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

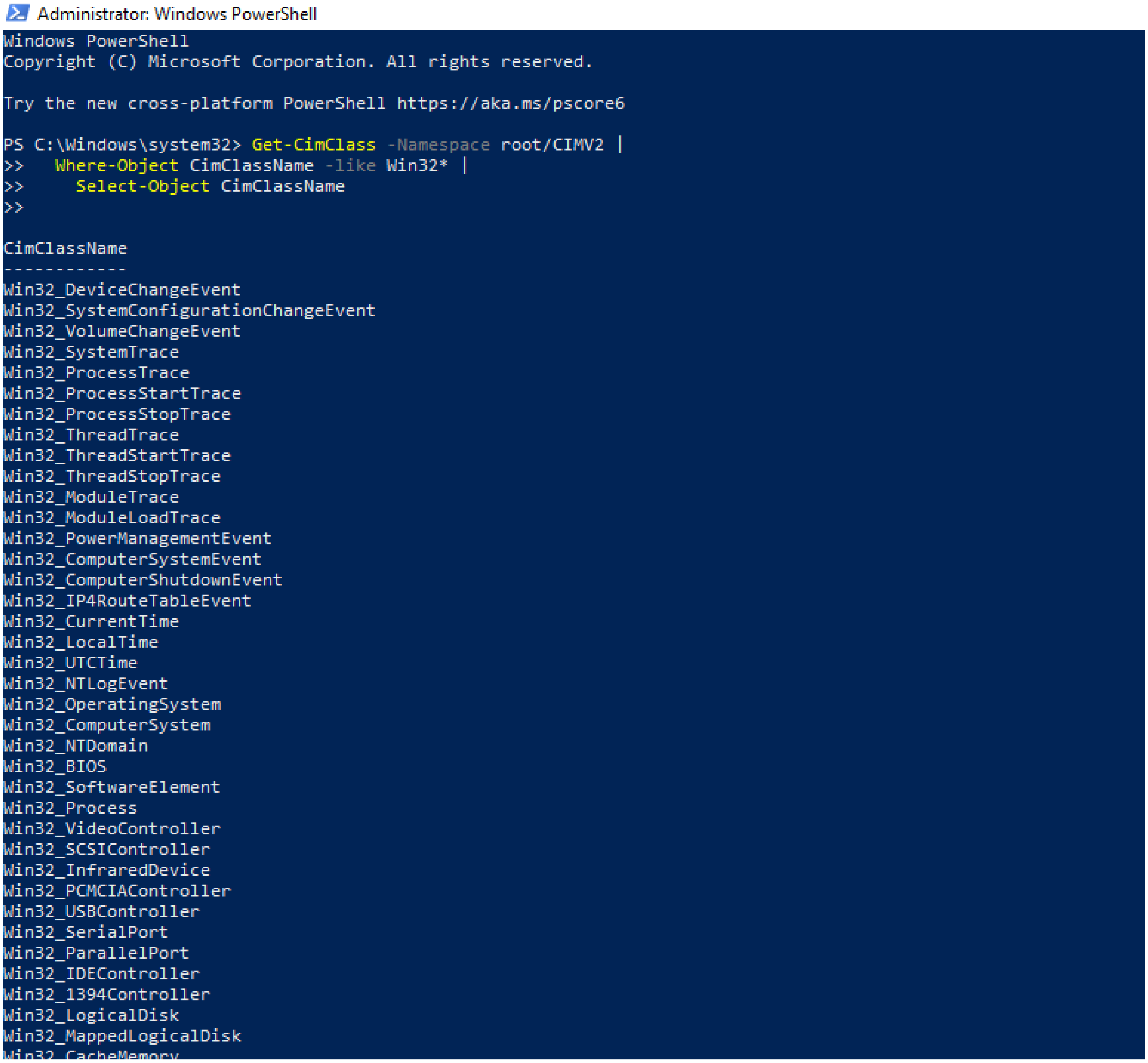
The first problem most WMI users encounter is trying to find out what can be done with WMI. WMI classes describe the resources that can be managed. There are hundreds of WMI classes, some of which contain dozens of properties.

`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

```powershell
Get-CimClass -Namespace root/CIMV2 |
  Where-Object CimClassName -like Win32* |
    Select-Object CimClassName
```

OutputCopy



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> Get-CimClass -Namespace root/CIMV2 |
>>   Where-Object CimClassName -like Win32* |
>>     Select-Object CimClassName
>>

CimClassName
------------
Win32_DeviceChangeEvent
Win32_SystemConfigurationChangeEvent
Win32_VolumeChangeEvent
Win32_SystemTrace
Win32_ProcessTrace
Win32_ProcessStartTrace
Win32_ProcessStopTrace
Win32_ThreadTrace
Win32_ThreadStartTrace
Win32_ThreadStopTrace
Win32_ModuleTrace
Win32_ModuleLoadTrace
Win32_PowerManagementEvent
Win32_ComputerSystemEvent
Win32_ComputerShutdownEvent
Win32_IP4RouteTableEvent
Win32_CurrentTime
Win32_LocalTime
Win32_UTCTime
Win32_NTLogEvent
Win32_OperatingSystem
Win32_ComputerSystem
Win32_NTDomain
Win32_BIOS
Win32_SoftwareElement
Win32_Process
Win32_VideoController
Win32_SCSIController
Win32_InfraredDevice
Win32_PCMCIAController
Win32_USBController
Win32_SerialPort
Win32_ParallelPort
Win32_IDEController
Win32_1394Controller
Win32_LogicalDisk
Win32_MappedLogicalDisk
Win32_CacheMemory
```

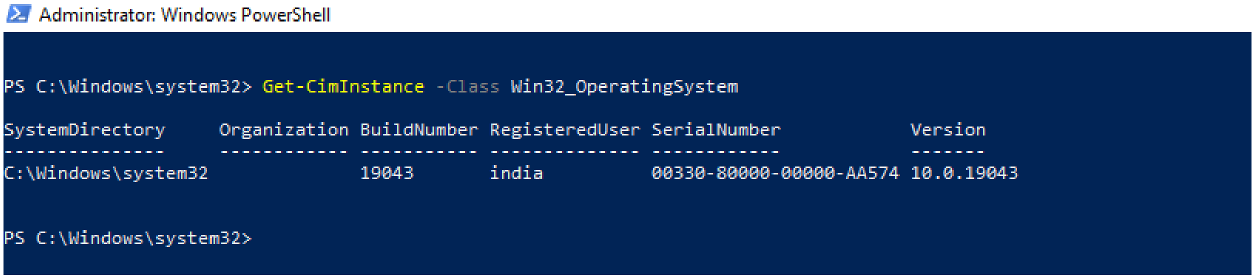
Displaying WMI Class Details

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

```powershell
Get-CimInstance -Class Win32_OperatingSystem
```

OutputCopy



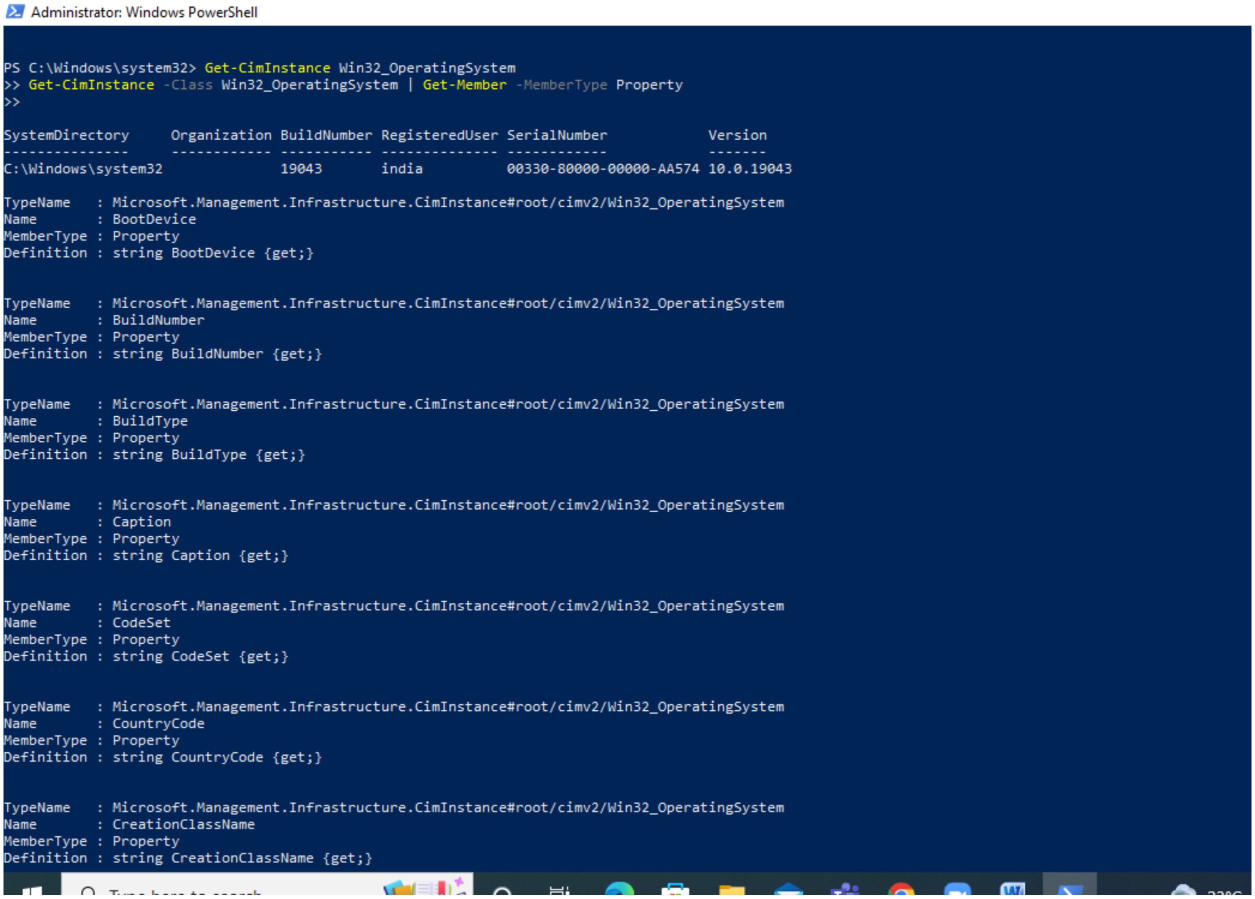
```
PS C:\Windows\system32> Get-CimInstance -Class Win32_OperatingSystem

SystemDirectory     Organization BuildNumber RegisteredUser SerialNumber            Version
---------------     ------------ ----------- -------------- ------------            -------
C:\Windows\system32              19043       india          00330-80000-00000-AA574 10.0.19043


PS C:\Windows\system32>
```

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

```powershell
Get-CimInstance Win32_OperatingSystem
Get-CimInstance -Class Win32_OperatingSystem | Get-Member -MemberType Property
```

OutputCopy



```
PS C:\Windows\system32> Get-CimInstance Win32_OperatingSystem
>> Get-CimInstance -Class Win32_OperatingSystem | Get-Member -MemberType Property
>>

SystemDirectory     Organization BuildNumber RegisteredUser SerialNumber            Version
---------------     ------------ ----------- -------------- ------------            -------
C:\Windows\system32              19043       india          00330-80000-00000-AA574 10.0.19043

TypeName   : Microsoft.Management.Infrastructure.CimInstance#root/cimv2/Win32_OperatingSystem
Name       : BootDevice
MemberType : Property
Definition : string BootDevice {get;}


TypeName   : Microsoft.Management.Infrastructure.CimInstance#root/cimv2/Win32_OperatingSystem
Name       : BuildNumber
MemberType : Property
Definition : string BuildNumber {get;}


TypeName   : Microsoft.Management.Infrastructure.CimInstance#root/cimv2/Win32_OperatingSystem
Name       : BuildType
MemberType : Property
Definition : string BuildType {get;}


TypeName   : Microsoft.Management.Infrastructure.CimInstance#root/cimv2/Win32_OperatingSystem
Name       : Caption
MemberType : Property
Definition : string Caption {get;}


TypeName   : Microsoft.Management.Infrastructure.CimInstance#root/cimv2/Win32_OperatingSystem
Name       : CodeSet
MemberType : Property
Definition : string CodeSet {get;}


TypeName   : Microsoft.Management.Infrastructure.CimInstance#root/cimv2/Win32_OperatingSystem
Name       : CountryCode
MemberType : Property
Definition : string CountryCode {get;}


TypeName   : Microsoft.Management.Infrastructure.CimInstance#root/cimv2/Win32_OperatingSystem
Name       : CreationClassName
MemberType : Property
Definition : string CreationClassName {get;}
```

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
```

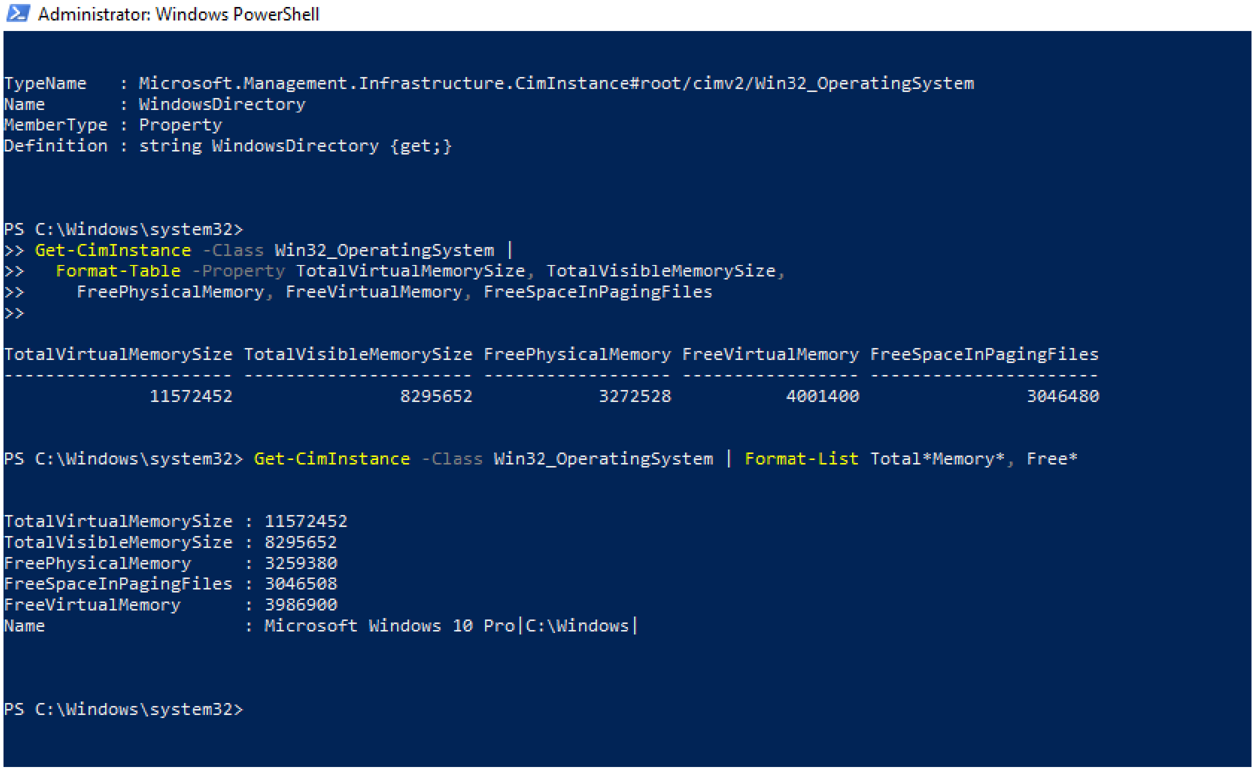
Wildcards work with property names in `Format-Table`, so the final pipeline element can be reduced to `Format-Table -Property Total*Memory*, Free*`

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*
```

Output :



```
TypeName   : Microsoft.Management.Infrastructure.CimInstance#root/cimv2/Win32_OperatingSystem
Name       : WindowsDirectory
MemberType : Property
Definition : string WindowsDirectory {get;}


PS C:\Windows\system32>
>> Get-CimInstance -Class Win32_OperatingSystem |
>>   Format-Table -Property TotalVirtualMemorySize, TotalVisibleMemorySize,
>>     FreePhysicalMemory, FreeVirtualMemory, FreeSpaceInPagingFiles
>>

TotalVirtualMemorySize TotalVisibleMemorySize FreePhysicalMemory FreeVirtualMemory FreeSpaceInPagingFiles
---------------------- ---------------------- ------------------ ----------------- ----------------------
              11572452                8295652            3272528           4001400                3046480

PS C:\Windows\system32> Get-CimInstance -Class Win32_OperatingSystem | Format-List Total*Memory*, Free*


TotalVirtualMemorySize : 11572452
TotalVisibleMemorySize : 8295652
FreePhysicalMemory     : 3259380
FreeSpaceInPagingFiles : 3046508
FreeVirtualMemory      : 3986900
Name                   : Microsoft Windows 10 Pro|C:\Windows|



PS C:\Windows\system32>
```