

Forum:[Hack Forums](#)

Thread: [Setlocal & Delayed Expansion](#)

White Light - July 4, 2012, 12:42 pm

Before you start reading:

>>> [Here](#) <<< is the download link for the files you need. Please extract them to their own sub folder, and it is recommended that you don't run them until asked in the tutorial. Feel free to use any code in the archive for your personal use, but if you release it you must give me credit.

Now, I know that some of you guys have seen me post some fairly complex code. One of the more common commands that I use when I'm doing something like this is

Code setlocal enabledelayedexpansionNow, I know some of you may be thinking "White Fang, dafuq you doin? What da hell does dis shit mean?" Well, to answer that question, it means I have a fair bit of extra power in my scripts.

What does it mean?

The problem with explaining this command is that there's two parts to it that are ENTIRELY different.

Setlocal - This is the part of the command that's completely different from what the rest of the command does. Basically, it isolates that instance of CMD.exe from any others. Under normal circumstances, nested (one inside the other) processes of cmd.exe will affect each others' variables. Please take this moment to run "Number 1.bat".

Your output should look like this:

Code Orig: 1

New: 2

As you can see, the second batch file changed the value of var from 1 to 2. This can be handy for something like a plugin batch file which returns values to the parent batch file.

To see how SETLOCAL effects the interactions of the 2 batch files, please run "Number 2.bat".

The output will look like this:

Code Orig: 1

New: 1

This is because all the changes to variables within the second script are LOCKED there, and won't effect anything else.

EnableDelayedExpansion - This is what really gives your scripts the power they really need. This allows you to use the "Almighty ! Variables".

"But what does it really mean?"

Variables by themselves don't magically turn into what they are, they are "expanded" into what they need to be. When the interpreter reads the code, it expands variables once per line.

For an example of how the interpreter works, run "Number 3.bat".

Logically, %var% should expand to "goodbye world", but the interpreter has already expanded %var% to "hello world".

Just like setlocal by itself modifies the properties of the interpreter, EnableDelayedExpansion also modifies it. Like its name suggests, the expansion of variables is delayed until the interpreter REACHES the variable.

"Number 4.bat" has nearly the same code as "Number 3.bat", except delayed expansion has been enabled, and the second %var% is now !var!. Now, the result follows the logical thought process, and !var! becomes "goodbye world".

"But you almost never write code like that!"

And neither should you, it's not good coding practice. It just increases the visual complexity of the code and makes it harder to debug. That being said though, when a FOR loop is used, all the commands in the brackets are treated as one line.

Running "Number 5.bat" displays similar counter-logical results. It should display the name of all the files in the folder, but instead it displays "hello world". It's the same conundrum as before, the interpreter treats it as one line, so %var% is expanded only once.

A simple fix is found in "Number 6.bat". All that is needed is to enable delayed expansion, and activate it for !var!.

Well, that's it. Happy coding.

AMDjack - July 4, 2012, 1:11 pm

Nice tutorial, to be honest I never understood what it really did. Cheers

jessman420 - July 5, 2012, 6:24 am

Great tutorial I am sure this has helped many users in this section understand.

?CARUs L?VE?™ - January 11, 2014, 10:58 pm

Yet another HQ thread with only one page? More people need to understand this