



PSScriptAnalyzer

BY

CHRISTOPH BERGMEISTER

> whoami

- ▶ Christoph Bergmeister
- ▶ MSc. Computational Physics, then in Software for 3 years now at Willis Towers Watson
- ▶ Full & Cross Stack .Net (Core) with a DevOps focus (PowerShell/Azure/Docker/VSTS)
- ▶ Author of [posh-dotnet](#) and [posh-vsts-cli](#)
- ▶ Community Maintainer of [PSScriptAnalyzer](#) and contributor to the [PowerShell Core](#) repo
- ▶ GitHub: <https://github.com/bergmeister/>
- ▶ Twitter: <https://twitter.com/ChristophBergm>
- ▶ Blog: <https://bracketsandbraces.blogspot.co.uk/>



Ice Breaker & Presentation Theme

<https://github.com/bergmeister/cmatrix>

What is PSScriptAnalyzer?

- ▶ Open Source (MIT) and owned by MSFT
- ▶ Static Code Analysis
- ▶ Style Linter & Formatter
- ▶ Offers corrections
- ▶ Integrated into PowerShellEditorServices
 - VSCode, Sublime, Vim, etc
- ▶ Offer advice on PS Core compatibility (under development)
- ▶ Built using netstandard2.0 and net451
 - Cross Platform but dependency on PowerShell itself, especially its Language.Parser methods for getting the AST
 - More like netcore2.0 in practice

What does PSScriptAnalyzer not do?

- ▶ Help you to write correct syntax. This is PowerShellEditorServices.
- ▶ Warn you on any mistake. PSSA is not your mum
- ▶ Help you learn PowerShell
- ▶ Learning source for writing good code
→ <https://github.com/PoshCode/PowerShellPracticeAndStyle>
- ▶ Warnings sometimes do not trigger when they should and sometimes they do when they should not
- ▶ PSSA is clever but not intelligent

Why PSScriptAnalyzer?

- ▶ Learning rules means learning about best practices or gotchas
- ▶ Find bugs → prevent expensive debugging time.
- ▶ It's free!
- ▶ Code style rules
- ▶ Help review code changes

Demo

- ▶ VSCode integration and rule selection and .vscode
- ▶ Cmdlets usage (note to self: use Ubuntu VM)
<https://github.com/PowerShell/PSScriptAnalyzer#usage>
 - ▶ Get-ScriptAnalyzerRule
 - ▶ Invoke-ScriptAnalyzer
 - ▶ Invoke-Formatter
- ▶ Suppressing warnings: functions/scripts (param gotcha)
- ▶ CI example: <https://github.com/bergmeister/posh-profile/blob/master/test/psscriptanalyzer.tests.ps1>

Overview

Rules:

- ▶ <https://github.com/PowerShell/PSScriptAnalyzer/tree/development/RuleDocumentation>
- ▶ Known issues:
 - ▶ PSUseDeclaredVarsMoreThanAssignment
 - ▶ Parsing imported types:
<https://github.com/PowerShell/PSScriptAnalyzer/issues/850>
 - ▶ Docs → PRs welcome!
 - ▶ SSA (Single Statement Analysis) → e.g. for += on array → PR
- ▶ Help: GitHub or Slack: <https://powershell.slack.com>

My current work so far (available in vNext)

- ▶ Invoke-ScriptAnalyzer **-Fix**
- ▶ Invoke-ScriptAnalyzer **-EnableExit**
- ▶ Invoke-ScriptAnalyzer **-ReportSummary**
- ▶ Upgrade build to .Net Core 2.0 SDK
- ▶ Building and debugging with Visual Studio
- ▶ New rules:
 - ▶ **AvoidAssignmentToAutomaticVariable**
 - ▶ **PossibleIncorrectUsageOfAssignmentOperator**
- ▶ Warn on Get-* (added to **AvoidUsingCmdletAlias**)
- ▶ Various bug and documentation fixes. Especially PSUseDeclaredVarsMoreThanAssignment, PSOpenBrace and setting path parsing
- ▶ CI: PowerShell Core on Windows/Linux

How does it work?

- ▶ AST → Demo: Show-Ast from <https://www.powershellgallery.com/packages/ShowPSAst/1.0>

```
Show-Ast { if($a -eq "$($b.Wtf())"){ } foreach($i in $x){} }
```

- ▶ Custom rules:

Example:

```
Invoke-ScriptAnalyzer -ScriptDefinition 'write-Host' -CustomRulePath  
"C:\Users\cberg\git\London\January2018\ChrisDent_AdventuresInAST\AllR  
ules.psm1"
```

More examples:

https://github.com/powershellorguk/London/tree/master/January2018/ChrisDent_AdventuresInAST

More AST stuff: Find-Ast in VSCode and EditorAstProvider:
<https://github.com/SeeminglyScience/EditorAstProvider>

How to build, debug PSSA and consume it in VSCode

- ▶ Demo using GitHub (from scratch)

<https://github.com/PowerShell/PSScriptAnalyzer#from-source>

- ▶ Show VsCode integration in
C:\Users\cberg\.vscode\extensions\ms-vscode.powershell-1.6.0\modules

Future directions

- ▶ X-Platform support/testing getting more serious → Jim
- ▶ Rule/Doc fixes
- ▶ PSCore compatibility (like PowerShellStandard for scripts)
- ▶ New rules
- ▶ SSA (single statement analysis)

Thank you for your attention
and patience!

Any Questions?