



PowerShell Core 6.1 and Beyond

Steve Lee

Principal Software Engineer Manager
PowerShell/PSScriptAnalyzer/PSEditorServices

Azure Compute

Microsoft Corporation

<https://GitHub.com/PowerShell/PowerShell>



2018

PowerShell Core 6.1 and Beyond



Steve Lee

**Finally installed
PowerShell Core 6.0**

**PowerShell Core 6.1 is
the new hotness**

Agenda

- Roadmap
- New Investments - Demos
- PowerShell Standard
- Future Thoughts

Roadmap

- Dependency on .Net Core 2.1
- Continue Preview releases approx. ~3 weeks
- Target General Availability (GA) early July
- New minor releases approx. ~6 months

Amazing Community Support!

- PSCore6.1 Preview1 includes 129 PRs (changes)
 - 68 from the community!
- Thanks!

@markekraus

@bergmeister

@iSazonov

@kwkam

@KevinMarquette

@IISResetMe

@stknohg

@tandasat

@CallmeJoeBob

@seemethere

@kanjibates

@ppadmavilasom

@rpalo

@timothywlewis

@brianbunke

@thezim

@mababio

@CloudyDino

@DarqueWarrior

@strawgate

@Pawamoy

@Stanzilla

@tomconte

@zackJKnight

@kasper3

@powercode

@maybe-hello-world



Engineering Improvements

- Automation of Releases
- Nightly runs on all supported platforms
- Move to Pester 4.x
- Servicing via Microsoft Update
- Stable vs Preview release channels

New Investments (from PowerShell Team)

- Continue to address customer reported issues
- Experimental feature flags
- Improvements to help system
 - Markdown rendering
 - Refactoring as a module
 - Updateable help per user

Demo

Markdown Parsing and Rendering



PowerShell-Blog — pwsh • pwsh — 115x30

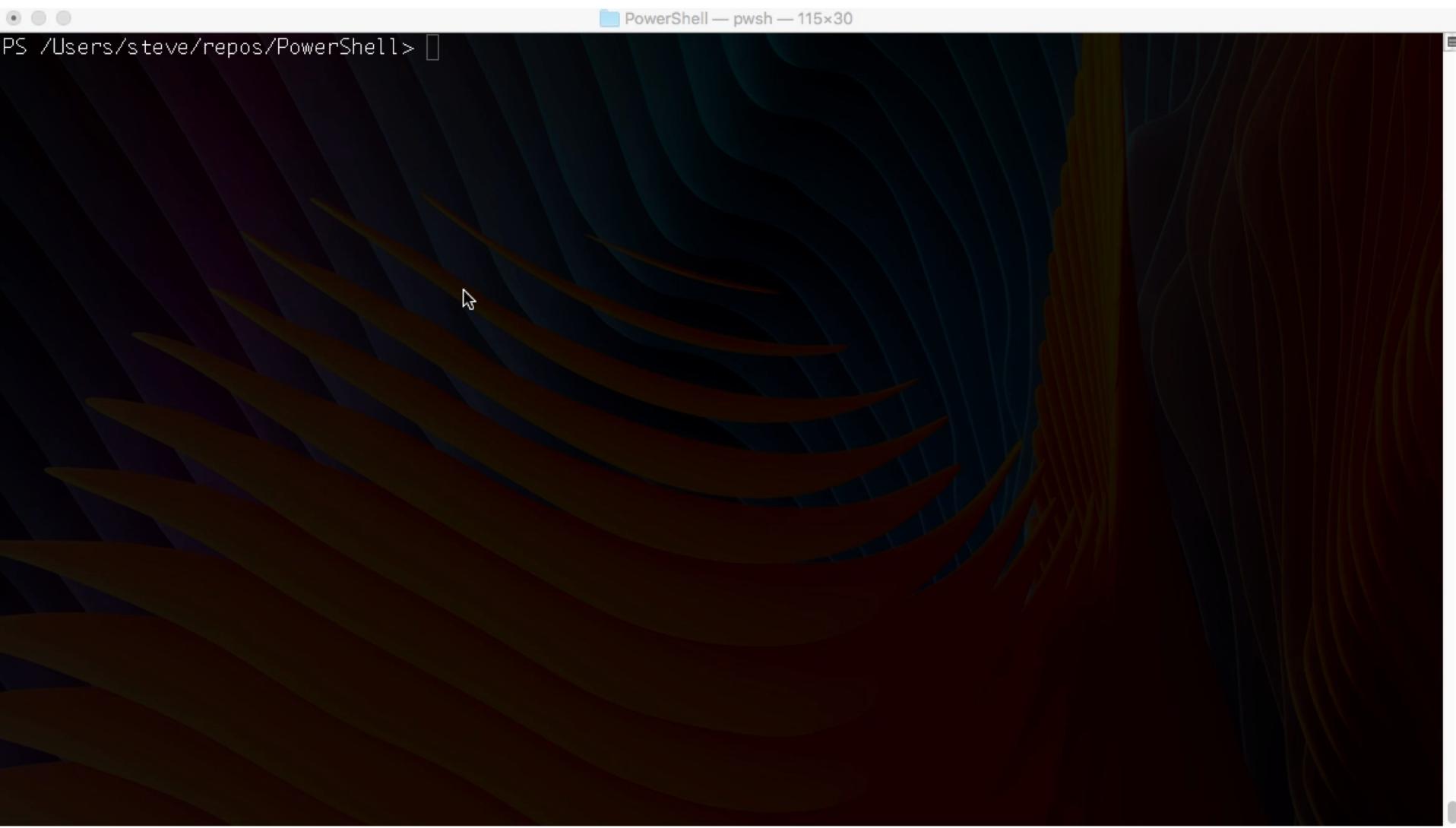
PS /Users/steve/repos/PowerShell-Docs> █



Demo

Per User Updateable Help





New Investments Continued

- DeviceGuard/AppLocker integration parity with Windows PowerShell
- Easier threading in PowerShell script
 - Considering different options

Demo

ThreadJob



PowerShell-Blog — pwsh — 115x30

PS /Users/steve/test> █

█



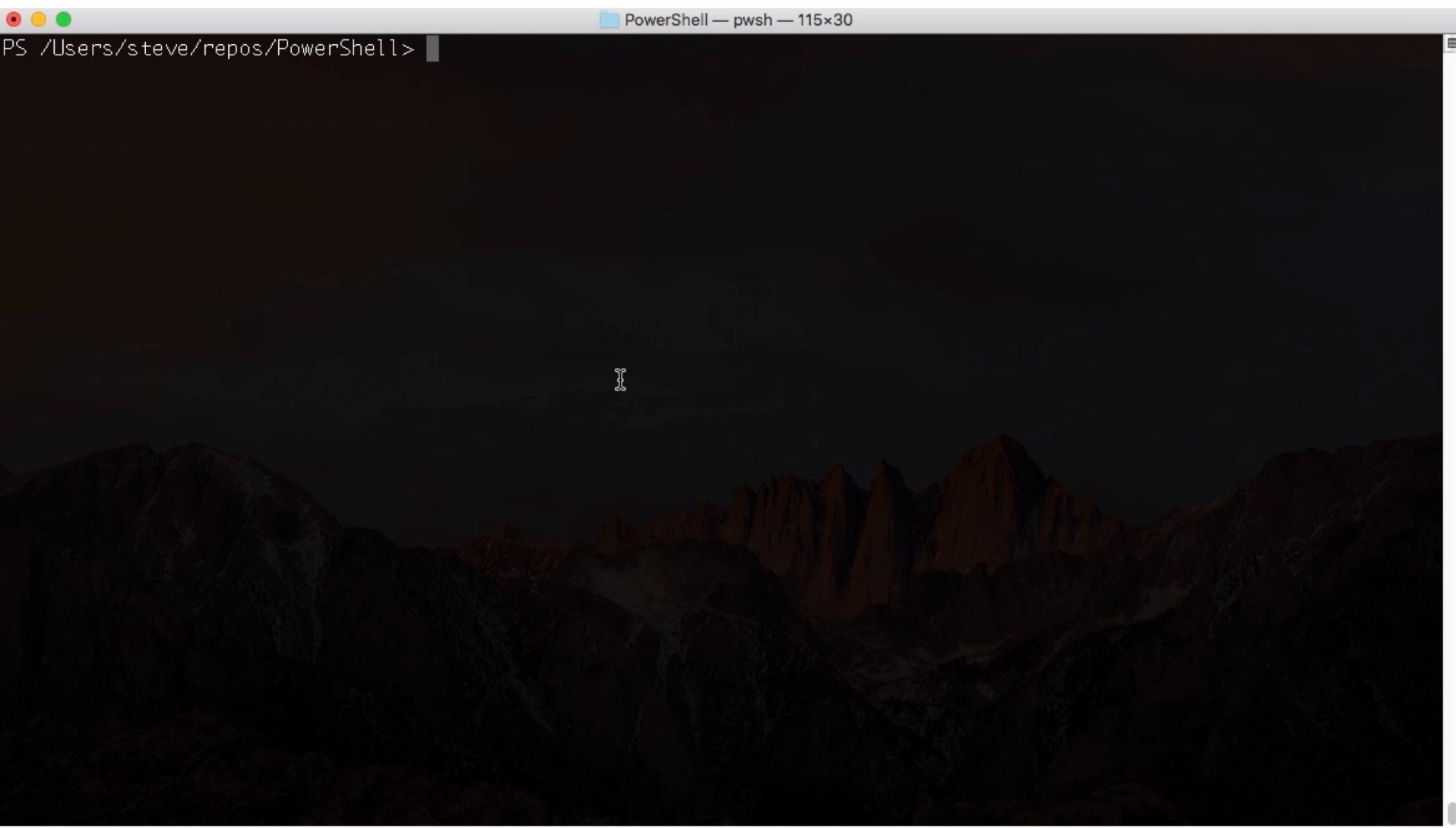
New Investments Continued

- PSReadline 2.0
- Continued ML experiments
- Continue enhancing PowerShell remoting over SSH

Demo

PowerShell Remoting Over SSH Improvements





PowerShell — pwsh — 115x30

PS /Users/steve/repos/PowerShell> █

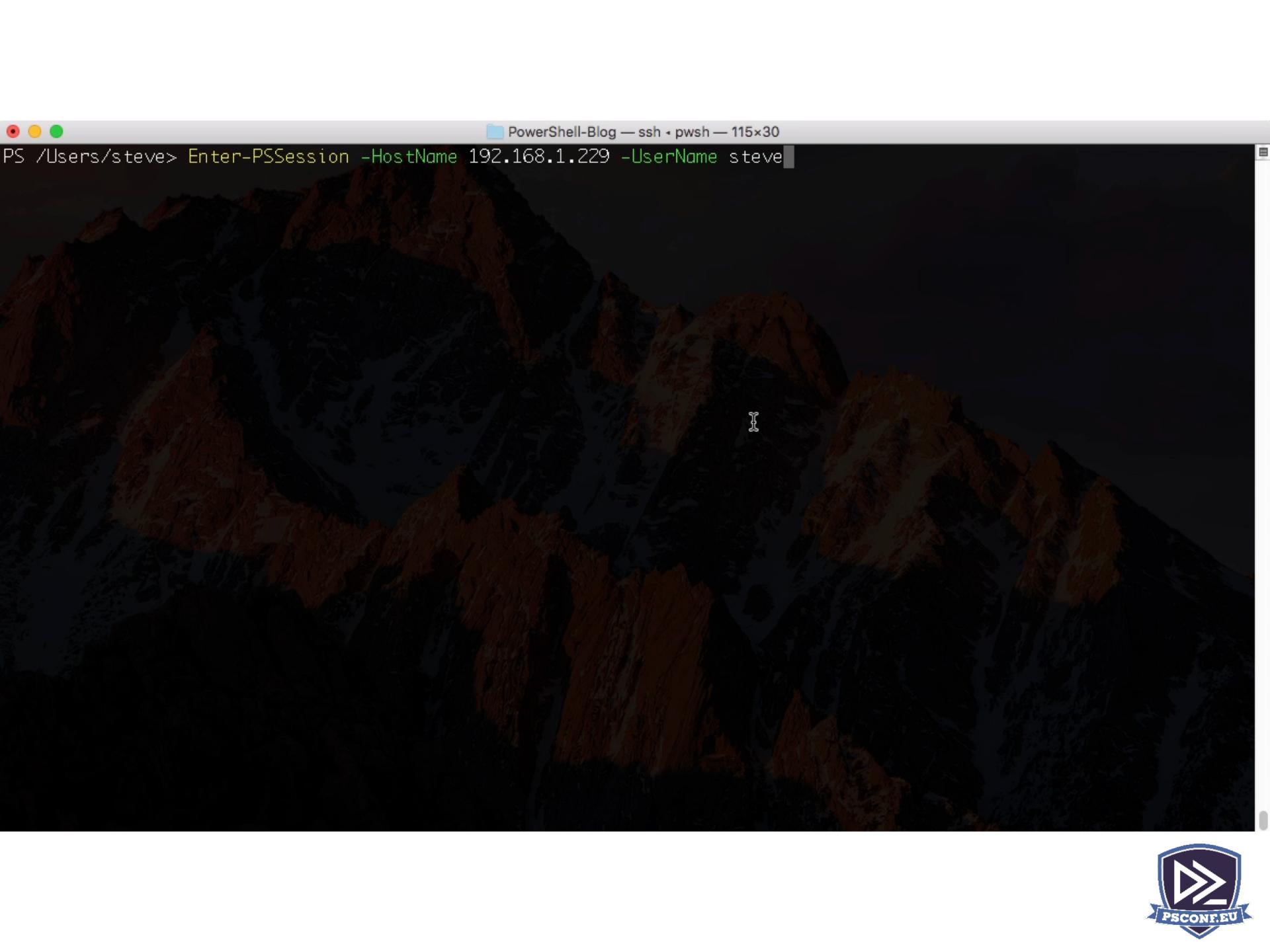
WindowsPowerShellCompatibilityPack

- <https://github.com/powershell/windowspowershellcompatibilitypack>
- Supercedes WindowsPSModulePath module
- Add Windows Compatibility Pack for .Net Core
- Leverage implicit remoting for calling Windows PowerShell cmdlets from PowerShell Core 6
- Windows PowerShell aliases

Demo

WindowsPSModulePath





```
PowerShell-Blog — ssh • pwsh — 115x30
PS /Users/steve> Enter-PSSession -HostName 192.168.1.229 -UserName steve
```

New Investments Continued

- [Alpine Linux](#)
- [PowerShell IoT](#)
- [Snapcraft](#)

Demo

PowerShell IoT module on Raspberry Pi-3





Terminal — pwsh — 105x31

PS /Users/steve> [



PowerShell Standard Library

- Why?
 - Build modules compatible with different versions of Windows PowerShell and PowerShell Core and cross platform compatible
- What?
 - Subset of APIs that are ensured to be forward compatible (each new version is a strict superset)
- How?
 - Target NetStandard2.0
 - Reference PowerShellStandardLibrary instead of System.Management.Automation.dll

PowerShell Standard Library

SMA.dll PSv6
SMA.dll PSv5.1

PS Std Lib v5.1

PowerShell Standard Library

- Explicitly removed
 - Custom PSSnapIn (superceded by Modules)
 - Workflow, Transactions, legacy Pipeline API, legacy Parser API
 - PerfCounters, Registry, CIM support (not cross platform)
- <https://github.com/powershell/powershellstandard>
- <https://www.nuget.org/packages/PowerShellStandard.Library>

PowerShell Core 6.2 and Beyond

- Community driven vs PowerShell Team driven
 - <http://aka.ms/psgithubbi>
 - ~50% of PRs from the community
 - Incremental improvements in all areas
 - RFCs authored by community members
 - Please use `Thumbs Up` on issues/RFCs to give us a sense of importance of specific issues/RFCs

PowerShell Core 6.2 future thoughts

- <https://github.com/PowerShell/PowerShell/projects>
- Ubiquity
 - Enable in Azure Functions and Azure Automation
 - Easily available or by default in Azure VMs
- Developer experience
 - PowerShell Classes improvements
 - Custom keyword/DSL support
 - GUI support (console/web/vscode)
 - Extensibility improvements to intellisense/tabcomplete

PowerShell Core 6.2 future thoughts – cont.

- Cloud scale readiness
 - Interactive vs execution agent
 - Performance improvements
 - Debugging/logging improvements
- Security
 - Foreign input tracking
 - Windows Device Guard audit mode

PSEditorServices/VSCode-PS future thoughts

- Continue to address customer issues
- Integration with Azure Functions and Azure Automation
- Vim integration
- Perf/reliability improvements
- Thanks to our community maintainers!
 - Keith Hill - @r_keith_hill
 - Patrick Meinecke - @SeeminglyScienc
 - David Wilson - @daviwil



PSScriptAnalyzer future thoughts

- Continue to address customer issues
- Enable PowerShell Script based rules
- Rules for script compatibility
 - Windows PowerShell vs PowerShell Core
 - Cross-platform
- Thanks to our community maintainer!
 - Christoph Bergmeister - @bergmeister

Summary

- PowerShell Core 6 updates frequently, adopt fix-forward
- Leverage your PowerShell skills everywhere: Windows, Linux, macOS, containers, cloud, IoT
- Build/update modules using PowerShell Standard Library
- Migrate from Windows PowerShell using WindowsPowerShellCompatibilityPack



Next Steps

- Now: 15 min break
- Grab a coffee
- Stay here to enjoy next presentation
- Change track and switch to another room
- Ask me questions or meet me in a breakout session room afterwards



Questions?

about_Author

- Worked in Systems Management space at Microsoft for over 17 years
 - WMI
 - WinRM
 - IPMI Provider/Driver
 - DMTF member
 - [Physical Computer System View Profile](#)
 - PowerShell
- Poker [Enthusiast](#)