



Windows PowerShell Get-Help on Cmdlet 'Assert-VerifiableMocks'

PS:\>Get-HELP Assert-VerifiableMocks -Full

NAME

Assert-VerifiableMocks

SYNOPSIS

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

SYNTAX

Assert-VerifiableMocks [<CommonParameters>]

DESCRIPTION

This can be used in tandem with the -Verifiable switch of the Mock function. Mock can be used to mock the behavior of an existing command and optionally take a -Verifiable switch. When Assert-VerifiableMocks is called, it checks to see if any Mock marked Verifiable has not been invoked. If any mocks have been found that specified -Verifiable and have not been invoked, an exception will be thrown.

PARAMETERS

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see [about_CommonParameters](https://go.microsoft.com/fwlink/?LinkID=113216) (<https://go.microsoft.com/fwlink/?LinkID=113216>).

INPUTS

OUTPUTS

----- EXAMPLE 1 -----

```
PS C:\>Mock Set-Content {} -Verifiable -ParameterFilter {$Path -eq "some_path" -and $Value -eq "Expected Value"}
```

```
{ ...some code that never calls Set-Content some_path -Value "Expected Value"... }
```

Assert-VerifiableMocks

This will throw an exception and cause the test to fail.

----- EXAMPLE 2 -----

```
PS C:\>Mock Set-Content {} -Verifiable -ParameterFilter {$Path -eq "some_path" -and $Value -eq "Expected Value"}
```

```
Set-Content some_path -Value "Expected Value"
```

Assert-VerifiableMocks

This will not throw an exception because the mock was invoked.

RELATED LINKS