

EBRU CUCEN

POWERSHELL MODULE LIFECYCLE



ABOUT ME

- ▶ EBRU CUCEN
- ▶ @ebrucucen
- ▶ <https://github.com/ebrucucen>
- ▶ Co-ordinator for Get-PSUGUK
- ▶ Associate Trainer for DevOps Tools
- ▶ .NET dev\trainer turned into software lifecycle inhabitant



AGENDA

- ▶ Script Modules
- ▶ Lifecycle of Modules



SCRIPT MODULE

- ▶ Control
 - ▶ public/private functions/variables
- ▶ Group
 - ▶ reuse/abstraction
- ▶ Describe
 - ▶ versions, dependencies

SCRIPT MODULE – [CONTROL]

```
#Get public and private function definition files.
```

```
$Public = @( Get-ChildItem -Path $PSScriptRoot\Public\*.ps1 -ErrorAction SilentlyContinue )
```

```
$Private = @( Get-ChildItem -Path $PSScriptRoot\Private\*.ps1 -ErrorAction SilentlyContinue )
```

```
#Dot source the files
```

```
Foreach($import in @($Public + $Private))  
{  
    Try  
    {  
        . $import.fullname  
    }  
    Catch  
    {  
        Write-Error -Message "Failed to import function $($import.fullname): $_"  
    }  
}
```

```
Export-ModuleMember -Function $Public.Basename
```



MODULE MANAGEMENT

- New-Module
- New-ModuleManifest
- Import-Module
- Get-Module
- Export-ModuleMember
- Remove-Module
- Test-ModuleManifest

SCRIPT MODULE [GROUP]

- ▶ appveyor.yml
- ▶ deploy.PSDeploy.ps1
- ▶ main.build.ps1
- ▶ README.md
- ▶ PSEventLogEntry\
 - ▶ en-US\
 - ▶ Public\
 - ▶ Private\
 - ▶ Test\
 - ▶ PSEventLogEntry.psd1
 - ▶ PSEventLogEntry.psm1

SCRIPT MODULE [DESCRIBE]

New-ModuleManifest -path .\sample.psd1

<https://github.com/ebrucucen/PSModules/blob/master/sample.psd1>

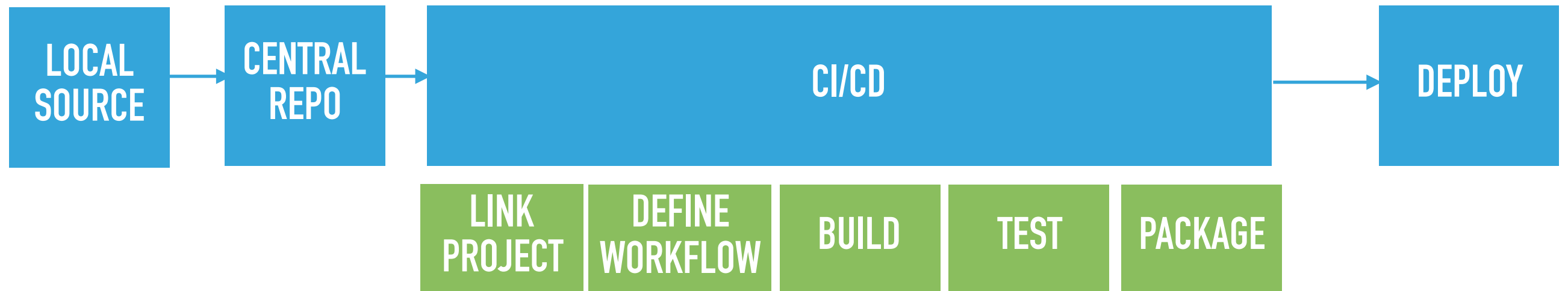
[https://github.com/ebrucucen/PSModules/blob/master/
PSEventLogEntry/PSEventLogEntry.psd1](https://github.com/ebrucucen/PSModules/blob/master/PSEventLogEntry/PSEventLogEntry.psd1)

WHAT IS THE BIG PICTURE?

[illegible]

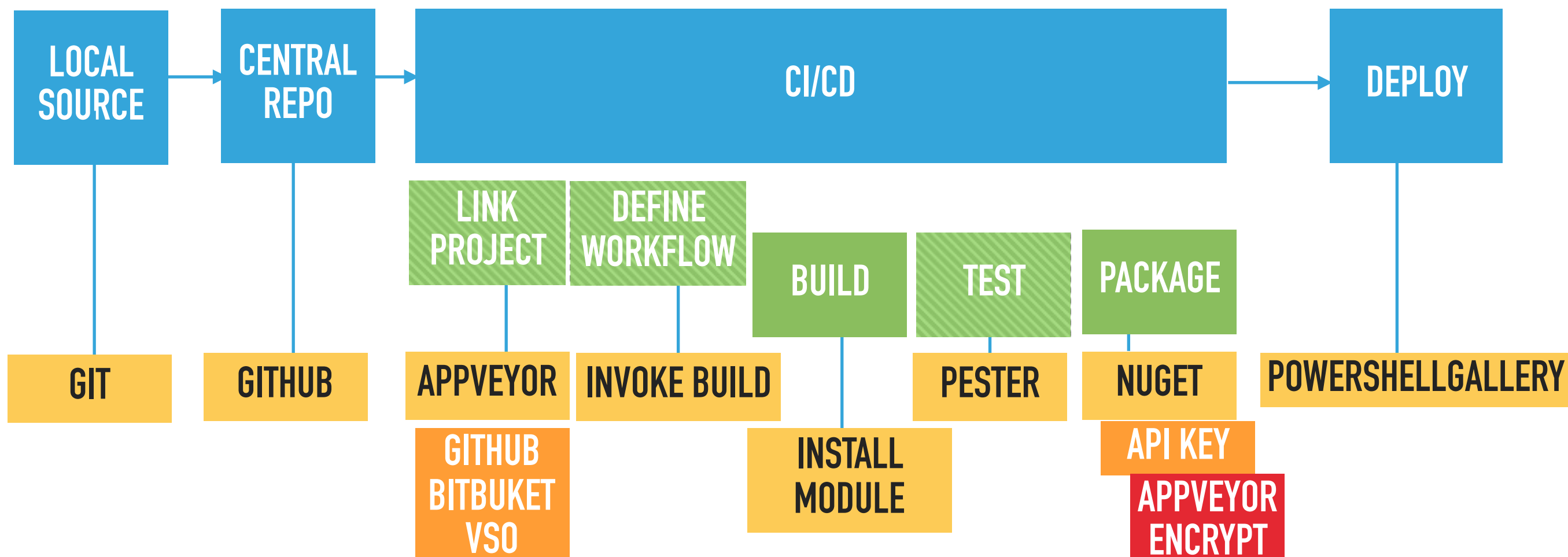


POWERSHELL LIFE CYCLE





SELECTED TOOLSET FOR POWERSHELL




This slides will cover



LINK PROJECT


- ▶ Step1: Sign in:
 - ▶ <https://ci.appveyor.com/login>
- ▶ Step2: Login with with developer/AppVeyor account


The screenshot shows the AppVeyor login interface. At the top is a dark header with the AppVeyor logo and a 'SIGN IN' link. Below the header, the word 'Login' is displayed. There are two main sections: 'Login with your developer account' on the left and 'Login with AppVeyor account' on the right. The left section contains three blue buttons for GitHub, BitBucket, and Visual Studio Online. The right section contains input fields for 'Email' and 'Password', a 'Remember me' checkbox, a 'Login' button, and a 'Forgot password?' link.


 AppVeyor SIGN IN

Login

Login with your developer account

 GitHub

 BitBucket

 Visual Studio Online

Login with AppVeyor account

Email

Password

☐ Remember me


Login

[Forgot password?](#)




LINK PROJECT

► Step3: Select the project to link

 AppVeyor

PROJECTS ENVIRONMENTS DOCS SUPPORT

 EBRU CUCEN

Select repository

GitHub

BitBucket

Visual Studio Online

GitLab

Kiln


GitHub Enterprise

Stash


Git

Mercurial

Subversion

 ebrucucen

PSModules

 ADD



LINK PROJECT

► Step4: Add appveyor.yml

appveyor.yml x

```
1
2  # This is appveyor yml configuration to tell
3  # what to do for each logical build/deployment phase:
4  # all the options are here: https://www.appveyor.com/docs/appveyor-yml/
5
6  # we want disable the MSBuild
7  build: false
8
9  #Kick off the CI/CD pipeline
10 before_build:
11   - echo "started"
12   - ps: install-module invokebuild
13   - echo "installed"
14   - ps: import-module invokebuild
15   - echo "imported"
16
17 #define the build script instead of automated MSBuild
18 build_script:
19   - ps: invoke-build
20
```

<https://github.com/ebrucucen/PSModules/blob/master/appveyor.yml>



APPEYOR REFERENCE*

GENERAL	ENVIRONMENT	BUILD	TEST	PACKAGE	DEPLOY	NOTIFICATION	GLOBAL HANDLERS
skip_commits:	init:	before_build:	before_test:	artifacts:	before_deploy:	provider:	on_success:
- message: - author:	image:	build:	test:	- path:	deploy:	on_build_success:	on_failure:
	hosts:	build_script:	test_script:		deploy_script:	on_build_failure:	on_finish:
	install:	after_build:	after_test:		after_deploy:	on_build_status_changed:	
	- git config	no need MSBuild	we'll run Pester		alternative to PSDeploy	providers: Email, HipChat, Slack, Campfire, Webhook	
		- build: off	- test: off		- provider: - artifact:		
		we'll run Invoke-Build script			providers: Local, FTP, WebDeploy, AzureCS, AzureBlob, S3, NuGet, Environment		
		- build: - ps:			- deploy: off/on		

*These are options relevant for a PowerShell lifecycle



DEFINE WORKFLOW

- ▶ Invoke-Build calls

- ▶ *.build.ps1

[from the same level]

- ▶ a template on the right hand side:

- ▶ the last line calls the tasks in order:

task . Init, Build, Test, Clean, Deploy

```
# Define properties such as:  
# Test/Output Directories ...
```

```
task Init {  
    # Initial tasks such as:  
    # Install/Import packages ...  
}
```

```
task Build -Depend Init{  
    #Build tasks such as:  
    #Generate the resources you want ...  
}
```

```
task Test{  
    #Test tasks such as:  
    #Run Invoke-Pester ...  
}
```

```
task Deploy{  
    #Deploy tasks such as:  
    #Run PSDeploy.ps1  
}
```

```
task Clean {  
    #Clean tasks such as:  
    #Remove test files  
}
```

```
#Define the order you want to call:  
task . Init, Build, Test, Clean, Deploy
```

TEST EXAMPLE

► main.Build.ps1 Test task :

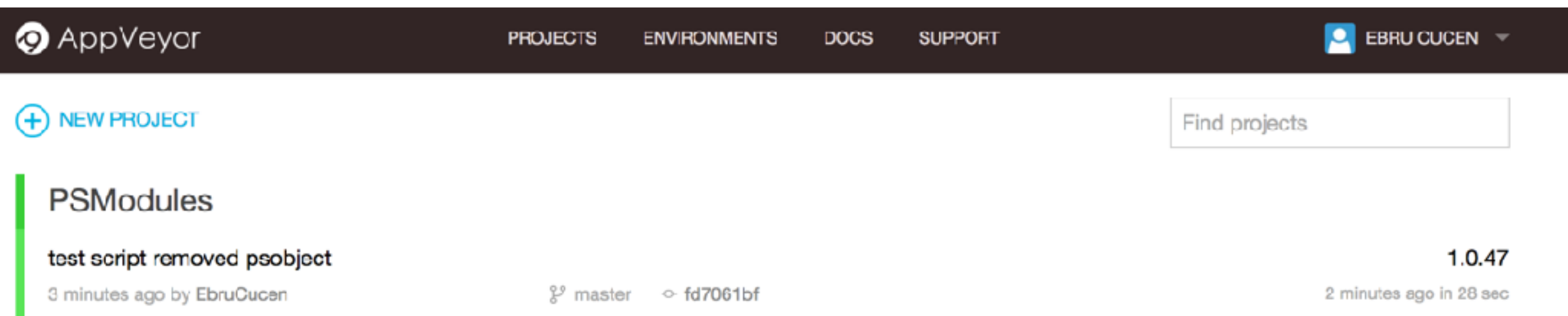
```
#Test Task
task Test{
    $lines
    'TDD: Tests first! '
    #Get the test files:
    $TestFiles= Get-ChildItem -Path $TestLocation -Filter "*.Tests.*"
    foreach ($testFile in $testFiles){
        $testOutputFileName= Join-path -path $testFile.Directory -ChildPath "$($testFile.basename)_$timestamp.xml"
        $testResult=Invoke-Pester -Script $testFile.Fullname -OutputFile $testOutputFileName -OutputFormat NUnitXml
    }
    #upload to Appveyor
    (New-Object 'System.Net.WebClient').UploadFile("https://ci.appveyor.com/api/testresults/nunit/$($env:APPVEYOR_JOB_ID)", `
        (Resolve-Path $testOutputFileName))
    if($testResult.FailedCount -gt 0)
    {
        Write-Error "Failed '$($testResult.FailedCount)' tests, build failed"
    }
    $lines
}
```

ENJOY AUTOMATED CI OUTPUT

- ▶ README.md on Github site can display the status:



- ▶ Appveyor website will have all build history, logs ...

A screenshot of the AppVeyor website interface. The top navigation bar is dark brown with the AppVeyor logo on the left and links for PROJECTS, ENVIRONMENTS, DOCS, and SUPPORT in the center. On the right of the navigation bar is a user profile icon and the name 'EBRU CUCEN'. Below the navigation bar, on the left, is a blue button with a plus icon and the text 'NEW PROJECT'. On the right is a search bar with the placeholder text 'Find projects'. The main content area shows a project named 'PSModules' with a green vertical bar on the left. Below the project name, it says 'test script removed psubject' and '3 minutes ago by EbruCucen'. To the right of this, it shows 'master' and 'fd7061bf'. On the far right, it displays the version '1.0.47' and the build time '2 minutes ago in 28 sec'.

BUILD OUTPUT [1: APPVEYOR.YML CALLED]

PSModules

LATEST BUILD

HISTORY

DEPLOYMENTS

SETTINGS

 NEW BUILD

 RE-BUILD COMMIT

 DEPLOY

 LOG

extra tidy up on init task

1.0.49

a few seconds ago by EbruCucen

 master  5821101b

in a few seconds in 28 sec

CONSOLE

MESSAGES

TESTS 3

ARTIFACTS

```
1 Build started
2 git clone -q --branch=master https://github.com/ebrucucen/PSModules.git C:\projects\psmodules
3 git checkout -qf 5821101b0fef6c430c265166072e4afd8c881f6e
4 echo "started"
5 "started"
6 install-module invokebuild
7 echo "installed"
8 "installed"
9 import-module invokebuild
10 echo "imported"
11 "imported"
12 invoke-build
13 Build . C:\projects\psmodules\main.build.ps1
```

BUILD OUTPUT 2: [INIT TASK IN MAIN.BUILD.PS1 CALLED]

```
14 Task ./Init
15 -----
16 Build System Details:
17
18
19 Name : BHBranchName
20 Value : master
21
22 Name : BHProjectName
23 Value : PSEventLogEntry
24
25 Name : BHPSModuleManifest
26 Value : C:\projects\psmodules\PSEventLogEntry\PSEventLogEntry.psd1
27
28 Name : BHBuildNumber
29 Value : 49
30
31 Name : BHPSModulePath
32 Value : C:\projects\psmodules\PSEventLogEntry
33
34 Name : BHBuildSystem
35 Value : AppVeyor
36
37 Name : BHProjectPath
38 Value : C:\projects\psmodules
39
40 Name : BHCommitMessage
41 Value : extra tidy up on init task
42
43
44
45
46
47 -----
48 Done ./Init 00:00:09.6815866
```

BUILD OUTPUT 3: [PRETEST, TEST]

49 Task ./PreTest

50 -----

51

52

53 Directory: C:\projects\psmodules\PSEventlogEntry

54

55

Mode	LastWriteTime	Length	Name
------	---------------	--------	------

57 -----

-a----	3/24/2017 8:25 PM	595	PSEventlogEntry.psm1
--------	-------------------	-----	----------------------

59 Success import

60 -----

61 Done ./PreTest 00:00:00.0781290

62 Task ./Test

63 -----

64 TDD: Tests first!

65 Executing all tests in C:\projects\psmodules\PSEventLogEntry\Test\PSEventLogEntry.Tests.ps1

66

67 Executing script C:\projects\psmodules\PSEventLogEntry\Test\PSEventLogEntry.Tests.ps1

68

69 Describing Get-EventLogEntry Function Basic Tests

70

71 Context Invalid Parameter Set

72 [+] Should Throw Error for invalid ServerName 3.35s

73 [+] Should Throw Error for invalid LogName 145ms

74

75 Context Valid Parameter Set

76 [+] Should Not Throw An Error for the Valid Logname, Start/End Time inputs 189ms

77 Tests completed in 3.69s

78 Tests Passed: 3, Failed: 0, Skipped: 0, Pending: 0, Inconclusive: 0

79 -----

80 Done ./Test 00:00:04.7536498

81 Done /. 00:00:14.5446171

82 Build succeeded. 4 tasks, 0 errors, 0 warnings 00:00:14.6540113

83

84

85 Discovering tests...OK

86 Build success

REFERENCES

Invoke-build:

<https://github.com/nightroman/Invoke-Build>

My examples:

<https://github.com/ebrucucen/PSModules>

<https://ci.appveyor.com/project/ebrucucen/psmodules>

More github badges?

<https://shields.io>

Appveyor:

<https://www.appveyor.com/docs/appveyor-yml/>

<https://ci.appveyor.com/tools/encrypt>

<https://github.com/PowerShell/DscResource.Tests/blob/dev/AppVeyor.psm1>

Devops:

<https://xebialabs.com/periodic-table-of-devops-tools>