**Getting WMI Objects (Get-CimInstance)**

Windows Management Instrumentation (WMI) is a core technology for Windows system administration because it exposes a wide range of information in a uniform manner. Because of how much WMI makes possible, the PowerShell cmdlet for accessing WMI objects, Get-CimInstance, is one of the most useful for doing real work. We are going to discuss how to use the CimCmdlets to access WMI objects and then how to use WMI objects to do specific things.

**Listing WMI Classes**

The first problem most WMI users encounter is trying to find out what can be done with WMI. WMI classes describe the resources that can be managed. There are hundreds of WMI classes, some of which contain dozens of properties.

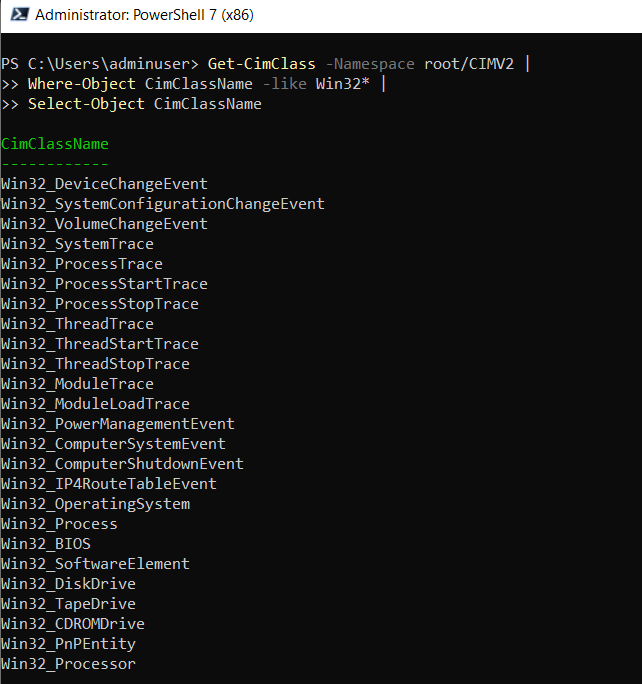
Get-CimClass addresses this problem by making WMI discoverable. You can get a list of the WMI classes available on the local computer by typing:

PowerShellCopy

Get-CimClass -Namespace root/CIMV2 |

Where-Object CimClassName -like Win32\* |

Select-Object CimClassName

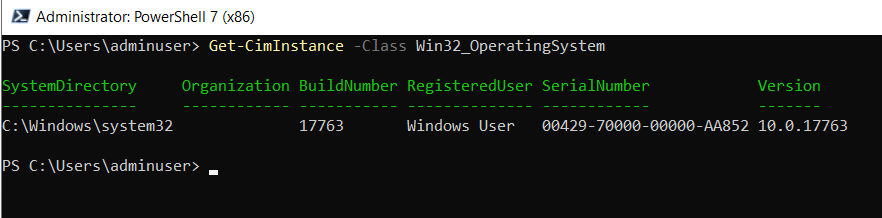
****

**Displaying WMI Class Details**

If you already know the name of a WMI class, you can use it to get information immediately. For example, one of the WMI classes commonly used for retrieving information about a computer is Win32\_OperatingSystem

PowerShellCopy

Get-CimInstance -Class Win32\_OperatingSystem

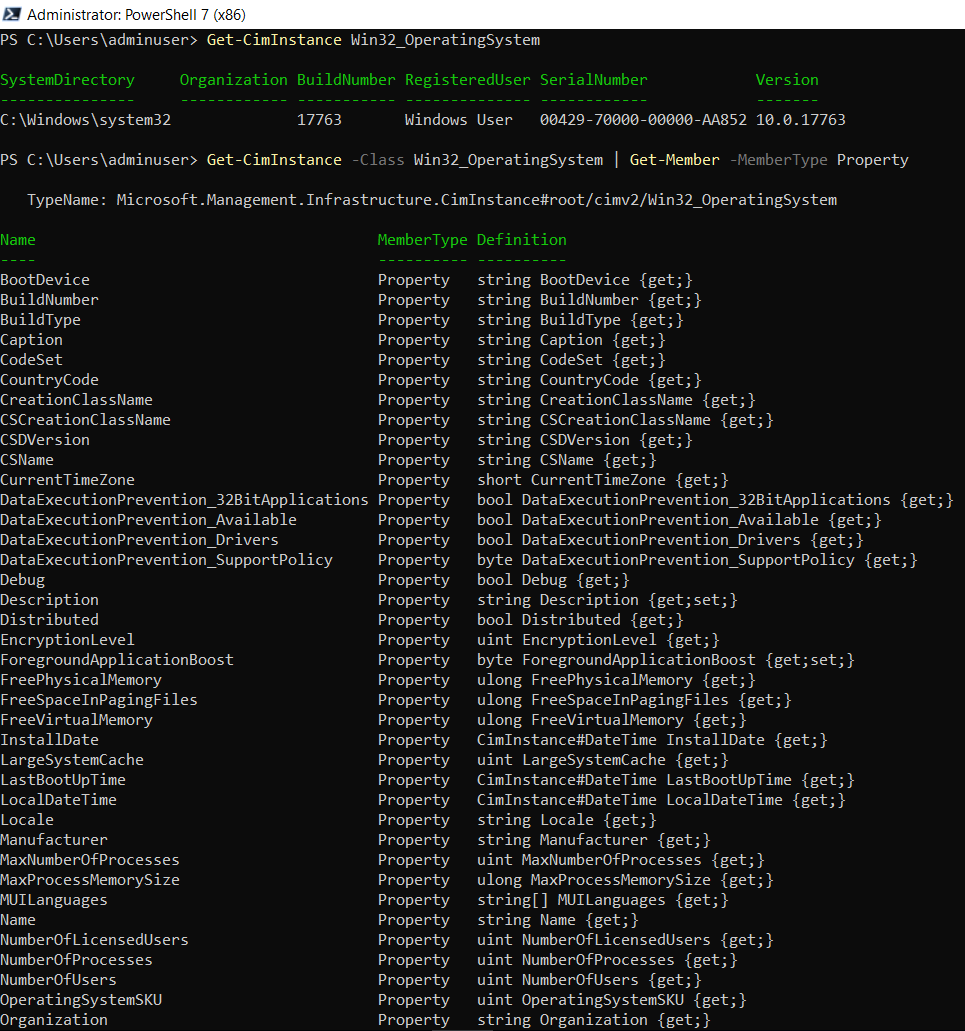
****

Although we are showing all of the parameters, the command can be expressed in a more succinct way. The ComputerName parameter is not necessary when connecting to the local system. We show it to demonstrate the most general case and remind you about the parameter. The Namespace defaults to root/CIMV2, and can be omitted as well. Finally, most cmdlets allow you to omit the name of common parameters. With Get-CimInstance, if no name is specified for the first parameter, PowerShell treats it as the Class parameter. This means the last command could have been issued by typing:

PowerShellCopy

Get-CimInstance Win32\_OperatingSystem

Get-CimInstance -Class Win32\_OperatingSystem | Get-Member -MemberType Property

****

**Displaying Non-Default Properties with Format Cmdlets**

If you want information contained in the Win32\_OperatingSystem class that is not displayed by default, you can display it by using the Format cmdlets. For example, if you want to display available memory data, type:

PowerShellCopy

Get-CimInstance -Class Win32\_OperatingSystem |

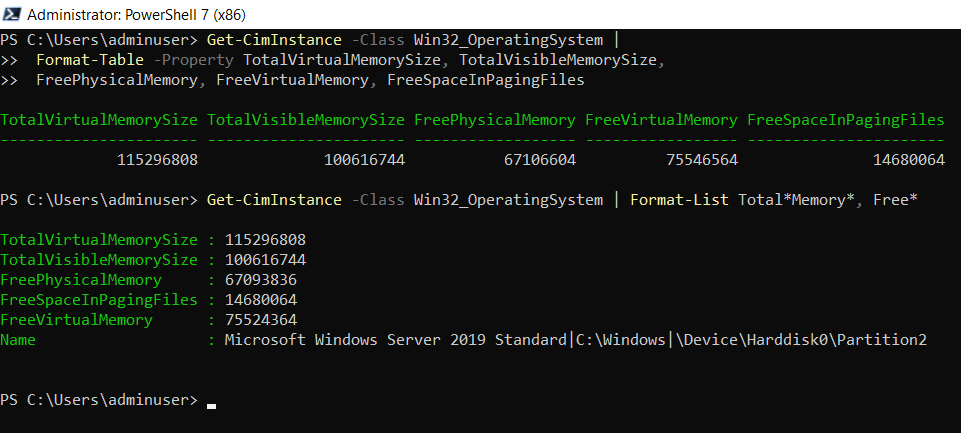
Format-Table -Property TotalVirtualMemorySize, TotalVisibleMemorySize, FreePhysicalMemory, FreeVirtualMemory, FreeSpaceInPagingFiles

Wildcards work with property names in Format-Table, so the final pipeline element can be reduced to Format-Table -Property Total\*Memory\*, Free\*

The memory data might be more readable if you format it as a list by typing:

PowerShellCopy

Get-CimInstance -Class Win32\_OperatingSystem | Format-List Total\*Memory\*, Free\*

****