

[7.25] Use scripts to test program and functions

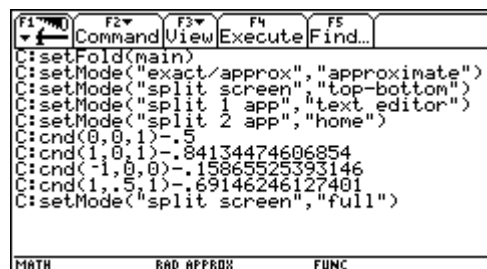
Scripts are text files that include commands that you can execute from the text editor. See the *T189/92+ Guidebook*, page 328 for details. Also see page 94 for instructions to save the home screen as a text file, to be used as a script.

This can be very useful as you are testing and debugging functions with different arguments. You can enter each function call as a command in the script file. This also makes 'regression testing' easier. Regression testing is a software engineering term that refers to verifying that a function still works properly after changes have been made. Suppose that you have written a function and tested it, and you think that it is correct. However, after using it for a while, you find that it does not work for some arguments. If you have saved the test calls as a script file, you can easily verify that your corrections do not affect the parts of the program that *did* work, before.

You can also use commands in the script file to change the modes (Auto, Exact, Radians, Complex, etc.) and change folders, if necessary. Any command that can be executed from the command line can be included in the script file.

Using script files for testing works best if the Split screen display mode is used. This way, you can see both the script file commands and the results.

As an example, suppose I have written a function to find the cumulative normal distribution called *cnd()* in my *main* folder. This function takes three arguments and returns a single numeric result. I want to test it with a variety of input argument combinations, and compare the results to those I have found from another source. I use the text editor ([APPS] [8]) to create a new text file with the name *cndtest*. The text file looks like this:



```
C:\setFold(main)
C:setMode("exact/approx","approximate")
C:setMode("split screen","top-bottom")
C:setMode("split 1 app","text editor")
C:setMode("split 2 app","home")
C:cnd(0,0,1)-.5
C:cnd(1,0,1)-.84134474606854
C:cnd(-1,0,0)-.15865525393146
C:cnd(1,.5,1)-.69146246127401
C:setMode("split screen","full")
```

MATH RAD APPROX FUNC

While entering the command lines, use [F2] [1] to make each line a command line. The first line sets the folder to Main. The second line sets the mode to Approximate, which is how I want to test *cnd()*. The next three lines split the screen horizontally, and put the text editor in the top window, and the home screen in the bottom window. This is just personal preference. The next four lines execute my test calls. Note that I subtract the actual desired answer from the *cnd()* result, so I can quickly see if it is working: all the results should be near zero. The last line sets the split screen mode back to Full.

Whenever I want to use this test script, I use [APPS] [8] [2] to open the *cndtest* text file, then [F4] (Execute) to execute each line.

It is not necessary to enter each command line in the text file. Instead, you can use the instructions on page 94 of the *89/92+ Guidebook* to copy most of the commands from the home screen into a text file. However, this won't necessarily work with the split screen commands, as the *setMode()* functions to set the Split 1 application won't work, of course, if you haven't yet created the *cndtest* text file.

Of course, you can automate this testing in another way: by creating a program that does the same operations as the text file.