How to Use the MathScript Node Tutorial

Functions -> Programming -> Structures [icon] -> MathScript Node [icon]

Start by opening up the Functions palette and expanding the Programming palette. Under Programming select the Structures palette [icon] where you will find the MathScript Node [icon] structure.

When you select the MathScript Node structure on the palette, you will notice that your cursor will change to a cursor with an "fx" symbol.

To create the structure on the block diagram, click and drag an appropriate sized box for your MathScript Node (note you will be doing some coding here so it's best to make it large enough to edit code). You can resize the structure later by selecting the structure and dragging the handles.

Now that we have created a MathScript structure we will explain what a MathScript structure actually does. You will notice that in the MathScript structure there is an editor where you will put your code. In this editor region, you will write your code telling the MathScript Node what to do with the inputs and what to pass to the output.

To create inputs and outputs, first select the MathScript Node that you want to add an input or output to and right click on the structure border. In the menu that appears, select the "Add Input" or "Add Output" option depending on what you wish to do (Figure 1) and how many you need.
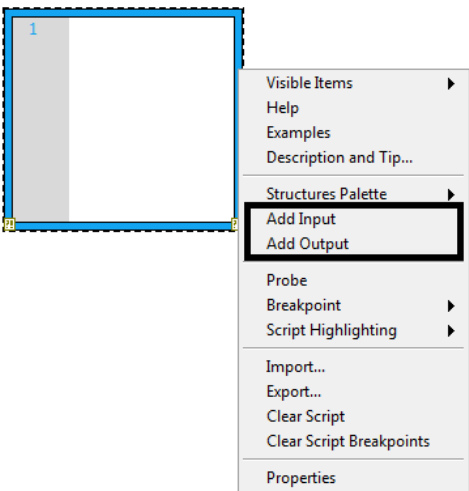

Figure 1

When you create an input or an output, a box will appear on the border of the MathScript Node structure and will be highlighted for you to edit. What you put into the box matters as we will see later since the names associated with each input and output will correspond to the variable names

that will used in the MathScript editor (you can change the names later by double clicking the input or output box).
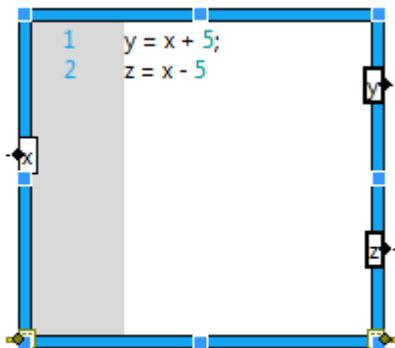

Figure 2

Now that we have created inputs and outputs, and labeled each one appropriately we can now insert some code into the MathScript box such as the example in Figure 2 (Note that you can add inputs and outputs anytime you want while you're coding also). The syntax convention in the MathScript Node structure is similar to .m files used in Matlab. The advantage of using the MathScript Node is that it allows you to interface your block diagram with actual code or script.

In the MathScript Node editor, you can write the necessary instructions to compute the output values from the input values. Notice that you must use the names of the input values and output values in your code in order for your code to perform the appropriate tasks.
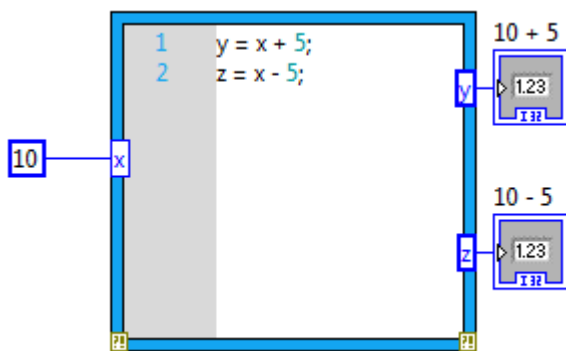

Figure 3

Finally, you can wire the input and output values to indicators, constants or controls depending on what your block diagram will do as shown in Figure 3. Realize that the MathScript Node structure is simply a tool that allows you to build a custom function that performs the tasks specified in the code component of the structure. Thus it acts as an interface between the text representation of a function and the block diagram representation of the function.

It is important to notice that the different dynamics in the execution of the MathScript Node as compared with the data flow directed programming. The MathScript Node executes linearly with one thread within the MathScript Node's traditional program model as opposed to the multiple Parallel Threads that execute concurrently in the data flow model. Therefore, when we use a

MathScript Node, it essentially a programmable function that behaves like any other built in function in the block diagram.

Tutorial Notes

SECOND SET OF REVISIONS COMPLETED 12:54 PM 8-16-10

Explanation of data flow directed programming and traditional programming model may be extraneous. Not really necessary for this tutorial but could simply link it to an article highlighting how data flow directed programming differs from what students are used to. Would end up in the introductory section. Probably shouldn't be too long or else people might get bored.

Article on data flow programming will be part of the Lab 4 set

Unable to capture "fx" cursor icon with print screen