Why You Should -Be Using Pester!

Intro

- Senior Systems Engineer for an Insurance Company
- twitter.com/jwmoss ••
- github.com/jwmoss
- Using Powershell for 2 years
- Live outside Raleigh, North Carolina
- Published in a book about a tweet I posted in 2011 about Steve Jobs







What is Pester?

- A testing framework written in Powershell
 - Unit tests: test a piece (unit) of your code, don't touch external system.
 - Integration tests: reach out to external system and tests your code.
- A way to check the thing you're doing in Powershell is actually the thing you meant to do in Powershell.
- Forces you to re-think how you write Powershell.
- Runs on Linux, macOS, or Windows.
- Comes pre-installed with Windows 10 and Windows Server 2016.
- Microsoft wants you to use this tool, read the PDF https://docs.microsoft.com/en-us/powershell/dsc/further-reading/whitepapers
- WARNING: This presentation may turn you more into a developer than you thought was possible.



Demo!

Pester Use-Case 1

- On-boarding New Users:
 - AD Account exists
 - UPN matches email address
 - UPN matches sip address
 - Company, Department, EmployeeID, Title, City, Office, State, and Home Drive is not empty
 - Verify common AD Group memberships are set
 - Azure AD Group-Based licensing + firewall access
 - Location-based AD Groups are set
 - O365 licensing is set.

```
PS C:\Users\jmoss-admin> Test- User
Already connected to enant
Enter new users's first name: Jonathan
Enter users's last name: Moss
Mobile User? [Y] or 「№1・∨
[1] CN=Moss\, Jonat
[2] CN=Moss\, Jonat
Select the correct
Describing User Validation
  Context UPN Sanity Checks
       UPN matches EmailAddress 12ms
    [+] UPN matches SIP Address 18ms
  Context User Profile Information
      Company is Set 34ms
       Title is Set 7ms
        Manager is Set 11ms
        Profile is accurate 11ms
      Expected strings to be the same, because the profile path is required, but they were different. Expected length: 35
Actual length: 30
Strings differ at index 25.
Expected: '\SUSERNAME%'
  Context Email Sanity Check
  Context Location Information
     +] Country is Set 36ms
     +] Office is Set 9ms
    [+] State is Set 9ms
  Context AD Group Membership Validation
      l Mailbox License Group configured for Jonathan Moss 49ms
     +] 0365 Firewall Access Group is allowed for Jonathan Moss 25ms
      l AppVol Core Apps are assigned to Jonathan Moss 10ms
        UEM is enabled for Jonathan Moss 10ms
      RSA Group is set for Jonathan Moss 10ms
      ] Windows 10 VM is provisioned for Jonathan Moss 11ms
    [+] Folder Redirection is set for Jonathan Moss 9ms
  Context Office 365 Validation
     F] 0365 User Sync Status 37ms
       Licensed Status is set 8ms
        Skype Online Status is Enabled 7ms
        Skype On-Prem Hosting Provider is sipfed.online.lync.com 11ms
```

Pester Use-Case 2

- Creating new Servers:
 - AD Description is set
 - Exists in DNS
 - In the correct OU
 - Correct network based on pre-defined rules
 - Validate vCenter portgroup
 - Validate it's running/turned on
 - Validate base software installed (AV, Solarwinds, etc).

Other Uses of Pester

- PSKoans https://github.com/vexx32/PSKoans COMING SOON
 TO A USERGROUP NEAR YOU!
- DBAchecks (SQL Server Validation) https://github.com/sqlcollaborative/dbachecks
- Operation-Validation-Framework (Infrastructure Validation) https://github.com/PowerShell/Operation-Validation-Framework
- Vester (Validate vSphere Configuration) https://github.com/
 WahlNetwork/Vester

More Uses of Pester

- Validating Active Directory Attributes https://mattmcnabb.github.io/
 pester-lob
- Validating DNS Records https://pshirwin.wordpress.com/2015/11/06/
 pester-script-to-test-dns-configuration/
- Skype/Lync Infrastructure Validation https://sysnetdevops.com/2017/06/05/testing-infrastructure-with-pester/
- Domain Controller Validation https://github.com/EvotecIT/
 PesterInfrastructureTests
- Citrix/XenApp Validation http://citrixlab.dk/archives/812
- Software deployment readiness check https://github.com/
 MathieuBuisson/DeploymentReadinessChecker

Credit where Credit's Due

- Mark Wragg (https://github.com/markwragg | @markwragg)
- Jakob Jares, one of the main developers of Pester (http://jakubjares.com/ | @nohwnd)
- Reddit https://www.reddit.com/r/PowerShell/search?
 q=Pester&restrict_sr=1&sort=top
- Pester Wiki https://github.com/pester/Pester/wiki
- DBAChecks PS Module https://github.com/sqlcollaborative/dbachecks
- The first 10 pages of Google Results for "Powershell Pester"
- The Pester Book by Adam Bertram (https://leanpub.com/pesterbook)

"I really wish there was a way to test or validate my Powershell code."

-Said No One Ever (Hopefully after this talk)

Questions?