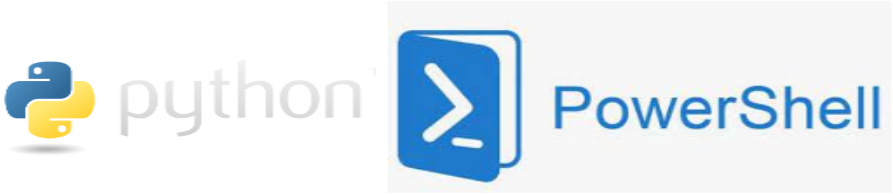

Full credit is given to the above companies including the OS that this PDF file was generated!

PowerShell Get-Help on command 'Get-PSDrive'

PS C:\Users\wahid> Get-Help Get-PSDrive

NAME

Get-PSDrive

SYNOPSIS

Gets drives in the current session.

SYNTAX

Get-PSDrive [-LiteralName] <System.String[]> [-PSProvider <System.String[]>]

[-Scope <System.String>] [-UseTransaction] [<CommonParameters>]

Get-PSDrive [[-Name] <System.String[]>] [-PSProvider <System.String[]>]

[-Scope <System.String>] [-UseTransaction] [<CommonParameters>]

DESCRIPTION

The `Get-PSDrive` cmdlet gets the drives in the current session. You can get a

particular drive or all drives in the session.

This cmdlet gets the following types of drives:

- Windows logical drives on the computer, including drives mapped to network shares.

- Drives exposed by PowerShell providers (such as the Certificate:, Function:, and Alias:

drives) and the HKLM: and HKCU: drives that are exposed by the Windows PowerShell Registry provider. - Session-specified temporary drives and persistent mapped network drives that you create by using the New-PSDrive cmdlet.

Beginning in Windows PowerShell 3.0, the Persist parameter of the `New-PSDrive` cmdlet can create mapped network drives that are saved on the local computer and are available in other sessions. For more information, see New-PSDrive.

Also, beginning in Windows PowerShell 3.0, when an external drive is connected to the computer, Windows PowerShell automatically adds a PSDrive to the file system that represents the new drive. You do not need to restart Windows PowerShell. Similarly, when an external drive is disconnected from the computer, Windows PowerShell automatically deletes the PSDrive that represents the removed drive.

PARAMETERS

-LiteralName <System.String[]>

Specifies the name of the drive.

The value of LiteralName is used exactly as it is typed. No characters are interpreted as wildcards. If the name includes escape characters, enclose it in single quotation marks. Single quotation marks tell Windows PowerShell not to interpret any characters as escape sequences.

-Name <System.String[]>

Specifies, as a string array, the name or name of drives that this cmdlet gets in the operation. Type the drive name or letter without a colon (`:`).

-PSProvider <System.String[]>

Specifies, as a string array, the Windows PowerShell provider. This cmdlet gets only the drives supported by this provider. Type the name of a provider, such as FileSystem, Registry, or Certificate.

-Scope <System.String>

Specifies the scope in which this cmdlet gets the drives.

The acceptable values for this parameter are:

- Global

- Local

- Script

- a number relative to the current scope (0 through the number of scopes, where 0 is the current

scope and 1 is its parent). "Local" is the default.

For more information, see about_Scopes (../Microsoft.PowerShell.Core/About/about_Scopes.md).

-UseTransaction <System.Management.Automation.SwitchParameter>

Includes the command in the active transaction. This parameter is valid only when a transaction is in progress. For more information, see about_Transactions (../Microsoft.PowerShell.Core/About/about_Transactions.md).

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

--------- Example 1: Get drives in the current session ---------

```
PS C:\> Get-PSDrive

Name           Used (GB)     Free (GB) Provider      Root
----           ---------     --------- --------      ----
Alias                                  Alias
C                 202.06      23718.91 FileSystem    C:\
Cert                                   Certificate   \
D                1211.06     123642.32 FileSystem    D:\
Env                                    Environment
Function                               Function
HKCU                                   Registry      HKEY_CURRENT_USER
HKLM                                   Registry      HKEY_LOCAL_MACHINE
Variable                               Variable
```

This command gets the drives in the current session.

The output shows the hard drive (C:), CD-ROM drive (D:), and the drives exposed by the Windows PowerShell providers (Alias:, Cert:, Env:, Function:, HKCU:, HKLM:, and Variable:).

---------- Example 2: Get a drive on the computer ----------

```
PS C:\foo> Get-PSDrive D

Name           Used (GB)     Free (GB) Provider      Root
```

```
----           ---------     --------- --------       ----
D                 1211.06     123642.32 FileSystem     D:\
```

This command gets the D: drive on the computer. Note that the drive letter in the command is not followed by a colon.

Example 3: Get all the drives that are supported by the Windows PowerShell file system provider

```
PS C:\> Get-PSDrive -PSProvider FileSystem
Name           Used (GB)     Free (GB) Provider      Root
----           ---------     --------- --------      ----
A                                      A:\
C                 202.06      23718.91 FileSystem    C:\
D                1211.06     123642.32 FileSystem    D:\
G                 202.06        710.91 FileSystem    \\Music\GratefulDead
```

This command gets all of the drives that are supported by the Windows PowerShell FileSystem provider. This includes fixed drives, logical partitions, mapped network drives, and temporary drives that you create by using the New-PSDrive cmdlet.

Example 4: Check to see if a drive is in use as a Windows PowerShell drive name

```
if (Get-PSDrive X -ErrorAction SilentlyContinue) {
    Write-Host 'The X: drive is already in use.'
} else {
    New-PSDrive -Name X -PSProvider Registry -Root HKLM:\SOFTWARE
}
```

This command checks to see whether the X drive is already in use as a Windows PowerShell drive name. If it is not, the command uses the `New-PSDrive` cmdlet to create a temporary drive that is mapped to the HKLM:\SOFTWARE registry key.

----- Example 5: Compare the types of files system drives -----

```
PS C:\> Get-PSDrive -PSProvider FileSystem
Name           Used (GB)     Free (GB) Provider      Root
----           ---------     --------- --------      ----
A                                      A:\
C                202.06      23718.91 FileSystem    C:\
D               1211.06     123642.32 FileSystem    D:\
G                202.06        710.91 FileSystem    \\Music\GratefulDead
X                             Registry      HKLM:\Network

PS C:\> net use
New connections will be remembered.
Status       Local     Remote                    Network
-------------------------------------------------------------------------------
OK           G:        \\Server01\Public         Microsoft Windows Network

PS C:\> [System.IO.DriveInfo]::GetDrives() | Format-Table
Name DriveType DriveFormat IsReady AvailableFreeSpace TotalFreeSpace TotalSize
   RootDirectory VolumeLabel
---- --------- ----------- ------- ------------------ -------------- ---------
   ------------- -----------
A:\    Network              False
   A:\
C:\      Fixed NTFS          True  771920580608       771920580608
988877418496  C:\           Windows
D:\      Fixed NTFS          True  689684144128       689684144128
1990045179904 D:\           Big Drive
E:\      CDRom               False
   E:\
G:\    Network NTFS          True      69120000          69120000
104853504 G:\           GratefulDead

PS N:\> Get-CimInstance -Class Win32_LogicalDisk
```

```
DeviceID DriveType ProviderName   VolumeName        Size         FreeSpace
-------- --------- ------------   ----------        ----         ---------
A:       4
C:       3                      Windows           988877418496  771926069248
D:       3                      Big!              1990045179904  689684144128
E:       5
G:       4         \\Music\GratefulDead            988877418496  771926069248


PS C:\> Get-CimInstance -Class Win32_NetworkConnection
LocalName RemoteName           ConnectionState Status
--------- ----------           --------------- ------
G:        \\Music\GratefulDead  Connected       OK
```

This example compares the types of file system drives that are displayed by `Get-PSDrive` to those displayed by using other methods. This example demonstrates different ways to display drives in Windows PowerShell, and it shows that session-specific drives created by using the New-PSDrive cmdlet are accessible only in Windows PowerShell.

The first command uses `Get-PSDrive` to get all of the file system drives in the session. This includes the fixed drives (C: and D:), a mapped network drive (G:) that was created by using the Persist parameter of `New-PSDrive`, and a PowerShell drive (T:) that was created by using `New-PSDrive` without the Persist parameter.

The net use command displays Windows mapped network drives, in this case it displays only the G drive. It does not display the X: drive that was created by `New-PSDrive`. It shows that the G: drive is also mapped to \\Music\GratefulDead.

The third command uses the GetDrives method of the Microsoft .NET Framework System.IO.DriveInfo class. This command gets the Windows file system drives,

including drive G:, but it does not get the drives created by `New-PSDrive`.

The fourth command uses the `Get-CimInstance` cmdlet to get the instances of the Win32_LogicalDisk class. It returns the A:, C:, D:, E:, and G: drives, but not the drives created by `New-PSDrive`.

The last command uses the `Get-CimInstance` cmdlet to display the instances of the Win32_NetworkConnection class. Like net use , it returns only the persistent G: drive created by `New-PSDrive`.

REMARKS

To see the examples, type: "get-help Get-PSDrive -examples".

For more information, type: "get-help Get-PSDrive -detailed".

For technical information, type: "get-help Get-PSDrive -full".

For online help, type: "get-help Get-PSDrive -online"