**Working with Registry Entries**

* Article
* 10/08/2021
* 5 minutes to read
* 4 contributors

Because registry entries are properties of keys and, as such, cannot be directly browsed, we need to take a slightly different approach when working with them.

**Listing Registry Entries**

There are many different ways to examine registry entries. The simplest way is to get the property names associated with a key. For example, to see the names of the entries in the registry key HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion, use Get-Item. Registry keys have a property with the generic name of "Property" that is a list of registry entries in the key. The following command selects the Property property and expands the items so that they are displayed in a list:

PowerShellCopy

Get-Item -Path Registry::HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion |

Select-Object -ExpandProperty Property

OutputCopy

DevicePath

MediaPathUnexpanded

ProgramFilesDir

CommonFilesDir

ProductId

To view the registry entries in a more readable form, use Get-ItemProperty:

PowerShellCopy

Get-ItemProperty -Path Registry::HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion

OutputCopy

PSPath : Microsoft.PowerShell.Core\Registry::HKEY\_LOCAL\_MACHINE\SO

FTWARE\Microsoft\Windows\CurrentVersion

PSParentPath : Microsoft.PowerShell.Core\Registry::HKEY\_LOCAL\_MACHINE\SO

FTWARE\Microsoft\Windows

PSChildName : CurrentVersion

PSDrive : HKLM

PSProvider : Microsoft.PowerShell.Core\Registry

DevicePath : C:\WINDOWS\inf

MediaPathUnexpanded : C:\WINDOWS\Media

ProgramFilesDir : C:\Program Files

CommonFilesDir : C:\Program Files\Common Files

ProductId : 76487-338-1167776-22465

WallPaperDir : C:\WINDOWS\Web\Wallpaper

MediaPath : C:\WINDOWS\Media

ProgramFilesPath : C:\Program Files

PF\_AccessoriesName : Accessories

(default) :

The Windows PowerShell-related properties for the key are all prefixed with "PS", such as **PSPath**, **PSParentPath**, **PSChildName**, and **PSProvider**.

You can use the \*.\* notation for referring to the current location. You can use Set-Location to change to the **CurrentVersion** registry container first:

PowerShellCopy

Set-Location -Path Registry::HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion

Alternatively, you can use the built-in HKLM PSDrive with Set-Location:

PowerShellCopy

Set-Location -Path hklm:\SOFTWARE\Microsoft\Windows\CurrentVersion

You can then use the \*.\* notation for the current location to list the properties without specifying a full path:

PowerShellCopy

Get-ItemProperty -Path .

OutputCopy

...

DevicePath : C:\WINDOWS\inf

MediaPathUnexpanded : C:\WINDOWS\Media

ProgramFilesDir : C:\Program Files

...

Path expansion works the same as it does within the file system, so from this location you can get the **ItemProperty** listing for HKLM:\SOFTWARE\Microsoft\Windows\Help by using Get-ItemProperty -Path ..\Help.

**Getting a Single Registry Entry**

If you want to retrieve a specific entry in a registry key, you can use one of several possible approaches. This example finds the value of **DevicePath** in HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion.

Using Get-ItemProperty, use the **Path** parameter to specify the name of the key, and the **Name** parameter to specify the name of the **DevicePath** entry.

PowerShellCopy

Get-ItemProperty -Path HKLM:\Software\Microsoft\Windows\CurrentVersion -Name DevicePath

OutputCopy

PSPath : Microsoft.PowerShell.Core\Registry::HKEY\_LOCAL\_MACHINE\Software\

Microsoft\Windows\CurrentVersion

PSParentPath : Microsoft.PowerShell.Core\Registry::HKEY\_LOCAL\_MACHINE\Software\

Microsoft\Windows

PSChildName : CurrentVersion

PSDrive : HKLM

PSProvider : Microsoft.PowerShell.Core\Registry

DevicePath : C:\WINDOWS\inf

This command returns the standard Windows PowerShell properties as well as the **DevicePath** property.

**Note**

Although Get-ItemProperty has **Filter**, **Include**, and **Exclude** parameters, they cannot be used to filter by property name. These parameters refer to registry keys, which are item paths and not registry entries, which are item properties.

Another option is to use the Reg.exe command line tool. For help with reg.exe, type reg.exe /? at a command prompt. To find the DevicePath entry, use reg.exe as shown in the following command:

PowerShellCopy

reg query HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion /v DevicePath

OutputCopy

! REG.EXE VERSION 3.0

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion

DevicePath REG\_EXPAND\_SZ %SystemRoot%\inf

You can also use the **WshShell** COM object to find some registry entries, although this method does not work with large binary data or with registry entry names that include characters such as backslash (\). Append the property name to the item path with a \ separator:

PowerShellCopy

(New-Object -ComObject WScript.Shell).RegRead("HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\DevicePath")

OutputCopy

%SystemRoot%\inf

**Setting a Single Registry Entry**

If you want to change a specific entry in a registry key, you can use one of several possible approaches. This example modifies the **Path** entry under HKEY\_CURRENT\_USER\Environment. The **Path** entry specifies where to find executable files.

1. Retrieve the current value of the **Path** entry using Get-ItemProperty.
2. Add the new value, separating it with a ;.
3. Use Set-ItemProperty with the specified key, entry name, and value to modify the registry entry.

PowerShellCopy

$value = Get-ItemProperty -Path HKCU:\Environment -Name Path

$newpath = $value.Path += ";C:\src\bin\"

Set-ItemProperty -Path HKCU:\Environment -Name Path -Value $newpath

**Note**

Although Set-ItemProperty has **Filter**, **Include**, and **Exclude** parameters, they cannot be used to filter by property name. These parameters refer to registry keys—which are item paths—and not registry entries—which are item properties.

Another option is to use the Reg.exe command line tool. For help with reg.exe, type **reg.exe /?** at a command prompt.

The following example changes the **Path** entry by removing the path added in the example above. Get-ItemProperty is still used to retrieve the current value to avoid having to parse the string returned from reg query. The **SubString** and **LastIndexOf** methods are used to retrieve the last path added to the **Path** entry.

PowerShellCopy

$value = Get-ItemProperty -Path HKCU:\Environment -Name Path

$newpath = $value.Path.SubString(0, $value.Path.LastIndexOf(';'))

reg add HKCU\Environment /v Path /d $newpath /f

OutputCopy

The operation completed successfully.

**Creating New Registry Entries**

To add a new entry named "PowerShellPath" to the **CurrentVersion** key, use New-ItemProperty with the path to the key, the entry name, and the value of the entry. For this example, we will take the value of the Windows PowerShell variable $PSHome, which stores the path to the installation directory for Windows PowerShell.

You can add the new entry to the key by using the following command, and the command also returns information about the new entry:

PowerShellCopy

New-ItemProperty -Path HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion -Name PowerShellPath -PropertyType String -Value $PSHome

OutputCopy

PSPath : Microsoft.PowerShell.Core\Registry::HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion

PSParentPath : Microsoft.PowerShell.Core\Registry::HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows

PSChildName : CurrentVersion

PSDrive : HKLM

PSProvider : Microsoft.PowerShell.Core\Registry

PowerShellPath : C:\Program Files\Windows PowerShell\v1.0

The **PropertyType** must be the name of a **Microsoft.Win32.RegistryValueKind** enumeration member from the following table:

| **PropertyType Value** | **Meaning** |
| --- | --- |
| Binary | Binary data |
| DWord | A number that is a valid UInt32 |
| ExpandString | A string that can contain environment variables that are dynamically expanded |
| MultiString | A multiline string |
| String | Any string value |
| QWord | 8 bytes of binary data |

**Note**

You can add a registry entry to multiple locations by specifying an array of values for the **Path** parameter:

PowerShellCopy

New-ItemProperty -Name PowerShellPath -PropertyType String -Value $PSHome `

-Path HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion, HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion

You can also overwrite a pre-existing registry entry value by adding the **Force** parameter to any New-ItemProperty command.

**Renaming Registry Entries**

To rename the **PowerShellPath** entry to "PSHome," use Rename-ItemProperty:

PowerShellCopy

Rename-ItemProperty -Path HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion -Name PowerShellPath -NewName PSHome

To display the renamed value, add the **PassThru** parameter to the command.

PowerShellCopy

Rename-ItemProperty -Path HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion -Name PowerShellPath -NewName PSHome -passthru

**Deleting Registry Entries**

To delete both the PSHome and PowerShellPath registry entries, use Remove-ItemProperty:

PowerShellCopy

Remove-ItemProperty -Path HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion -Name PSHome

Remove-ItemProperty -Path HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion -Name PowerShellPath