Convert your Windows PowerShell module so its compatible with PowerShell Core 6.x

Wojciech Sciesinski
email: wojciech@sciesinski.net

ITpraktyk

2018

Convert your Windows PowerShell module so its compatible with PowerShell Core 6.x

Wojciech Sciesinski

Agenda

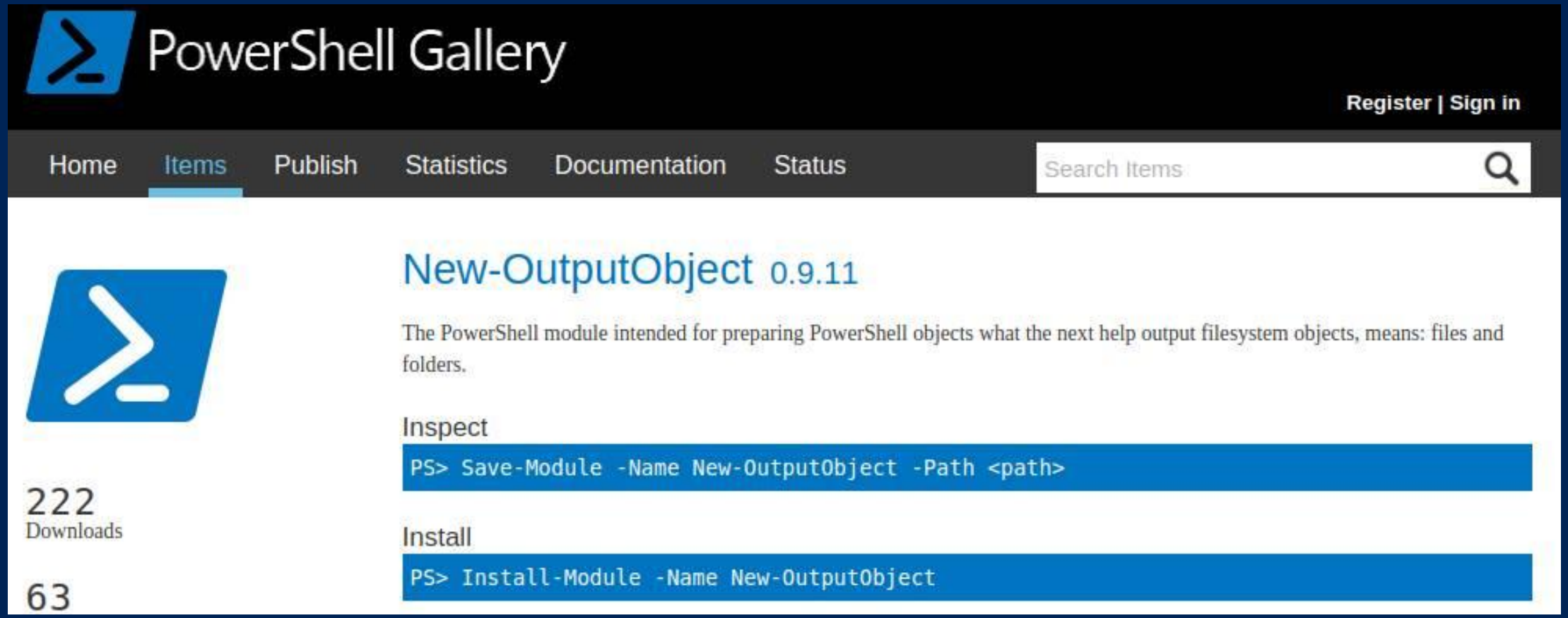
- Why me?
- What does PowerShell Core 6.x compatibility mean?
- Why? Do you really need this?
- Where we are? How to recognize the running environment?
- What are? The main areas of incompatibility
- Does this work? How to test?
- Other gotchas
- Finally!

Why me?

- I have done the work to adapt the Pester PowerShell module to work on PowerShell Core 6.x
  http://bit.ly/2qC7OfH and http://bit.ly/2qD2Qzg
- I have made the New-OutputObject PowerShell module
- I have done some work to adapt the PScribo module to be compatible with PowerShell Core 6.x

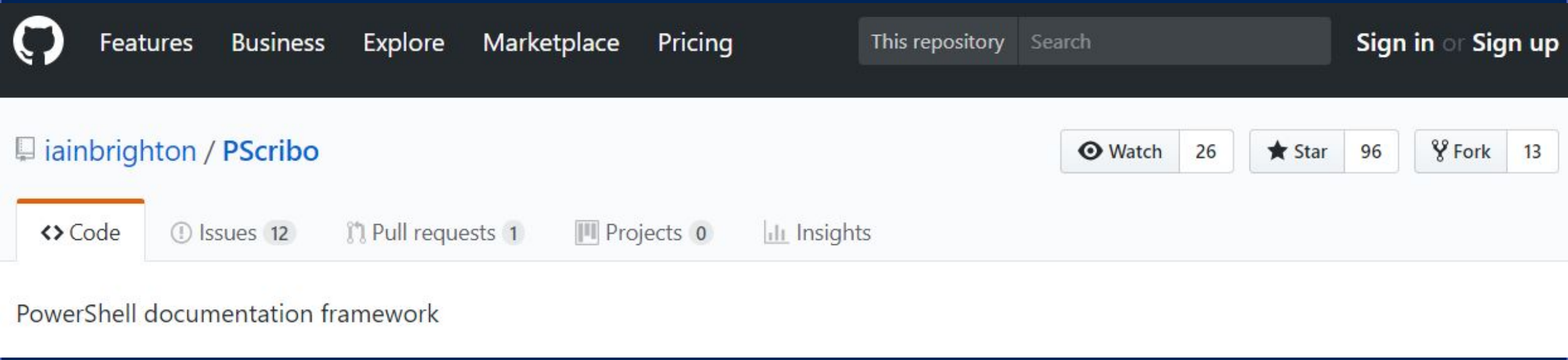
Why me?

- I have made the New-OutputObject PowerShell module



Why me?

- I have done some work to adapt the PScribo module to be compatible with PowerShell Core 6.x



Who else?

- Who else contributed to adapt Pester for PSCore 6.x ?

Felix Becker

felixfbecker

Gavin Eke

GavinEke

Joel Bennett

Jaykul

Edmund Dipple

elmundio87

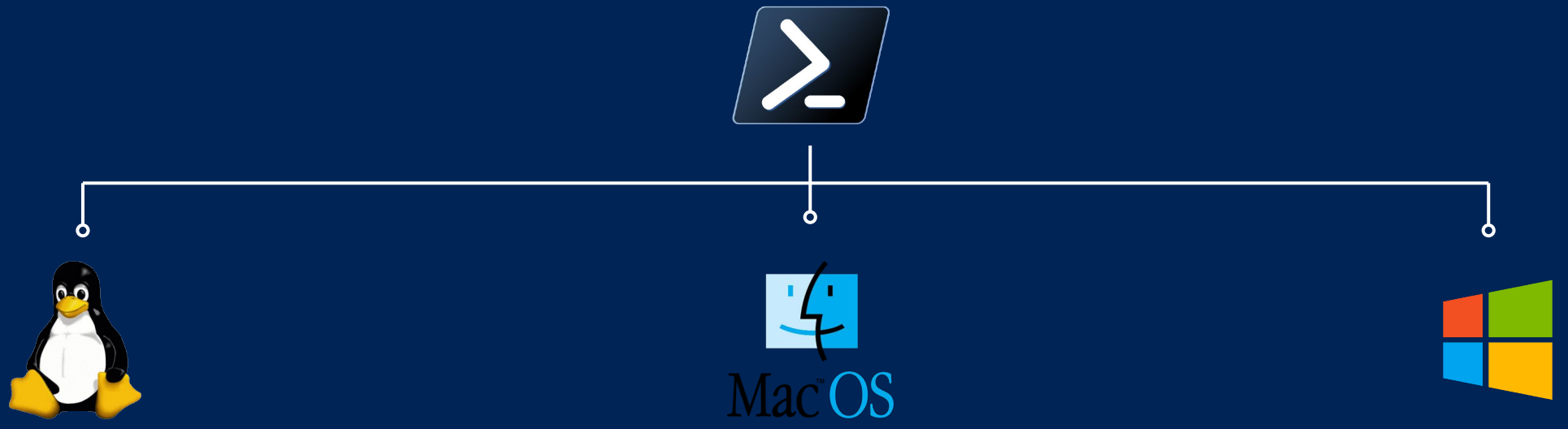
Jakub Jareš

nohwnd

In alphabetical order

What does PowerShell Core 6.x compatibility mean?

- Can be run using Windows PowerShell
- Can be run using PowerShell Core 6.x on:



OS's are listed in the alphabetical order :-P

- Work properly (as expected) on all supported operating systems

Why?

- To reuse your or your team skills
- To extend existing workflows (e.g. a continues integration) for Linux or unify used tools
- The use of CI platforms free features e.g. https://travis-ci.org/
- Or to do everything for which PowerShell Core "is optimized for dealing with structured data (e.g. JSON, CSV, XML, etc.), REST APIs, and object models."[1]

1. The quote from the PowerShell project description on GitHub https://github.com/powershell/powershell

Can I?

- Is your module dependent on from other modules and tools?
  - Check the PowerShell Gallery and search using the tag “PSEdition_Core” http://bit.ly/2qBx2e1
  - Examine a project/tool documentation
  - Try find a replacement
- Read “Known Issues” at the PowerShell project site http://bit.ly/2IXiY5l
- Do you really have to reinvent a wheel? Maybe you can use/extend Ansible? ;–) –> please attend my second session on Friday “Use your PowerShell skills to extend Ansible workflows. Create your own Ansible module for Windows platform”.

Where are we?

- PowerShell way – the Get-EnvironmentInformation function http://bit.ly/2qBppDs creates the dedicated variable $Environment
- Pester way – the GetPesterOS function returns the name of the OS run http://bit.ly/2H8tDcE
- Direct check of $PSVersionTable and defined variables values

Demo - 1

How to detect an environment?

What are? The main areas of incompatibility – Filesystem

```
PS C:\Windows\system32> Get-ChildItem c:\

    Directory: C:\

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        1/22/2018  11:29 AM                PerfLogs
d-r---        4/17/2018   2:25 AM                Program Files
d-r---        1/22/2018  11:24 AM                Program Files (x86)
d-r---        1/22/2018  11:29 AM                Users
d----l        3/10/2018  10:00 AM                vagrant
d-----         4/1/2018   8:03 AM                Windows
```

```
PS /home/wojtek> Get-ChildItem /

    Directory: /

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d----l         8/3/17   1:32 AM                bin
d-r---        4/15/18  11:21 PM                boot
d-r---        4/17/18  11:00 AM                dev
d-r---        4/17/18  11:00 AM                etc
d-r---         8/3/17   1:32 AM                home
d----l         8/3/17   1:32 AM                lib
d----l         8/3/17   1:32 AM                lib64
d-----        9/30/16   6:40 PM                lost+found
d-r---         8/3/17   1:32 AM                media
d-r---        4/15/18  11:23 PM                misc
```

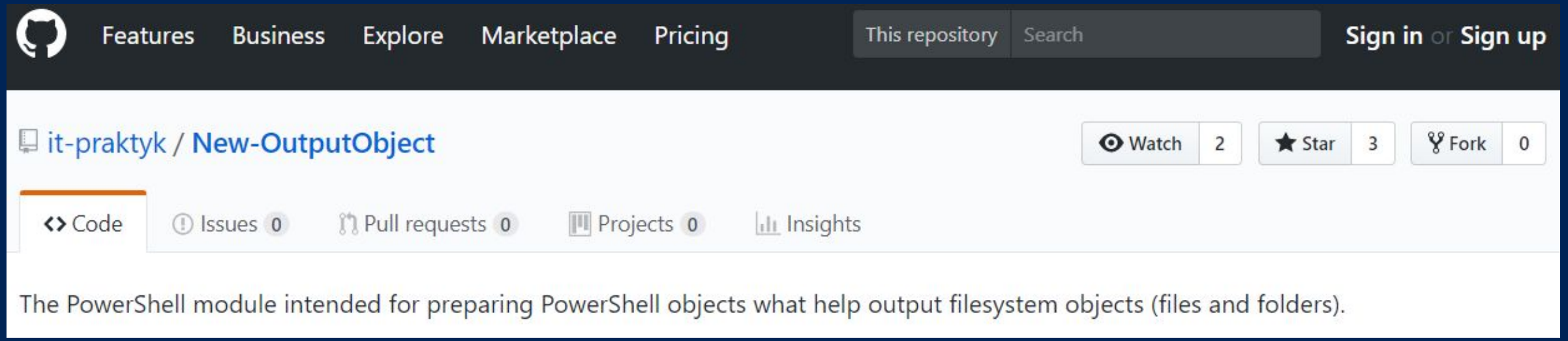
- Use methods provided by .Net Core
- Use declared variables
- Use dedicated cmdlets

Demo – 2

What are? The main areas of incompatibility – Filesystem

What? - Output - I

The New-OutputObject PowerShell module



Demo – 3

What are? The main areas of incompatibility – Output

What are? The main areas of incompatibility – Output – II

- End of Lines (EOL)
  - CR (Linux), CRLF (Windows)
  - can/should be converted? http://bit.ly/2vjGszB
  - you can use the unix2dos tool
- Control characters
  - `n , `r
  - special cases ? – PSCribo

What are? The main areas of incompatibility – Output – II

```
Describing OutText.Internal\OutTextTOC
  [-] outputs TOC name and section break 238ms
    Expected regular expression '^Table of contents\r
    -+' to case sensitively match 'Table of contents
    ----------------------------------------------------------------------------------------------------
    Heading 1
     Heading 2
    ', but it did not match.
    359:              $result | Should MatchExactly $expected;
    at Invoke-LegacyAssertion, /usr/local/share/powershell/Modules/Pester/4.3.1/Functions/Assertions/Should.ps1: line 188
    at <ScriptBlock>, /home/wojtek/Scripts/PScribo/Tests/Plugins/OutText.Tests.ps1: line 359
  [-] adds section numbers when "EnableSectionNumbering" is enabled (#20) 194ms
    Expected regular expression '^Table of contents\r
    -+\r
    1\s+Heading 1\r
    1.1\s+Heading 2\r
    $' to match 'Table of contents
    ----------------------------------------------------------------------------------------------------
    1    Heading 1
    1.1   Heading 2
    ', but it did not match.
    379:              $result | Should Match $expected;
    at Invoke-LegacyAssertion, /usr/local/share/powershell/Modules/Pester/4.3.1/Functions/Assertions/Should.ps1: line 188
    at <ScriptBlock>, /home/wojtek/Scripts/PScribo/Tests/Plugins/OutText.Tests.ps1: line 379
```

Demo – 4

What are? The main areas of incompatibility – Output – control characters

What are? The main areas of incompatibility – OS specific functions and tools or lack of them

- Do inaccessible feature everything?
  - replace them
  - use “hacks”
  - skip if the feature is not critical

Demo – 5

What are? The main areas of incompatibility
OS specific functions and tools or lack of them

Does this work? How to test?

- If you don't have tests yet write them now
- Tests also have to be update for compatibility!
- Test using CI solutions – you can use Travis-CI to tests your open source code on Linux and macOS for free
- Read Doug's Finke blog post "PowerShell – How to test your scripts for Windows, Linux and macOS" http://bit.ly/2H9jWPw

Other gotchas

- Case sensitivity http://bit.ly/2qG3ZVT
- .gitconfig http://bit.ly/2qH69Fm or not http://bit.ly/2H8g3WN
- A direct sharing of the repository folder between OS's can be tricky you can modify the setting ' filemode = false' in the file .git/config

Demo – 6

Gotchas

Finally!

- Update tags in the module manifest and add
  - 'PSEdition_Core'
  - 'PSEdition_Desktop'
- Find modules with the ‘PSEdition_Core’ tag in the PowerShell Gallery
  http://bit.ly/2qBx2e1

Demo – 7

Run some tests

Summary

- Adapting of the existing code is possible
- That work requires
  - tests
  - knowledge about
    - the code
    - operating systems
  - brain work - as an opposite to manual ;-)

Next Steps

- Now: 30 min break (if we are on time)
- Next: Workshops (17:00 - 20:00)
  - The Power of the Dark Side: Offensive PowerShell - Track 1
  - PlatyPS: From 0 to 1 and beyond - Track 2
  - Bring Existing Infrastructure under Code Control - Track 3

Questions?

about_Author

- https://www.linkedin.com/in/sciesinskiwojciech
- Twitter: @ITpraktyk
- GitHub https://github.com/it-praktyk

Building software for tomorrow's world

Many of our solutions
are used in everyday life, from
smartphones and TVs to cars and intelligent buildings.

Automotive

Mobile

Semiconductor

FinTech

Marine,
Aviation
and Defence

TV &
Digital
Broadcasting

Medical

Space

MOBICA IN NUMBERS

12 locations worldwide

In 14 years
over 400+ customers

Over 70%
projects in R&D

Over 120
staff bonding events per year

Over 1000 employees
and still growing

Over 1000 projects
completed

Over 150 QA Testing Engineers
experience in over 200 projects

Over 800 engineers
and more to come!

Is there something else you would like to know?

mobica.com

facebook.com/MobicaLimited

linkedin.com/company/mobica