

Plaster
Pester
Git & GitHub
VSTS

CONTENT

Learn And Understand Powershell Modules with Plaster How To Create Powershell Tests with Pester How & Why To Use Version Control Systems Like Git How To Setup Easy Continuous Deployment With VSTS

OBJECTIVE

UNDERSTAND

UNDERSTAND

What Is A Module?

Package

Distributable

Shareable

Reusable

Partition, Organize & Abstract PowerShell Code Into Self-Contained, Reusable Units

UNDERSTAND

Module Content

CODE

Functionality

CODE SUPPORT

Assemblies, Help Files, Or Scripts

MANIFEST FILE

Describes, Metadata, Author & Versioning

DIRECTORY

Contains Content



What Is Plaster?

Scaffold

Parametrization

Repeatable

Automate The Process To Create Modules

_ Module Framework

What Is Pester?

Verify Work

Design Guidance

Documentation

ScriptAnalyser & Help

Define What Is Expected & Prove It, Create Quality & Ensure Reusability

What Is Git?

Version Control Track Changes

Coordinate

Work Together

Work On A Module Like A Developer

What Is Azure DevOps?

Platform Software Development

Measure

Report

Data Storage

Collaborate

Source Control Work Item
Tracking

Build Server

Central Platform For Your Development,
Maintenance & Delivery Of Code



aka.ms/mark/pscd/code

DEMO



Steps

PowerShellGet https://docs.microsoft.com/en-us/powershell/gallery/readme

Git https://git-scm.com/

VSTS https://www.visualstudio.com/team-services/

Github https://github.com/

Install-Module Plaster https://github.com/PowerShell/Plaster

Install-Module -Name Pester https://github.com/pester/Pester

(Install-Module posh-git) https://github.com/dahlbyk/posh-git



<?xml version="1.0" encoding="utf-8"?>

Plaster Manifest

<plaster|Manifest schemaVersion="0.4" xmlns="http://www.microsoft.com/schemas/PowerShell/Plaster/v1">

<metadata></metadata>

<parameters></parameters>

<content></content>

</plasterManifest>

XSD Validation

Template Metadata

Module Parameter

Files, Folders Templates, Validation

```
<metadata>
 <name>NewPowerShellModule</title>
 <id>38bd80d9-3a47-4916-9220-bed383d90876</id>
 <version>0.2.0</version>
 <title>New PowerShell Module</title>
 <description>Plaster template for creating the files for a PowerShell module.</description>
 <author>Plaster Project</author>
 <tags>Module, ModuleManifest, Build</tags>
</metadata>
```

Metadata

```
<parameter name='Version'</pre>
                                                                                           Parameters
           type='text'
           default='0.1.0'
           prompt='Enter the version number for the module'/>
<parameter name='License' type='choice' default='2' store='text' prompt='Select a license'>
  <choice label='&amp;Apache'</pre>
          help="Adds an Apache license file."
                                                           type='multichoice' default='0,1,..'
          value="Apache"/>
  <choice label='&amp;MIT'</pre>
          help="Adds an MIT license file."
          value="MIT"/>
</parameter>
```

```
<file source='Tests\*.tests.ps1'

destination='test\'

condition='$PLASTER_PARAM_Options -contains "Pester"'/>
<templateFile source='test\Shared.ps1'

destination='test\Shared.ps1'

condition="$PLASTER_PARAM_Options -contains 'Pester'"/>
```

Content Elements

<message nonewline='true'>`n`nYour new PowerShell module project \$PLASTER_PARAM_ModuleName </message>

<newModuleManifest destination='src\\${PLASTER_PARAM_ModuleName}.psd1'</pre>

moduleVersion='\$PLASTER_PARAM_Version'

rootModule='\${PLASTER_PARAM_ModuleName}.psm1'

author='\$PLASTER_PARAM_FullName'

description='\$PLASTER_PARAM_ModuleDesc'

encoding='UTF8-NoBOM'

powershellVersion='5.1'/>

Manifest File

```
<%
  if ($PLASTER_PARAM_Ensure -eq 'Yes') {
     # Ensure the presence/absene of the resource."
     [ValidateSet('Present','Absent')]"
     [string]"
     `$Ensure = 'Present'"
%>
<%= $PLASTER_PARAM_ParameterName %>
```

Use Parameter

Use Parameters Inside <% %> To Template

Single Output in Template <%=

Azure DevOps

Visual Studio Team Services > Build and release > Pester Test Runner Build Task



Pester Test Runner Build Task

Richard Fennell (Black Marble) | ≛ 612 installs | ★ (5)

Run Pester tests without the need to install Pester in with the PMModule folder or in the source repo (Using Pester 3.4.3 or 4.0.8).

Install

Download

Click Install for Visual Studio Team Services and Download for Team Foundation Server.

Marketplace

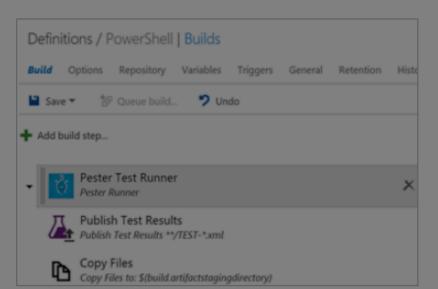
Overview

Q & A

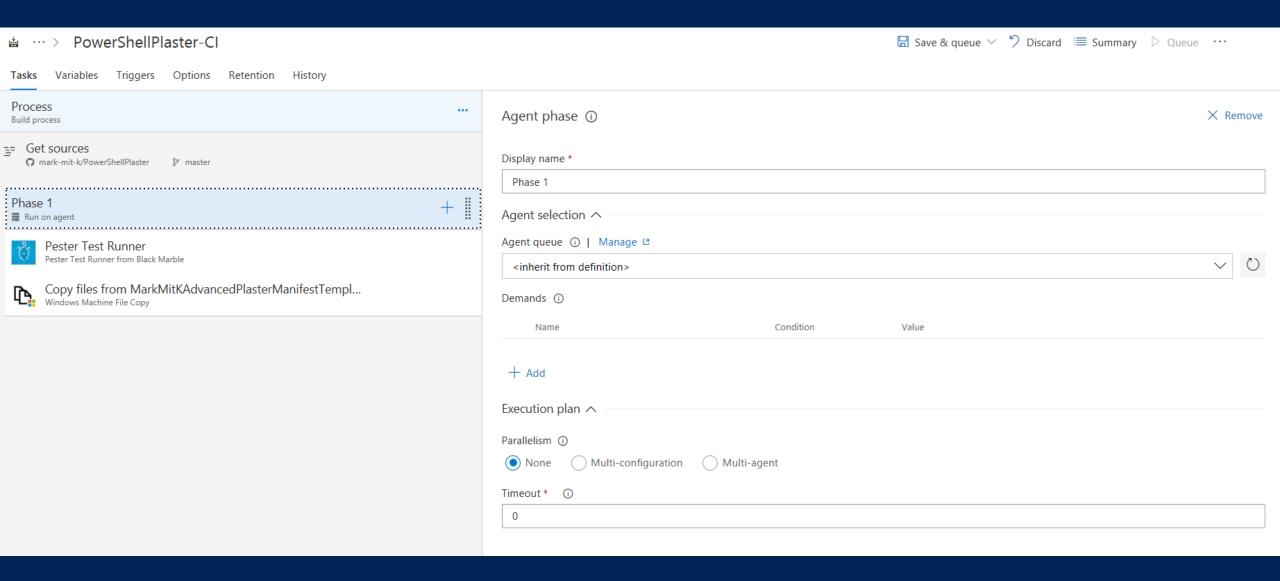
Rating & Review

Releases

- 1.0.x initial public release (targets Pester 3.3.5)
- 4.0.x changed versioning to fit with new release pipeline (still targets Pester 3.4.0)
- 4.1.x now targets 3.4.3, but allows the path to the module to be overriden
- · 4.2.x.- removed non-required demand
- 4.3.x.- Added support for Tag and ExcludeTag parameters
- 4.4.x Added support for multiple comma separated tags
- 5.0.x Added optional support for Pester 4.0.3
- 5.1.x Fixes issue with running tests using the 32Bit process switch (#150)
- 6.0.x Add support for JaCoCo code coverage as provided by Pester 4.0.8 (issue #152)



Azure DevOps Build Tasks



SUMMARY

SUMMARY

Continuous Delivery VSTS Testing Framework Pester Have Fun!
Create some
Modules
Today

Module Framework Plaster







Resources

- https://github.com/PowerShell/Plaster/
- https://github.com/pester/Pester
- https://github.com/
- https://github.com/dahlbyk/posh-git
- https://www.visualstudio.com/team-services/
- https://code.visualstudio.com/
- https://git-scm.com/
- https://docs.microsoft.com/en-us/powershell/gallery/readme
- http://overpoweredshell.com/Working-with-Plaster/
- https://www.youtube.com/watch?v=16CYGTKH73U&feature=yout u.be

Scripts

 Code can be found: aka.ms/mark/pscd/code

