

# Taking Control of PowerShell Profile Scripts

---

POWERSHELL SATURDAY CHATTANOOGA 2018



**Tim Warner**

AUTHOR EVANGELIST, PLURALSIGHT

@TechTrainerTim

timw.info



**Microsoft  
CERTIFIED**

Trainer

Solutions Expert

Cloud Platform and  
Infrastructure

## Overview



Understand PowerShell profiles

Cool profile content ideas

Profile tips and tricks

Session Materials



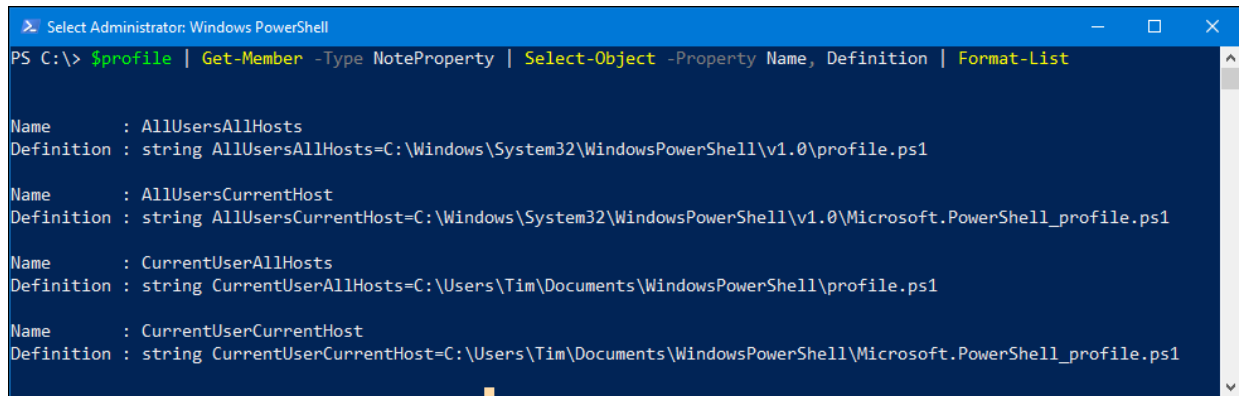
[timw.info/pro](https://timw.info/pro)

# Understand PowerShell Profiles

## PowerShell Profile

A script that runs when PowerShell starts. You can use the profile as a logon script to customize the session environment.

# The PowerShell Desktop Profiles



```
Select Administrator: Windows PowerShell
PS C:\> $profile | Get-Member -Type NoteProperty | Select-Object -Property Name, Definition | Format-List

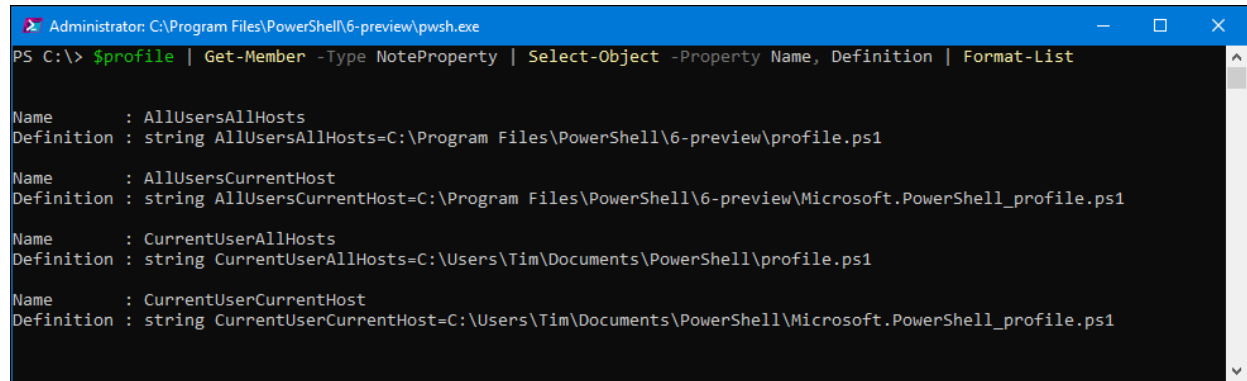
Name      : AllUsersAllHosts
Definition : string AllUsersAllHosts=C:\Windows\System32\WindowsPowerShell\v1.0\profile.ps1

Name      : AllUsersCurrentHost
Definition : string AllUsersCurrentHost=C:\Windows\System32\WindowsPowerShell\v1.0\Microsoft.PowerShell_profile.ps1

Name      : CurrentUserAllHosts
Definition : string CurrentUserAllHosts=C:\Users\Tim\Documents\WindowsPowerShell\profile.ps1

Name      : CurrentUserCurrentHost
Definition : string CurrentUserCurrentHost=C:\Users\Tim\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1
```

# The PowerShell Core Profiles



```
Administrator: C:\Program Files\PowerShell\6-preview\pwsh.exe
PS C:\> $profile | Get-Member -Type NoteProperty | Select-Object -Property Name, Definition | Format-List

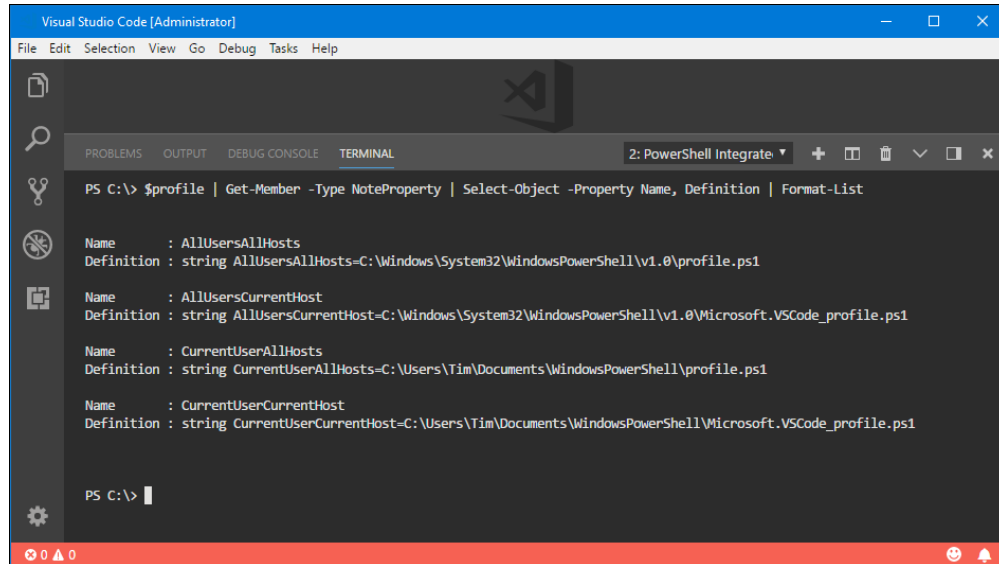
Name      : AllUsersAllHosts
Definition : string AllUsersAllHosts=C:\Program Files\PowerShell\6-preview\profile.ps1

Name      : AllUsersCurrentHost
Definition : string AllUsersCurrentHost=C:\Program Files\PowerShell\6-preview\Microsoft.PowerShell_profile.ps1

Name      : CurrentUserAllHosts
Definition : string CurrentUserAllHosts=C:\Users\Tim\Documents\PowerShell\profile.ps1

Name      : CurrentUserCurrentHost
Definition : string CurrentUserCurrentHost=C:\Users\Tim\Documents\PowerShell\Microsoft.PowerShell_profile.ps1
```

# PowerShell Profiles are Host-Specific



The screenshot shows the Visual Studio Code [Administrator] window with the terminal pane active. The terminal displays the output of the command `$profile | Get-Member -Type NoteProperty | Select-Object -Property Name, Definition | Format-List`. The output lists four PowerShell profiles with their names and definitions.

```
PS C:\> $profile | Get-Member -Type NoteProperty | Select-Object -Property Name, Definition | Format-List

Name       : AllUsersAllHosts
Definition : string AllUsersAllHosts=C:\Windows\System32\WindowsPowerShell\v1.0\profile.ps1

Name       : AllUsersCurrentHost
Definition : string AllUsersCurrentHost=C:\Windows\System32\WindowsPowerShell\v1.0\Microsoft.VSCode_profile.ps1

Name       : CurrentUserAllHosts
Definition : string CurrentUserAllHosts=C:\Users\Tim\Documents\WindowsPowerShell\profile.ps1

Name       : CurrentUserCurrentHost
Definition : string CurrentUserCurrentHost=C:\Users\Tim\Documents\WindowsPowerShell\Microsoft.VSCode_profile.ps1

PS C:\>
```



## Create a Profile

```
if (!(Test-Path -Path $profile.CurrentUserAllHosts))  
    {New-Item -ItemType File -Path $profile -Force}
```

**Check your script execution policy!**

## Start a PowerShell Session

```
powershell.exe "Start-Process powershell -ArgumentList '-  
ExecutionPolicy Bypass -NoProfile' -Verb RunAs"
```

Current User, All Hosts

This recipe is PowerShell launching PowerShell 😊

Check your script execution policy!

Demo



1

asdf

## Cool Profile Content Ideas

## Customize the Console

```
$a = (Get-Host).UI.RawUI  
$a.BackgroundColor = 'DarkBlue'  
$a.ForegroundColor = 'Yellow'  
Clear-Host  
$a.WindowTitle = "PowerShell Desktop " + (Get-Host).Version
```

## Customize the Console

```
$buffer = $console.BufferSize  
$buffer.Width = 80  
$buffer.Height = 5000  
$console.BufferSize = $buffer
```

```
$size = $console.WindowSize  
$size.Width = 80  
$size.Height = 25  
$console.WindowSize = $size
```

```
New-Item alias:np -value "C:\Windows\System32\notepad.exe"  
New-Item alias:st -value "C:\Program Files\Sublime Text 3\sublime_text.exe"
```

## Default PowerShell Prompt

```
(Get-Command -Name Prompt).ScriptBlock  
"PS  
$($ExecutionContext.SessionState.Path.CurrentLocation)$('>' * ($nestedPromptLevel + 1)) ";
```

## Customize the Prompt

```
function prompt {  
    $identity = [Security.Principal.WindowsIdentity]::GetCurrent()  
    $principal = [Security.Principal.WindowsPrincipal] $identity  
  
    $(if (test-path variable:/PSDebugContext) { '[DBG]: ' }  
        elseif  
        ($principal.IsInRole([Security.Principal.WindowsBuiltInRole]"Administrator")) {  
            "[ADMIN]: " }  
            else { ' ' }  
    ) + 'PS ' + $(Get-Location) +  
    $(if ($nestedpromptlevel -ge 1) { '>>' }) + '> '  
}
```



## Get System Uptime

```
function Get-Uptime {  
    $os = Get-WmiObject win32_operatingsystem  
    $uptime = (Get-Date) -  
    ($os.ConvertToDateTime($os.lastbootuptime))  
    $Display = "Uptime: " + $Uptime.Days + " days, " +  
    $Uptime.Hours + " hours, " + $Uptime.Minutes + " minutes"  
    Write-Output $Display  
}  
Clear-Host  
Get-Uptime
```

## Auto Login to Azure

```
Connect-AzureRmAccount -Subscription 'Microsoft Azure  
Sponsorship' -Credential (BetterCredentials\Get-Credential -  
username 'tim@timw.info')
```

# PowerShell Profile Tips and Tricks

```
$s = New-PSSession `
    -ComputerName srv1 `
    -Credential (Get-Credential)
```

```
Invoke-Command -Session $s -
    FilePath $profile
```

```
Invoke-Command -Session $s -
    ScriptBlock {
        $env:USERPROFILE\Documents\Window
        sPowerShell\Microsoft.PowerShell_p
        rofile.ps1"}

```

◀ By default, your profile won't load in remote sessions

◀ Loads the "Current user, current host" profile from the local computer in the remote session

◀ Loads the "Current user, current host" profile from the remote computer in the session

## Pulling a Centralized Profile

```
$ScriptFromGitHub = Invoke-WebRequest -Uri  
https://raw.githubusercontent.com/tomarbutnot/Run-PowerShell-  
Directly-From-GitHub/master/Run-FromGitHub-SamplePowerShell.ps1  
Add-Content -Path $profile -Value $($ScriptFromGitHub.Content)  
  
# For authenticated Gists  
Install-Module -Name PSGist -Repository PSGallery -Verbose  
Get-Command -Module PSGist
```

Thank You!



Tim Warner

Materials: [timw.info/pro](http://timw.info/pro)

Twitter: [@TechTrainerTim](https://twitter.com/TechTrainerTim)

Web: [pluralsight.com](http://pluralsight.com)

Email: [timothy-warner@pluralsight.com](mailto:timothy-warner@pluralsight.com)