

# PESTER CHEAT SHEET

## Creating a test fixture

### Syntax

```
New-Fixture -path -name
```

Generates two scripts, one that defines a function and another one that contains its tests.

-path: The path where the scripts are created

-name: The name of the scripts

### Example

```
New-Fixture -path c:\myScript -name Deploy
```

This generates Deploy.ps1 and Deploy.Tests.ps1, both in c:\myscript

## Executing test

### Syntax

```
Invoke-Pester -testname -strict -tag
```

Executes tests and provide output to the console.

-testname The name of the describe block to test

-strict All tests will either pass or fail. No inconclusive tests

-tag Comma separated list of test tags. See "Describe"

### Example

```
Invoke-Pester -testname "MyContext" -tag Unit, Integration -strict
```

See more parameters in the wiki

```
Invoke-Pester .\ script.Tests.ps1 -CodeCoverage .\script.ps1
```

## Test structure

### Describe – Context – It

The general structure of a Pester test is following this syntax:

```
Describe "MyDescription" -tag "MyTag" {
  Context "MyContext" {
    It "MyContext" {
      $false | should be $true
    }
  }
}
```

### Describe:

-tag Tags can be used to group the tests.

### It :

-skip Skips the test

-pending The test is not finished so result is not reliable.

## Assertion with Should

### Be

Not case sensitive.

```
"FOO" | Should Be "foo" #pass
```

### BeExactly

Case sensitive.

```
"FOO" | Should Be "FOO" #pass
```

```
"FOO" | Should Be "foo" #fail
```

### Exist

Checks if the object is present in a PS Provider.

```
$actual=(Dir .)[0].FullName
```

```
Remove-Item $actual
```

```
$actual | Should Exist # Fail
```

### Contain

Checks if a file contains a text. Not case sensitive. Uses RegEx

```
Set-Content -Path TestDrive:\file.txt -Value 'I am a file'
```

```
'TestDrive:\file.txt' | Should Contain 'I Am' # Pass
```

```
'TestDrive:\file.txt' | Should Contain '^I.*file$' # Pass
```

```
'TestDrive:\file.txt' | Should Contain 'I Am Not' #Fail
```

### ContainExactly

Checks if a file contains a text. Case sensitive. Uses RegEx

```
Set-Content -Path TestDrive:\file.txt -Value 'I am a file.'
```

```
'TestDrive:\file.txt' | Should ContainExactly 'I am' # Pass
```

```
'TestDrive:\file.txt' | Should ContainExactly 'I Am' # Fail
```

### Match

Uses a RegEx to compare. Not Case Sensitive

```
"I am a value" | Should Match "I Am" # Pass
```

```
"I am a value" | Should Match "I am a bad person" # Fail
```

### MatchExactly

Uses a RegEx to compare. Case Sensitive

```
"I am a value" | Should MatchExactly "I am" # Pass
```

```
"I am a value" | Should MatchExactly "I Am" # Fail
```

### Throw

Checks if expression is thrown

```
{ foo } | Should Not Throw #Fail
```

```
{ $foo = 1 } | Should Not Throw #Pass
```

```
{ throw "This is a test" } | Should Throw "This is a test" #Pass
```

```
{ throw "bar" } | Should Throw "This is a test" #Fail
```

**Warning:** The input object must be a ScriptBlock, otherwise it is processed outside of the assertion.

### BeNullOrEmpty

Checks values for null or empty (strings).

```
$null | Should BeNullOrEmpty # Pass
```

```
$null | Should Not BeNullOrEmpty # Fail
```

```
@() | Should BeNullOrEmpty # Pass
```

```
"" | Should BeNullOrEmpty # Pass
```

# Mocking

## Mock

This creates new behavior for any existing command within the scope of a Describe or Context block. The function allows you to specify a script block that will become the command's new behavior.

### -CommandName

The name of the command to be mocked.

### -MockWith

A ScriptBlock specifying the behavior that will be used to mock CommandName. The default is an empty ScriptBlock.

### -Verifiable

When this is set, the mock will be checked when Assert-VerifiableMocks is called.

### -ParameterFilter

Limit mocking behavior only to usages of CommandName where the values of the parameters passed to the command pass the filter.

This ScriptBlock must return a boolean value.

### Examples

```
Mock Get-ChildItem { return @{FullName =  
"A_File.TXT"} } -Verifiable -ParameterFilter { $Path  
-and $Path.StartsWith($env:temp) }
```

## Assert-VerifiableMocks

Checks if any Mock marked with -Verifiable has not been invoked. If so, this will throw an exception

## Assert-MockCalled

Checks if a Mocked command has been called a certain number of times and throws an exception if it has not.

**-CommandName:** The mocked command whose call history should be checked.

**-Times:** The number of times that the mock must be called to avoid an exception from throwing.

**-Exactly:** If this switch is present, the number specified in Times must match exactly the number of times the mock has been called. Otherwise it must match "at least" the number of times specified. If the value passed to the Times parameter is zero, the Exactly switch is implied.

**-ParameterFilter:** An optional filter to qualify which calls should be counted. Only those calls to the mock whose parameters cause this filter to return true will be counted.

**-Scope:** An optional parameter specifying the scope in which to check for calls. By default, Assert-MockCalled will find calls to the mocked command in the current Context block, or the current Describe block. Valid values are Describe, Context and It. If you use a scope of Describe or Context, the command will identify all calls to the mocked command in the current Describe / Context block, as well as all child scopes of that block.

### Examples

```
Assert-MockCalled -CommandName Set-Content -  
Times 2 -ParameterFilter { $path -eq  
"$env:temp\test.txt" }
```



Pester Cheat Sheet by Kaj Bonfils is licensed under a Creative Commons Attribution 4.0 International License.

Please feel free to send suggestions for updates and changes on [@kajbonfils](https://twitter.com/kajbonfils)

For more tips, tools and videos, please visit my blog at: <https://www.kajbonfils.com/>

Based on Pester 3.0  
<https://github.com/pester/Pester/wiki/>

<https://www.kajbonfils.com/>  
Based on Pester 3.0

<https://www.kajbonfils.com/>  
For more tips, tools and videos, please visit my blog at:

[@kajbonfils](https://twitter.com/kajbonfils)