ERRORLEVEL is not %ERRORLEVEL%

devblogs.microsoft.com/oldnewthing/20080926-00

September 26, 2008

Raymond Chen

The command interpreter `cmd.exe` has a concept known as the *error level*, which is the exit code of the program most recently run. You can test the error level with the `IF ERRORLEVEL` command:

```
IF ERRORLEVEL 1 ECHO error level is 1 or more
```

<sidebar>
The `IF ERRORLEVEL n` test succeeds if the error level is n *or more*. This was presumably because there were programs that expressed different degrees of failure with higher and higher exit codes. For example, the `diff` program has three exit codes: 0 means the files are the same; 1 means the files are different; 2 means that something terrible happened. There are also programs that use an exit code of zero to mean success and anything else to mean failure.
</sidebar>

In addition to this internal state, you can, if you wish, create an environment variable with the name `ERRORLEVEL`, in the same way that you can create an environment variable called `FRED`. But, as with `FRED`, that variable won't have any effect on the error level.

```
rem this next command sets the error level to zero
CMD /C EXIT 0
set ERRORLEVEL=1
if ERRORLEVEL 1 echo Does this print?
```

The message is not printed because the `ERRORLEVEL` environment variable has no effect on the error level. It's just a variable whose name happens to coincide with a command processor concept.

```
set BANKBALANCE=$1,000,000.00
```

"Hey, when I tried to withdraw the money, I got an insufficient funds error. What am I doing wrong?"

Now, it does happen to be the case that if command extensions are enabled and you say `%ERRORLEVEL%`, then the command processor first looks for an environment variable called `ERRORLEVEL`, and *if it can't find one*, then it replaces `%ERRORLEVEL%` with the current value of the internal error level value. It's a fallback step, in the same way that your neighbor is a fallback delivery location if you aren't home. If you file a change-of-address form for yourself, that doesn't affect packages sent to your neighbor.

The same behavior can be seen with `%CD%`: If you did not explicitly set an environment variable called `CD`, then `%CD%` expands to the command processor's current directory. But you can't change directories by saying `set CD=C:\Windows`.

I can think of a few reasons why this feature may have been added.

- So you can include the error level in a log file:
  `ECHO error level is %ERRORLEVEL%>logfile`
- So you can perform other types of tests against the error level, for example, to perform an equality test:
  `IF %ERRORLEVEL% EQU 1 echo Different!`

But I'm digressing. My point for today is that the error level is not the same as the `ERRORLEVEL` environment variable.

Raymond Chen

Follow