form to create a new scope sheet, and organizes the other procedures into the proper order. It also includes the same checks as described above to ensure that things function as they should.

7	Alternate				
	Alternate Bidder				
	<b>Sub-Total Cost of Work</b>				
	Add something extra				
	extra	1,000.00			
	extra	2,000.00			
	extra	3,000.00			
	<b>Subtotal</b>	6,000.00			

Figure 5 - New Alternate Addition to Pricing Summary with Mock Data

### Add Bidder Button

The 'add bidder' button is the third and final button of the custom buttons added to the ribbon. This button is meant to add a properly formatted bidder column, including the bidders name and sum functions to the active scope sheet. Each new bidder column is dynamically created to match the sizing of the columns already in place. The formulas are also input to change dynamically as the number of rows increases, meaning the user never has to worry about adding their own functions, nor do they need to worry about adding or removing rows.

Only one sub procedure was used to complete the functionality behind this button, which was called `addBidder`. This sub procedure includes all of the formatting and design for adding the new bidder to a scope sheet. It includes all of the checks to ensure that the bidder is added to the proper place on the sheet, is the same size as the other pieces of the table on the sheet, and also ensures that the sheet itself is a scope sheet before doing any actions. All functions requested for each bidder were included and are added upon running the sub procedure.

	A	B	C	D	E	F
1	<b>Example Project</b>					
2	Example Description			Example		
3	JOB # - 123456789					
4	<b>DESCRIPTION</b>	<b>BIDDER</b>	<b>BIDDER</b>	<b>BIDDER</b>		
5		Bid 1	Bid 2	Bid 3		
6	Plumbing					
7	Tubing	\$500.00	\$450.00	\$525.00		
8	Piping	\$1,200.00	\$1,400.00	\$1,100.00		
9	Other Materials	\$2,300.00	\$2,700.00	\$2,500.00		
10						
11						
12	<b>Base Bid Total &gt;&gt;&gt;&gt;&gt;&gt;</b>	\$4,000	\$4,550	\$4,125		
13						
14	Additional Labor	\$500.00	\$0.00	\$250.00		
15						
16						
17						
18	<b>Bid Total &gt;&gt;&gt;&gt;&gt;&gt;</b>	\$4,500	\$4,550	\$4,375		
19						
20						
21						
22						
23						

Figure 6 - Scope Sheet with Sample Bidders and Data

## Learning Concepts and Conceptual Difficulties

In doing this project, I learned that dynamic formatting is extremely difficult to do. When looking from the outside at the size of what can be done with the project, it may not seem like much, but it took many hours to work out everything according to the specifications that Construction Co. gave to me. One of the most difficult things at first was finding out how to place things dynamically without many reference points to be able to use on the page. After fiddling around, I started using the template formatting as my reference point, and was able to make everything happen in the way I wanted by using the color of a row as my guide, since many things start out blank. It was using the color of the row, along with a few other things, that I was able to create a way to dynamically place the correctly formatted sections for new bidders on the scope sheet, and for adding both regular and alternate scopes onto the pricing summary sheet. It was difficult, but fun to have to think in out of the box ways to get things done.

One of the other difficult things that I came across that took me a while to finish was inputting dynamic relative references to cells. When inputting the sum functions that should be added into each new bidder, the function needed to reference a range located above the cell so that it could sum the contents of the range. The problem was that new rows can be added to the scope sheet, which means absolute references were not the answer. The answer, instead, came in the R1C1 formula format. This allows you to reference the cells that you want, in reference to the location of the cell that you are inserting the formula into. Trying to use variables to create a range with this format got pretty confusing, and took a while to get correct.

```
baseBidCell.FormulaR1C1 = "=SUM(R[-" & (baseBidCell.Row - bidNameCell.Offset(1, 0).Row) & "]C:R[-1]C)"
```

*Figure 7 - Example Code Using R1C1 Format for Relative References*

Lastly, I learned a lot about how things need to all be put together in order to create a larger project. This final project is obviously much larger than any of the other projects that we have done for class, and because of that it needed to be organized in a much better way. There were several times when, after coding something, I knew there was a better way for it to be done. So refactoring the code and splitting things up into smaller sub procedures so that it was cleaner took up a good chunk of time as well. I was surprised at the amount of time and effort it took to create the functionality for this project.

## Assistance

I did not receive any assistance from others when completing this project. I communicated frequently with my contact at Construction Co., who gave me guidance in how they wanted things done, but no help was given in terms of how to code things. Everything that was coded was done by my own hands. I did search the internet several times for how to do things, like how to use the R1C1 formula formatting. The internet was a very valuable resource for me in completing this project.

## Conclusion

The goal of this project was to complete three main functions: 1) allowing new scope sheets to be added with proper formatting, 2) allowing new bidders to be added to scope sheets with dynamic formatting and correct calculations, and 3) allowing each new scope to be properly added to the correct place on the summary sheet with correct formatting. Though it was difficult and it seems like there isn't

way too much to show for the work done, the work was done well and successfully fulfills the requirements given by Construction Co.