

Automating Things with Batch Files They Work on Today's Computers Too!

If you're familiar with MS-DOS at all, you'll recall that it's a command-driven operating system that performs functions issued at the C:> prompt. The only way to get an MS-DOS computer to do something was to type a command at this prompt and if you can imagine, it was a rather cumbersome way to use a computer.

As an example, to load up Microsoft's simple editing program, you had to type the name of the drive that the program was on, the directory that the program was in, and then the name of the program. So if Microsoft Edit was in a directory or folder named "Process," you could start the program by typing, "C:>process\edit.com" Then, and only then would the program load up for use.

This is a small command, but just imagine if you had a program that was deeply nested within a series of folder. You could end up typing a command as wide as your computer screen or worse, long enough that the entire command would have to wrap onto the next line! Now imagine having to type these long commands every time that you wanted to start a program. Yikes!

That's one of the reasons why batch files became so popular. Batch files are small text-based documents that contain a bunch of these commands on their own lines. When executed, they would process each command without the user having to type each and every one of them.

When Windows was developed, the need for typing commands was essentially eradicated thanks to the introduction of the point-and-click (mouse) interface. But this didn't stop the batch file fever that started under MS-DOS - and in some small circles, batch files are still as popular as they were in the beginning.

Even though you may use Windows XP or Vista, batch files can save you tons of time by automatically starting multiple programs and performing different tasks at the single click of a button. They don't require any extensive programming background and they don't need to be encrypted with some weird, expensive compiler. Batch files are plain text files, and you can build one for your own personal use with Windows' Notepad.

You could make a batch file that loads up your favorite websites at once for example, or you could make a batch file that fills your desktop with the most important applications for the day. To do so only requires a little knowledge about the locations of these applications.

Let's say that every day we need to load up the Yahoo web browser, Microsoft

Word, and then the calculator that comes with Windows. Instead of doing this by hand, we could write a batch file to do it for us.

First, we'd load up Notepad and type in the following:

```
START "http://www.yahoo.com"  
START "c:/program files/microsoft office/office/winword.exe"  
START "c:/windows/calc.exe"
```

We would then save this data into a file named, "mytasks.bat" onto the Desktop for easy access. Each time we double-clicked on this file, the Yahoo website would load up, Microsoft Word would start, and the simple calculator would pop up.

Since we want these programs to load every day, we could create a shortcut to this file and then place the shortcut inside our computer's Start Up folder. That way, these three programs would load every time we turn on the computer. If you wanted these programs to start minimized, you could type the following into a batch file instead:

```
START http://www.yahoo.com /m  
START "c:/program files/microsoft office/office/winword.exe" /m  
START "c:/windows/calc.exe" /m
```

This will run all three programs as before, however the "/m" parameter will minimize them so that they don't clutter up the desktop.

Other people have found much more creative and effective ways to use batch files, but the important thing is that you know they're a resource you can use to save a few seconds or minutes in performing important tasks. We've come a long way from MS-DOS, but it's still a valuable source of automation that anyone can use with no programming knowledge at all.

PPPPP

Word count 681