How to Use MathScript from the Command Line Tutorial

Tools -> MathScript Window

To use MathScript with the command line, first we need to open up the MathScript Window which can be found under Tools as illustrated in Figure 1.

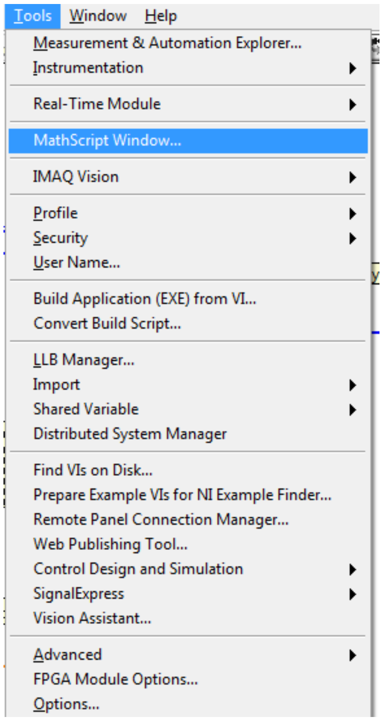
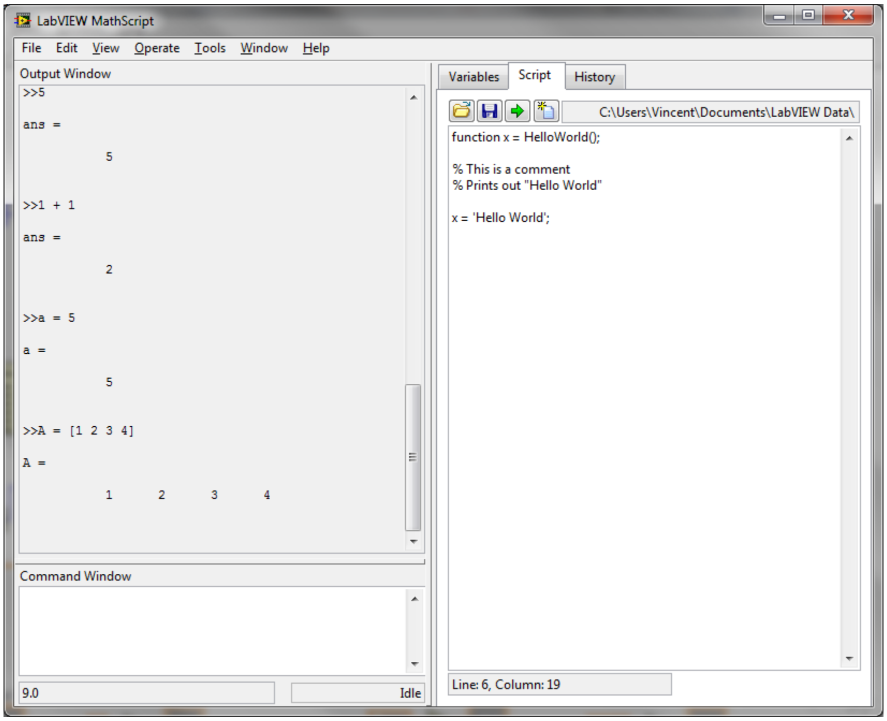


Figure 1

A new window should appear like the one in Figure 2. You will notice that the left half is partitioned into a command prompt on the bottom and an output display on the top. The command prompt and output display are simplified versions of the MatLab interface but otherwise behave the same way the MatLab command prompt works.

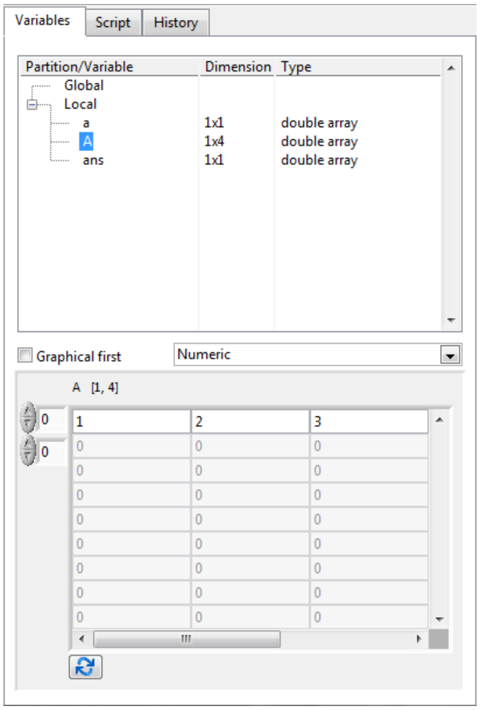
On the right half of the MathScript Window, you will find a Variables Tab, Script Tab, and a History Tab.



```
>>5

ans =

         5


>>1 + 1

ans =

         2


>>a = 5

a =

         5


>>A = [1 2 3 4]

A =

         1      2      3      4
```

```
function x = HelloWorld();

% This is a comment
% Prints out "Hello World"

x = 'Hello World';
```

Figure 2



Partition/Variable	Dimension	Type
Global		
Local		
a	1x1	double array
A	1x4	double array
ans	1x1	double array

□ Graphical first   Numeric

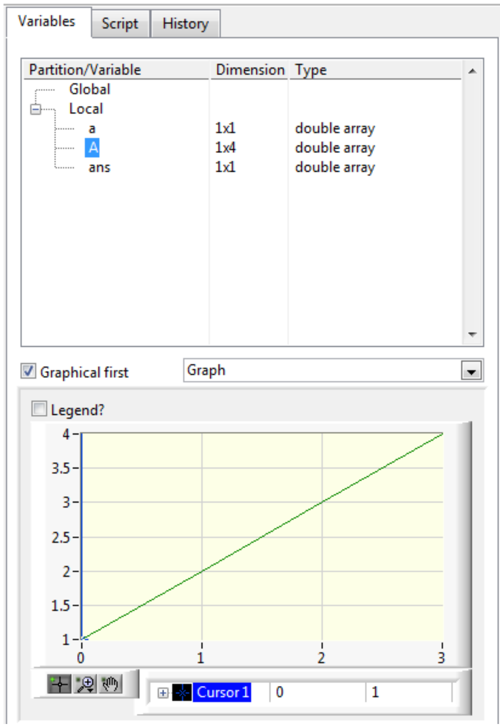
A [1, 4]

1	2	3
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

The Variables Tab will show you a list of each variable and what data type each variable is as shown in Figure 3. You will notice that when you select a variable that is in the list, the value or structure will be displayed in the section underneath. In Figure 3, a 1 x 4 array structure is selected so the display underneath shows a window with the values in the array at their corresponding positions.

In addition, you can also display graphs of the values in the variables. To display the graph for an array of values, click on the variable and check the "Graphical first" box as shown in Figure 4.

Figure 3   Figure 4



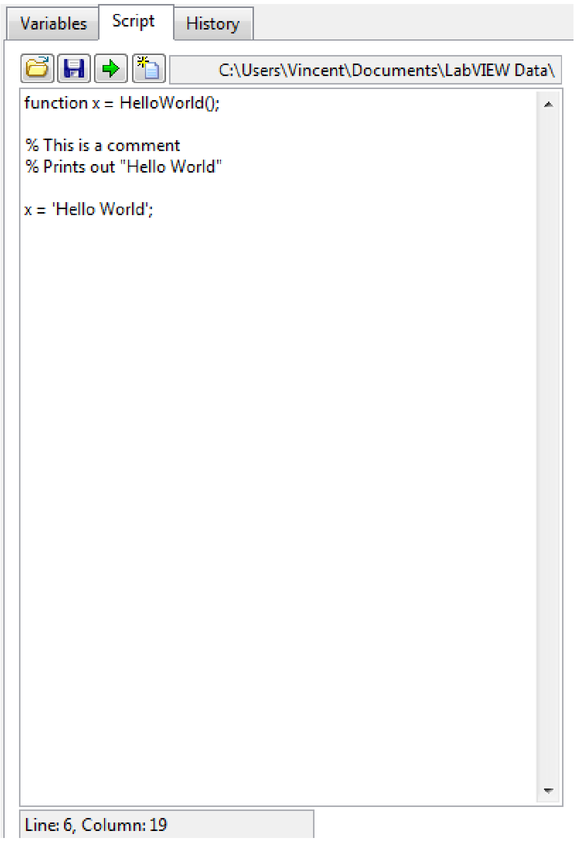
Partition/Variable	Dimension	Type
Global		
Local		
a	1x1	double array
A	1x4	double array
ans	1x1	double array

☑ Graphical first   Graph

□ Legend?

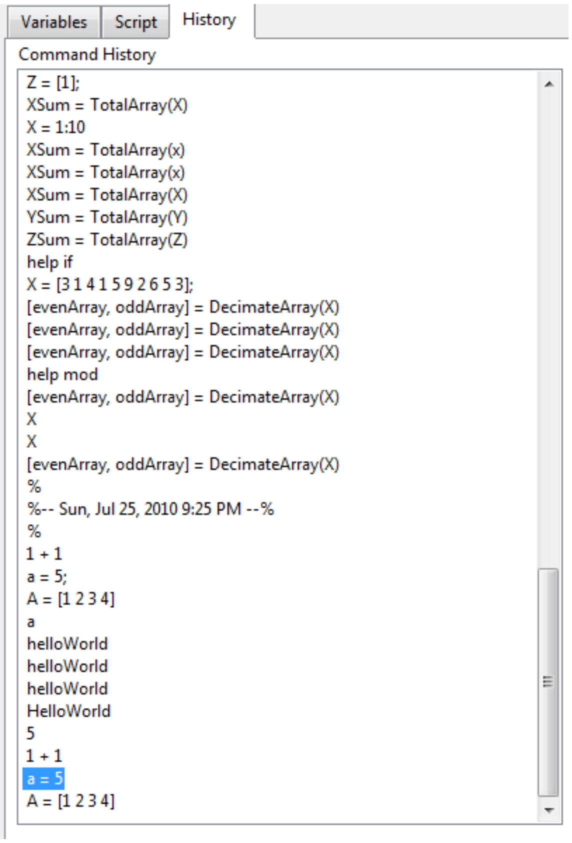
You will also notice that there is a Script Tab that contains a MathScript editor. This editor is intended to be used as an interface for you to write MathScript. You will notice in Figure 5 that you can load, save, run, and clear your script editor by selecting the appropriate button on the toolbar. You will notice that the save file format is a .m file and in order for LabView to detect your custom .m file it must be saved under LabVIEW Data or else when running your script in the command prompt, the command prompt will throw an error because your script was not loaded from where the library files are loaded.

Finally, the last tab you will notice is the History Tab where you will find a log of the entire history of commands that were entered into the command prompt as shown in Figure 6.



```
function x = HelloWorld();

% This is a comment
% Prints out "Hello World"

x = 'Hello World';
```

Line: 6, Column: 19



```
Z = [1];
XSum = TotalArray(X)
X = 1:10
XSum = TotalArray(x)
XSum = TotalArray(x)
XSum = TotalArray(X)
YSum = TotalArray(Y)
ZSum = TotalArray(Z)
help if
X = [3 1 4 1 5 9 2 6 5 3];
[evenArray, oddArray] = DecimateArray(X)
[evenArray, oddArray] = DecimateArray(X)
[evenArray, oddArray] = DecimateArray(X)
help mod
[evenArray, oddArray] = DecimateArray(X)
X
X
[evenArray, oddArray] = DecimateArray(X)
%
%-- Sun, Jul 25, 2010 9:25 PM --%
%
1 + 1
a = 5;
A = [1 2 3 4]
a
helloWorld
helloWorld
helloWorld
HelloWorld
5
1 + 1
a = 5
A = [1 2 3 4]
```

Figure 5

Figure 6

If you ever get stuck or need to find out how a certain built in function works in the command prompt, simply enter **help** followed by the function name and a help window will appear.