

## Writing Executable Script Files to Automate Tasks on the UNIX<sup>1</sup>

It is possible to create a file to execute a series of commands on UNIX-based machines. To automate chores, you can create ASCII files that contain a series of commands, and then make that file executable. These files are called “script” files. These files do not require any specific file extension. When choosing a name for your script file, make sure that it does not conflict with an already existing command name, program, or alias on your UNIX.

Use caution when creating new script files. It is wise to avoid including potentially dangerous commands such as `cp`, `mv`, or `rm` in them. Also, please note that any aliases that are in your `.cshrc` file cannot be executed from a script file. For example, you could not use the command “`dir`”, but you could use the command “`ls -l`”.

A script file can be placed in the directory that it will be used in, or it can be placed in any directory that is in your “path”. (The path structure is set in your `.cshrc` file, which sets the environment for your UNIX session.) If it is in your “path” rather than in the directory you will run it from, then before it will run, you must either logout and login again, or type “`source ~/.cshrc`”.

Following is an example of using script called “**scrptest**” that has a series of commands in it. You can create the file with any text editor on the UNIX.

**Step 1:** Create a file called “**scrptest**” using any text editor on the UNIX.

```
#!/bin/csh
cd /var/tmp
echo Hi. The computer can be your friend. I am here to serve you!
```

**Step 2:** Make the file executable with the command: `chmod u+x scrptest`

**Step 3:** If Change to the directory where the file is stored, and then execute the program. The “&” allows you to do other work, or logoff, while the commands in “**scrptest**” continue.

```
scrptest &
```

Of course, it is unlikely that you would need a script file for the commands listed in this sample. But, there are times when script files can be very useful. For example, you may have a series of SAS jobs that take a long time to run, and they must be run in certain order. You can create a script file to run the all (sequentially if “&” does not follow the commands in the script file), and then log off. You might create a script file called “**runsfiles**”.

```
#!/bin/csh
# runsfiles. (Lines that begin with # are comments) December 31, 1999
# next line changes to directory with sas programs.
cd /full/path
sas file1
sas file2
sas file3
```

Make the program executable by typing “`chmod u+x runsfiles`”. When you are ready to run it, be sure you are in the directory where the file “**runsfiles**” is (or that it is in your path and you have typed “`source ~/.cshrc`”, and type:

```
runsfiles &
```

You may logoff, or do other work. UNIX and SAS will do your work for you.

---

<sup>1</sup>Prepared by Patty Glynn and Ruby Wang, January 24, 2000, Center for Social and Demographic Analysis..