# PowerShell Providers

## Working with Provider cmdlets 2

**In this lecture** we will take a look at Item property and path cmdlets.

**ItemProperty cmdlets**

- **New-ItemProperty** - creates a new property for a specified item and sets its value. Typically, this cmdlet is used to create new registry values.
- **Get-itemProperty** gets the property of an item like viewing registry properties and their values. You can use this cmdlet to get information about directories, files or registry entries.
- **Remove-ItemProperty** – This deletes a property and its value from an item. You can use it to delete registry values and the data that they store.

Please note: That the registry entry must exist before you can edit the entry.

First, we need to create a **new registry key in the HKLM registry hive.** We will use the **New-item cmdlet.**
In the file system, **New-Item** creates files and folders. In the registry, New-Item creates registry keys and entries.

**Be sure to open PowerShell as an Administrator.**

Using this command we will create a registry key named **MyCompany.**

Command #1

**New-Item -Path "HKLM:\Software\MyCompany"**

Press return

**Displayed** is our registry key **MyCompany**

Now we need to use the command New-itemproperty to add a **new registry entry** to the MyCompany registry key.

**Command #2**

**New-Itemproperty -Path "HKLM:\Software\MyCompany" -name NewProperty -value NewPropertyValue**

Press return

We just added the name **NewProperty** to the registry.

Now we need to add a new value of **400 employees** to my company.

**Command #3**

**New-itemproperty -path HKLM:\Software\MyCompany -name NoOfEmployees -value 400**

Press return

And there is our employee values of 400.

Let's take a look at the **results** by taking a look at the windows registry editor.

- From the search bar type registry
- From HKLM, follow the path software\MyCompany
- Click MyCompany

And there is our registry key and our entries.

Now let's take a look at the **results** by using the **get-itemproperty** cmdlet.

**Type CLS – Press return**

**Command #4**

**Get-itemproperty** hklm:\software\mycompany

Press return

And you can see all our entries.

**Now** we will remove the registry value NewProperty that we created using Command #2

**Command#5**

**Remove-Itemproperty -Path "HKLM:\Software\MyCompany" -name NewProperty**

Press return

- Now run **Command #4 again** and notice that the NewProperty value has been removed.

Now we will remove the **MyCompany registry entry** by using the cmdlet **remove-item**

**Command #6**

**Remove-Item -Path "HKLM:\Software\MyCompany"**

Press return

**Rerun** Command #4 and you will see that MyCompany has been removed.

**Path cmdlets**

**Now let's checkout several of the path cmdlets**

- **Test-Path** - Determines whether all elements of the path exist. This cmdlet is handy if you are writing a script and you want to test whether the path to a file is true or false. Test path can save you a lot of time.

For Example: Perhaps you need to check to see if the password log is present in the debug directory.

**Command #7**

**Test-Path 'C:\windows\debug\PASSWD.LOG'**

**Press return**

The cmdlet returns **true,** so we know that passwd.log file is in that folder.

**Now let's check out the Resolve-path cmdlet**

- **Resolve-Path** - Displays the items and containers that match the wildcard pattern at the location specified.

Now let's say you need to take a look at all the **paths** in the **Windows directory.**

**Command #8**

**Resolve-path "C:\windows*"**

Notice the wildcard in the command.

Press return

This command lists all the folders, subfolders and files in the Windows directory.

- Let’s take a look at the **Split-path cmdlet.** Split-path returns a string that describes the location of the items and returns the specified part of a path. Let's say you want to list all the log files in this C:\Windows folder. You can use the split-path cmdlet.

**Command #9**

**Split-Path -Path "C:\windows*.Log" -Leaf -resolve**

Press return

**Now for Parameters we used:**

**-Path is all the log files located in C:\windows**

**-Leaf** This command displays the files that are referenced by the split path. Because this path is split to the last item, also known as the leaf, the command displays only the file names.

**-resolve** The resolve parameter tells Split path to display the items that display pass references. Instead of displaying the split path.

At this point we've completed this lecture. Thanks for watching and we'll see you in the next lecture.