**Module 5: Querying Management Information by Using WMI and CIM**

1. What class could be used to view the current IP address of a network adapter? Does the class have any methods that could be used to release a DHCP lease? (Hint: network is a good keyword here.)

You can use the Win32\_NetworkAdapterConfiguration class.  
If you run Get-Wmiobject for this class and pipe to Get-Member you should see a number of DHCP related methods. You can also find this using a CIM cmdlet:   
**Get-CimClass win32\_networkadapterconfiguration | select -expand methods | where Name -match "dhcp"**

**Or**

**get-wmiobject -class win32\_networkadapterconfiguration | where -property dhcpenabled**

1. Create a table that shows a computer name, operating system build number, operating system description (caption), and BIOS serial number. (Hint: you’ve seen this technique, but you’ll need to reverse it a bit and query the OS class first, then query the BIOS second).  
     
   get-wmiobject win32\_operatingsystem | Select BuildNumber,Caption,  
   @{l='Computername';e={$\_.\_\_SERVER}},  
   @{l='BIOSSerialNumber';e={(gwmi win32\_bios).serialnumber }} | ft –auto  
     
   or using the CIM cmdlets:  
   get-ciminstance win32\_operatingsystem | Select BuildNumber,Caption,  
   @{l='Computername';e={$\_.CSName}},  
   @{l='BIOSSerialNumber';e={(get-ciminstance win32\_bios).serialnumber }} | ft -auto

PS C:\Users\ve40013372> **get-wmiobject win32\_operatingsystem | select PscomputerName,buildnumber,caption, @{l="BIos Serial number"; e={(get-wmiobject win32\_bios).serialnumber}} | ft -autosize**

PSComputerName buildnumber caption BIos Serial number

-------------- ----------- ------- ------------------

L-185007166 17134 Microsoft Windows 10 Enterprise CND54555J3

1. Query a list of hotfixes using WMI. (Hint: Microsoft formally refers to these as quick fix engineering). Is the list different from that returned by the Get-Hotfix cmdlet?

We will be able to get the list of the hotfixes that are installed on the computer using the following cmdlet:

**Get-wmiobject win32\_quickfixengineering**

Is the list different from that returned by the Get-Hotfix cmdlet? 🡪 **Yes, we will able to get the same content by using the Get-Hotfix cmdlet also**

1. Display a list of services, including their current status, their start mode, and the account they use to log on.  
     
   We will be able to get the desired data by using any of the following cmdlet

get-wmiobject win32\_service | Select Name,State,StartMode,StartName  
OR  
get-ciminstance win32\_service | Select Name,State,StartMode,StartName

**PS C:\Users\ve40013372> get-wmiobject win32\_service | select name,state,startname,startmode**

1. Can you find a class that will display a list of installed software products? Do you consider the resulting list to be complete?  
     
   We have 6 classes to get the information related to product in PowerShell. They are:

CIM\_Product

Win32\_ComputerSystemProduct

**Win32\_Product**

SoftwareLicensingProduct

CIM\_FRUIncludesProduct

CIM\_CompatibleProduct

We can use the **get-wmiobject win32\_product | ft** cmdlet in order get the information of the software products that are installed in a system.

1. What method of a WMI Win32\_Process object would terminate a given process?  
     
   You could search the MSDN documentation for the Win32\_Process class. Or you might use the CIM cmdlets since they also work with WMI to list all of the possible methods.

The Terminate Method of the win32\_process is used to terminate a process. It gives us a return value by which we can know the current status of the process.

Successful completion (0)

Access denied (2)

Insufficient privilege (3)

Unknown failure (8)

Path not found (9)

Invalid parameter (21)

Other (22 4294967295)

Get-CimClass win32\_process | select -ExpandProperty methods  
  
In either event, you should see the Terminate() method.

1. Write four different commands that could be used to terminate all processes named “Notepad”, assuming that multiple processes might be running under that same name.  
   The four commands that can be used to terminate all process named “Notepad” are as follows:

get-process Notepad | stop-process  
stop-process -name Notepad   
get-process notepad | foreach {$\_.Kill()}  
Get-WmiObject win32\_process -filter {name='notepad.exe'} | Invoke-WmiMethod -Name Terminate