PSDAY.UK

PowerShell Core

Iain Brighton
22nd September 2017

Iain Brighton

- Principal Consultant @ Virtual Engine
    - Automation and end-user computing focused company
- RES Certified Trainer (RCT) and Valued Professional (RSVP)
- Amateur developer and PowerShell “junkie”
    - Community DSC resource contributor
    - Lability and PScribo author

Agenda

- .NET What?
- PowerShell Core
- Windows PowerShell
- Demo time!
- Desired State Configuration
- Wrap Up
- Questions

.NET What?

- .NET Framework
  - The "full" or "traditional" flavour of .NET that's distributed with Windows
- .NET Core
  - Cross-platform .NET that can run on Windows, Mac, and Linux
  - Subset of the “full” .NET framework API surface
  - Portable installation
- .NET Standard
  - Avoids recompiling for different targets, i.e. PCLs
  - Common API interface that is supported cross-platform

.NET Distribution

.NET FRAMEWORK	.NET CORE	XAMARIN
Platform for .NET applications on Windows	Cross-platform and open source framework optimized for modern app needs and developer workflows	Cross-platform and open source Mono-based runtime for iOS, OS X, and Android devices
Distributed with Windows	Distributed with app	Distributed with app

What is PowerShell Core?

- Open-source, cross-platform PowerShell
    - For Windows, Mac and Linux
    - Written on top of .NET Core
- Full PowerShell language and remoting support
- Support for all cmdlet types C#, script and CIM
    - Support for classes
- **Is the future of PowerShell**
    - Except for Workflows ☺

Why PowerShell Core?

- **Azure, Azure, Azure!**
  - Manage Azure from Windows, Mac or Linux
  - Reduce adoption friction
- Decouple from Windows Management Framework
  - And WMI
- Provide cross-platform management tooling
  - VSCode

Windows PowerShell

- Windows PowerShell 5.1 will continue to be distributed in-box
  - Tied to Windows Management Framework
  - Based upon the “full” .NET Framework
- Active Windows PowerShell development is ceasing
  - High-priority fixes will still be made available
  - New features may be back-ported for compatibility
  - Windows PowerShell 5.1 may be the last ever release..

Pray to the demo installation Gods

New Variables

- $IsCoreCLR
- $IsLinux
- $IsMacOS ($IsOSX)
- $IsWindows
- $PSVersionTable.GitCommitId
- $PSVersionTable.OS
- $PSVersionTable.Platform

"Gotchas"

- PowerShell.exe default is now -File
- Linux aliases removed on Linux!
- File encoding and line endings
- No Windows-specific cmdlets
  - Get-Service
  - Get-LocalUser
- No $PSVersionTable.BuildVersion/CLRVersion

Windows Subsystem for Linux

- Beta tag removed with the “Fall Creators Update” (1709)
  - Problems with Conhost.exe implementation in 1703
- Universal installation script works
- Manage WSL
  - Requires “Developer Mode” enabled
  - Requires Windows Subsystem for Linux feature installed
    - Get-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
  - lxrun.exe /install
  - lxrun.exe /uninstall

Windows PowerShell Remoting

- WSMan
  - Industry-standard communication protocol
  - WinRM is Microsoft's implementation of WSMan
  - SOAP/XML messaging/transport
  - Supports additional encryption with SSL/HTTPS
- PowerShell Remoting Protocol (PSRP)
  - Runs on top of WinRM
  - Stateful protocol for instantiating remote PowerShell instances
  - Sending and receiving encrypted pipeline data
- Linux requires OMI server and PSRP provider registration

PowerShell Core Remoting

- New SSH transport option
  - PSRP over SSH
  - Removes the WinRM/OMI configuration
  - WinRM/OMI can still be implemented
  - Remote endpoint configuration and JEA not currently supported
- Requires OpenSSH client installation on Windows
  - To permit remoting from Windows to Linux
  - Also requires SSH server configuration on Linux
- Requires OpenSSH server installation on Windows
  - To permit remoting from Linux to Windows

Pray to the PowerShell Remoting Gods

DSC Core

- Soon-to-be-released version that aligns with PowerShell Core
- No dependency on WMI or WMF
- Xcopy-able package
- Supports resources written in native C/C++ (no WMI), Python and PowerShell
- Runs on Windows and Linux
- Required (includes in package) PowerShell Core and .NET Core

Wrap Up

- Target PowerShell Core for future development
  - Windows PowerShell is going away, albeit slowly
  - Consider cross-platform differences/quirks/caveats
- Look at Win32 OpenSSH implementation
  - SSH key management will be important
- Desired State Configuration Core
  - Changes coming to core resources
  - tug as an alternate Pull Server implementation
  - MacOS support

Resources

- PowerShell 6.0 Roadmap
  - https://blogs.msdn.microsoft.com/powershell/2017/07/14/powershell-6-0-roadmap-coreclr-backwards-compatibility-and-more/
- DSC Future Direction
  - https://blogs.msdn.microsoft.com/powershell/2017/09/12/dsc-future-direction-update/
- PowerShell Remoting over SSH
  - https://github.com/PowerShell/PowerShell/tree/master/demos/SSHRemoting
- Universal PowerShell Core Installer
  - https://cloudywindows.io/post/the-universal-powershell-core-installer-install-powershell.sh/

Resources

- PowerShell remoting between Windows and Linux
  - https://4sysops.com/archives/powershell-remoting-between-windows-and-linux/
- Tug - Open-source, cross-platform Pull/Reporting Server
  - https://github.com/PowerShellOrg/tug
- Windows 10 Bash and PowerShell Redux
  - http://www.maxtblog.com/2017/01/windows-10-bash-and-powershell-redux/

Questions?

https://github.com/iainbrighton/PSDAYUK-PowerShell-Core

@iainbrighton

✉ iain.brighton@virtualengine.co.uk