

Christian Specht (/)

← Excluding integration tests in NUnit's GUI, part 2 - Specifying a fixture (/2013/04/16/excluding-integration-tests-in-nunit-s-gui-part-2-specifying-a-fixture/) | Tinkerforge Weather Station, part 1 - Intro and construction → (/2013/06/17/tinkerforge-weather-station-part-1-intro-and-construction/)

Batch File Cheat Sheet (/2013/05/12/batch-file-cheat-sheet/)

May 12 2013, 23:45:00 | tags: command-line (/tags/#command-line)

Recently, I had some tasks at work that required me to write a lot of batch files.
Here are some things about batch files that I either learned or re-discovered in the process:

Don't use blanks when assigning values to variables

This works:

```
set foo=bar
```

This **does not work** (no error message - `%foo%` is just empty afterwards):

```
set foo = bar
```

This one took me some time to figure out. I'm used to Visual Studio and the Access VBA editor, which both automatically insert these blanks into my code if I don't do it myself. So when I wrote batch files with a simple text editor, I inserted these blanks too, without thinking.

Execute .exe and save the output to a variable

This line saves the output of `foo.exe` to a variable named `%bar%`:

```
for /f "tokens=*" %i in ('path\to\foo.exe') do set bar=%i
```

(Credits for this go to user "lesmana" at Stack Overflow (<http://stackoverflow.com/a/4695680/6884>))

Disclaimer: I don't try to understand it, I just copy/paste it every time I need it.

(The whole usage of the `for` command looks equally complicated - to see it in all its beauty, type `for /?` and read the help.)

Get the path and file name of the currently running batch file

There are other commands to get more file information, but to me the most important ones are these:

- `%~dp0` outputs the path of the current batch file

- `%~nx0` outputs the name of the current batch file

To see it in action, put this into a batch file:

```
echo %~dp0
echo %~nx0
```

When the batch file is `c:\test.bat`, it will output this:

```
C:\
test.bat
```

The number zero at the end of `%~dp0` and `%~nx0` refers to `%0`, which is the currently executing batch file.

Likewise, if you pass a file as parameter `%1`, you can get the name of *that* file by using `%~nx1` instead of `%~nx0`.

`%~dp0` and `%~nx0` are actually combined commands (*you can also use `%~n0` to get just the file name without extension, and `%~x0` to get just the extension*)

To see the complete list of commands, open the command line and type in `call /?`.

Temporarily add a folder to the `%PATH%` variable

When you need to refer files from a certain folder in a batch file several times and you don't want to copy/paste the complete path every time, you can just add the folder to the `%PATH%` variable.

When you don't want to (*or aren't allowed to*) do this permanently on the machine, you can set the `%PATH%` variable in the batch file, which lasts only for the execution time of the batch file.

The following line **appends** `c:\foo` to the existing content of the `%PATH%` variable:

```
path=%path%;c:\foo
```

Afterwards, the `%PATH%` variable contains everything it contained before **plus** `c:\foo`, which is probably what you want.

A second (*but inferior*) way would be:

```
path=c:\foo
```

This is probably **not** what you want, because it **replaces the whole content of the `%PATH%` variable** by `c:\foo` !

In other words, `%PATH%` will **only** contain `c:\foo` afterwards.

Beware: you won't notice the difference as long as you only need files from `c:\foo`.

But as soon as you need some other command-line tool like `git` or `hg` (*or some Windows helper like `regsvr32.exe`*), your batch file won't find them anymore because you just deleted their locations from the `%PATH%`.

Variables are global in nested batch file calls

Take a look at these two batch files:

batch1.bat:

```
set var=foo
call batch2.bat
echo %var%
```

batch2.bat:

```
set var=bar
```

The first batch file calls the second one, so variables with the same name are shared between the two.

This means that the last line in the first batch file (`echo %var%`) will output `bar` and not `foo` !

← Excluding integration tests in NUnit's GUI, part 2 - Specifying a fixture (/2013/04/16/excluding-integration-tests-in-nunit-s-gui-part-2-specifying-a-fixture/) | Tinkerforge Weather Station, part 1 - Intro and construction → (/2013/06/17/tinkerforge-weather-station-part-1-intro-and-construction/)

ABOUT ME



CV (<https://stackoverflow.com/cv/christianspecht>)
Contact (/imprint/)

Developer for about 16 years, living in Kerpen, Germany (near Cologne). Working as an in-house developer (*C#/T-SQL/VBA*) and DBA at weinor (<http://www.weinor.de>), a mid-sized manufacturing company in Cologne.

PROJECTS

Bitbucket Backup (/bitbucket-backup/)

MissileSharp (/missilesharp/)

Recordset.Net (/recordset-net/)

RoboShell Backup (/roboshell-backup/)


SCM Backup (<https://scm-backup.org/>)

SimpleLeague (/simpleleague/)

VBA Helpers (/vba-helpers/)

LINKS



 **Christian Specht**
30,328
● 13 ● 106 ● 164

(<https://stackoverflow.com/users/6884/>)

Open Hub

Christian Specht
1426 commits



(<https://www.openhub.net/accounts/131837?ref=Detailed>)



(<https://github.com/christianspecht>)



(<https://bitbucket.org/christianspecht>)

RECENT POSTS

[Creating a map with markers for printing...with bigger fonts! \(/2019/08/29/creating-a-map-with-markers-for-printing-with-bigger-fonts/\)](#)

[Goodbye Bitbucket Backup, hello SCM Backup \(/2018/08/01/goodbye-bitbucket-backup-hello-scm-backup/\)](#)

[VeraCrypt integration for RoboShell Backup \(/2018/01/07/veracrypt-integration-for-roboshell-backup/\)](#)

[Fixing Resource.Designer.cs generation in .NET Core \(/2017/11/29/fixing-resourcedesigner-cs-generation-in-net-core/\)](#)

[Running a combined Jekyll/PHP site on XAMPP \(/2017/10/04/running-a-combined-jekyll-php-site-on-xampp/\)](#)

ARCHIVES

[2019 \(1\) \(/archive/#2019\)](#)

[2018 \(2\) \(/archive/#2018\)](#)

[2017 \(4\) \(/archive/#2017\)](#)

[2016 \(5\) \(/archive/#2016\)](#)

[2015 \(3\) \(/archive/#2015\)](#)

[2014 \(8\) \(/archive/#2014\)](#)

[2013 \(11\) \(/archive/#2013\)](#)

[2012 \(5\) \(/archive/#2012\)](#)

TAGS

[.net \(7\) \(/tags/#.net\)](#)

[authentication \(1\) \(/tags/#authentication\)](#)

[backup \(8\) \(/tags/#backup\)](#)

[bitbucket-backup \(3\) \(/tags/#bitbucket-backup\)](#)

[blogofile \(3\) \(/tags/#blogofile\)](#)

[command-line \(5\) \(/tags/#command-line\)](#)

[hardware \(2\) \(/tags/#hardware\)](#)

[jekyll \(12\) \(/tags/#jekyll\)](#)

[maps \(2\) \(/tags/#maps\)](#)

[markdown \(4\) \(/tags/#markdown\)](#)

[ms-office \(3\) \(/tags/#ms-office\)](#)

[msbuild \(1\) \(/tags/#msbuild\)](#)

[nuget \(1\) \(/tags/#nuget\)](#)

[nunit \(3\) \(/tags/#nunit\)](#)

[php \(4\) \(/tags/#php\)](#)

[raspberrypi \(1\) \(/tags/#raspberrypi\)](#)

[roboshell-backup \(4\) \(/tags/#roboshell-backup\)](#)

[ruby \(1\) \(/tags/#ruby\)](#)

[simpleleague \(1\) \(/tags/#simpleleague\)](#)

[source-control \(4\) \(/tags/#source-control\)](#)

[sql-server \(1\) \(/tags/#sql-server\)](#)
[truecrypt \(1\) \(/tags/#truecrypt\)](#)
[unit-testing \(3\) \(/tags/#unit-testing\)](#)
[vba \(2\) \(/tags/#vba\)](#)
[visual-studio \(3\) \(/tags/#visual-studio\)](#)
[web \(7\) \(/tags/#web\)](#)
[web-api \(1\) \(/tags/#web-api\)](#)
[wordpress \(2\) \(/tags/#wordpress\)](#)
[wtf \(1\) \(/tags/#wtf\)](#)

Copyright 2012-2019 Christian Specht

[Imprint / Impressum \(/imprint/\)](#) | [Data protection / Datenschutzerklärung \(/data-protection/\)](#)

Powered by Jekyll (<http://jekyllrb.com/>). Built with Bootstrap (<http://getbootstrap.com/>).

Cosmo (<http://bootswatch.com/cosmo/>) theme by bootswatch.com (<http://bootswatch.com/>).