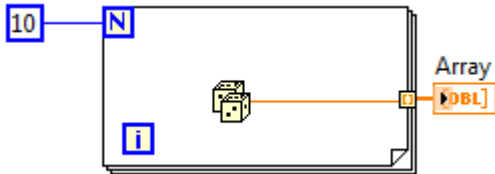


# Homework 5: SubVI's, Documentation and File I/O

---

## Part 1: Creating the VI

Create the following VI; try to do it without the instructions below.



*Check for understanding:* What does this VI do?

1. Open a Blank VI
2. Draw a **For Loop**.
3. Right click the *count (N) terminal* of the **For Loop** and choose create constant. Set the **For Loop** to execute 10 times.
4. Place a **Random Number (0-1)** primitive inside the while loop
5. Wire the output of the **Random Number (0-1)** primitive function to the right edge of the **For Loop**. The resulting tunnel that is created should default to be *auto-indexed*.
6. Right click the output of the *auto-indexed tunnel* and choose Create > Indicator.

## Part 2: Turning it into a Sub-VI



At this point you should now be able to run your program. You can expand the *Array Indicator* on the front panel by clicking and dragging the lower right corner. The program should output 10 random numbers between 0 and 1.

On the front panel in the top right corner is the connector pane, it looks like this in LabVIEW 2011:




The square on the right with the picture is the *Icon*, the square on the left with the grid marks is the *Connector*. In LabVIEW 2010 and earlier, you must right click the *Icon* and click Show Connector to view the *Connector Pane*.

## Editing the Icon

1. To edit the *Icon*, just double click it.
2. I always like to start out by double clicking the lasso selector tool  and then pressing the delete key. This gets rid of the existing picture.
3. Next, double click the box tool . This gives you a nice solid black border for your *Icon*.
4. You can then draw your own picture with the tools on the right, or go to the *Glyphs* tab for some premade ones. You also have the option of typing on the icon in the *Icon Text* tab. Create a picture so that others will have a clue as to what your program does.

## Editing the *Connector Pane*

Any control or indicator on the front panel can be wired as an input or output for your SubVI. Common coding practice is to wire inputs on the left and outputs on the right. We want to wire our *Array Indicator* as an output, here's how to do that.

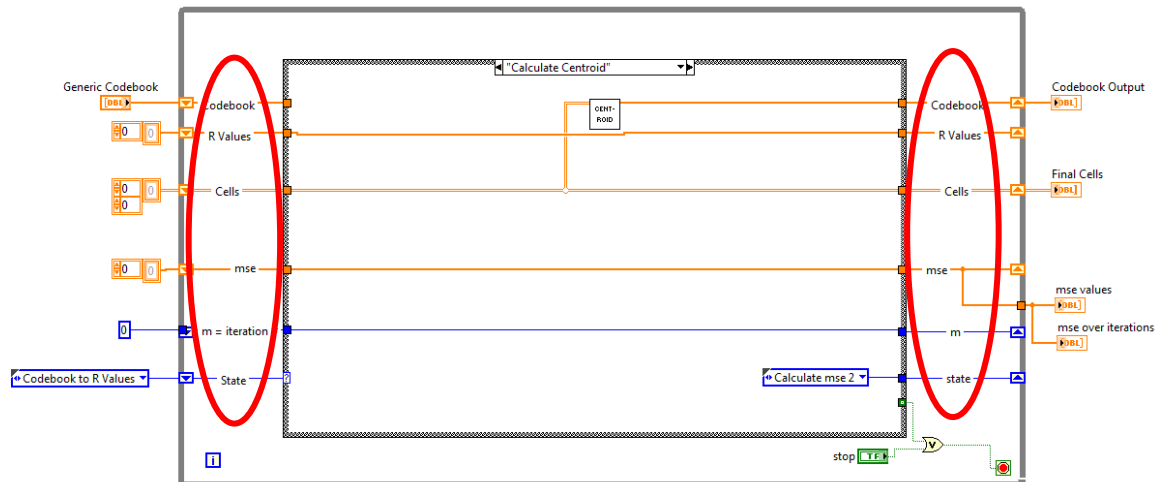
1. Mouse over the *Connector Pane* . You should see your cursor turn into the wiring tool
2. Any of the little grid squares on the *connector pane* can be used to create a terminal for your SubVI. Since this is an output, click on one of the boxes on the right side and then click the *Array Indicator*. You just linked those two things together.
3. Save the VI as "Random Array SubVI"

## Commenting your Code

There are many places you can document your code so that others can more easily figure out what your program does. Here are various ways to comment

1. **Double click anywhere in the gray area of the front panel, or the white area of the block diagram.** It will create a text box you can type anything you want in and then move around as you desire. Try this in the Random Array SubVI front panel and block diagram..
2. You can right click on Controls and Indicator and go to Description and Tip... . From here you can type what will show in the context help (Ctrl+H) by typing in the description. You can also edit what will show when you mouse-over the control or indicator by editing the tip. Try this with the Array indicator in your Random Array SubVI.
3. A new feature in LV 2011 is the ability to make Labels on wires. To do this, right click on a wire and go to Visible Items > Label. It might not be necessary to label every wire you create, but be mindful of confusing areas in your programs that might need this kind of documentation. Try this with the wire coming out of the random number generator in the for-loop of the Random Array SubVI.

A common place for these labels is on wires that are outside of the case structure but inside the while loop of a state machine structure VI like the image on the next page:



### Part 3: Using your SubVI

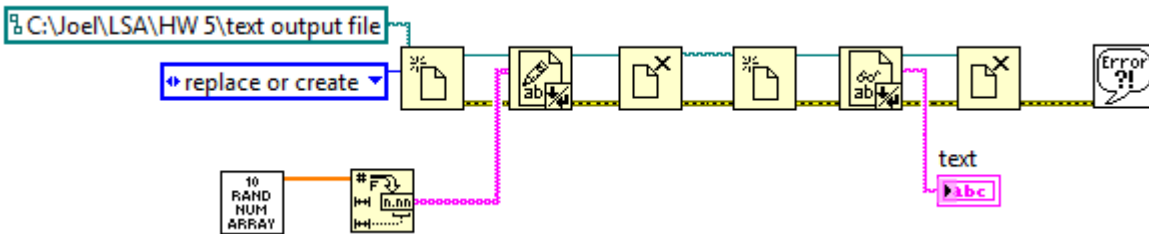
1. Open a new blank VI by going to File > New VI.
2. Click and drag the SubVI *Icon* you created earlier to the block diagram of the new VI. You can also choose the *Select a VI...* option from the functions palette, then navigate to your SubVI and place it.
3. Save the new VI as "Top Level"

Any VI can be made into a SubVI in this way.

NOTE: If you are working on a VI and it starts getting cluttered, think about how you could group sections of code into SubVI's to save space. If you do the same thing multiple times it would also be useful to create a SubVI of that section of code. To easily create a SubVI from code in an existing VI, you can select the section of code you want to turn into a SubVI and then go to Edit > Create SubVI.

## Part 4: File I/O

In your Top Level VI create the following:



1. Right click on the block diagram to bring up the functions palette. Go to String > String/Number Conversion and place a **Number to Fractional String** function.
2. Wire the output of your Random Array SubVI to the *number* input of the **Number to Fractional String** function.
3. Place an **Open/Create/Replace File** function; it is in the programming palette under File I/O.
4. Right click the *file path* input and click Create > Constant
5. Open a Windows Explorer window and navigate to where you want your output file saved. Copy that path and paste it in the file path box. At the end of that path type a “\” followed by what you want the name of the file to be. i.e: “\text output file”
6. Right click the *operation* input of the **Open/Create/Replace File** function and Create > Constant. Change value of the enum that appears to ‘replace or create’
7. Place a **Write to Text File** function and wire the *refnum out* of the **Open/Create/Replace File** function to the *file* input of the **Write to Text File** function.
8. Place two **Close File** functions, another **Open/Create/Replace File**, and a **Read from Text File** function. All of these functions are in the File I/O palette.
9. Place a **Simple Error Handler.vi** function from the Dialog & User Interface palette in the programming palette.
10. On the text output of the **Read From Text File** function, right click and create an indicator
11. Wire up the VI as shown above, don’t forget the error clusters!
12. Run the VI
13. Check the folder where you saved the output file. This file can be opened in Notepad, Word and Excel easily. An easy way to do this is to change the file extension in the *file path* box to .txt, .doc or .xls. i.e **C:\Joe\LSA\HW 5\text file output.xls** You can also double click a file with no extension and windows should ask you what program you wish to open it in.
14. On the front panel of your Top Level VI, click and drag the corner of the text indicator so you can see the entire output. Verify the output file and the front panel have the same 10 random numbers. You have just successfully written to a file, and then read back from it in the same VI!
15. Comment your code so that others can easily see what your program is doing.