**Collecting Information About Computers**

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

Cmdlets from **CimCmdlets** module are the most important cmdlets for general system management tasks. All critical subsystem settings are exposed through WMI. Furthermore, WMI treats data as objects that are in collections of one or more items. Because Windows PowerShell also works with objects and has a pipeline that allows you to treat single or multiple objects in the same way, generic WMI access allows you to perform some advanced tasks with very little work.

**Listing Desktop Settings**

We'll begin with a command that collects information about the desktops on the local computer.

PowerShellCopy

Get-CimInstance -ClassName Win32\_Desktop

This returns information for all desktops, whether they are in use or not.

**Note**

Information returned by some WMI classes can be very detailed, and often include metadata about the WMI class.

Because most of these metadata properties have names that begin with **Cim**, you can filter the properties using Select-Object. Specify the **-ExcludeProperty** parameter with "Cim\*" as the value. For example:

PowerShellCopy

Get-CimInstance -ClassName Win32\_Desktop | Select-Object -ExcludeProperty "CIM\*"

To filter out the metadata, use a pipeline operator (|) to send the results of the Get-CimInstance command to Select-Object -ExcludeProperty "CIM\*".

**Listing BIOS Information**

The WMI **Win32\_BIOS** class returns fairly compact and complete information about the system BIOS on the local computer:

PowerShellCopy

Get-CimInstance -ClassName Win32\_BIOS

**Listing Processor Information**

You can retrieve general processor information by using WMI's **Win32\_Processor** class, although you will likely want to filter the information:

PowerShellCopy

Get-CimInstance -ClassName Win32\_Processor | Select-Object -ExcludeProperty "CIM\*"

For a generic description string of the processor family, you can just return the **SystemType** property:

PowerShellCopy

Get-CimInstance -ClassName Win32\_ComputerSystem | Select-Object -Property SystemType

SystemType

----------

X86-based PC

**Listing Computer Manufacturer and Model**

Computer model information is also available from **Win32\_ComputerSystem**. The standard displayed output will not need any filtering to provide OEM data:

PowerShellCopy

Get-CimInstance -ClassName Win32\_ComputerSystem

OutputCopy

Name PrimaryOwnerName Domain TotalPhysicalMemory Model Manufacturer

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

MyPC Jane Doe WORKGROUP 804765696 DA243A-ABA 6415cl NA910 Compaq Presario 06

Your output from commands such as this, which return information directly from some hardware, is only as good as the data you have. Some information is not correctly configured by hardware manufacturers and may therefore be unavailable.

**Listing Installed Hotfixes**

You can list all installed hotfixes by using **Win32\_QuickFixEngineering**:

PowerShellCopy

Get-CimInstance -ClassName Win32\_QuickFixEngineering

This class returns a list of hotfixes that looks like this:

OutputCopy

Source Description HotFixID InstalledBy InstalledOn PSComputerName

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

Security Update KB4048951 Administrator 12/16/2017 .

For more succinct output, you may want to exclude some properties. Although you can use the Get-CimInstance's **Property** parameter to choose only the **HotFixID**, doing so will actually return more information, because all the metadata is displayed by default:

PowerShellCopy

Get-CimInstance -ClassName Win32\_QuickFixEngineering -Property HotFixID

OutputCopy

InstalledOn :

Caption :

Description :

InstallDate :

Name :

Status :

CSName :

FixComments :

HotFixID : KB4533002

InstalledBy :

ServicePackInEffect :

PSComputerName :

CimClass : root/cimv2:Win32\_QuickFixEngineering

CimInstanceProperties : {Caption, Description, InstallDate, Name…}

CimSystemProperties : Microsoft.Management.Infrastructure.CimSystemProperties

...

The additional data is returned, because the **Property** parameter in Get-CimInstance restricts the properties returned from WMI class instances, not the object returned to PowerShell. To reduce the output, use Select-Object:

PowerShellCopy

Get-CimInstance -ClassName Win32\_QuickFixEngineering -Property HotFixId |

Select-Object -Property HotFixId

OutputCopy

HotFixId

--------

KB4048951

**Listing Operating System Version Information**

The **Win32\_OperatingSystem** class properties include version and service pack information. You can explicitly select only these properties to get a version information summary from **Win32\_OperatingSystem**:

PowerShellCopy

Get-CimInstance -ClassName Win32\_OperatingSystem |

Select-Object -Property BuildNumber,BuildType,OSType,ServicePackMajorVersion,ServicePackMinorVersion

You can also use wildcards with the Select-Object's **Property** parameter. Because all the properties beginning with either **Build** or **ServicePack** are important to use here, we can shorten this to the following form:

PowerShellCopy

Get-CimInstance -ClassName Win32\_OperatingSystem | Select-Object -Property Build\*,OSType,ServicePack\*

OutputCopy

BuildNumber : 18362

BuildType : Multiprocessor Free

OSType : 18

ServicePackMajorVersion : 0

ServicePackMinorVersion : 0

**Listing Local Users and Owner**

Local general user information — number of licensed users, current number of users, and owner name — can be found with a selection of **Win32\_OperatingSystem** class properties. You can explicitly select the properties to display like this:

PowerShellCopy

Get-CimInstance -ClassName Win32\_OperatingSystem |

Select-Object -Property NumberOfLicensedUsers,NumberOfUsers,RegisteredUser

A more succinct version using wildcards is:

PowerShellCopy

Get-CimInstance -ClassName Win32\_OperatingSystem | Select-Object -Property \*user\*

**Getting Available Disk Space**

To see the disk space and free space for local drives, you can use the Win32\_LogicalDisk WMI class. You need to see only instances with a DriveType of 3 — the value WMI uses for fixed hard disks.

PowerShellCopy

Get-CimInstance -ClassName Win32\_LogicalDisk -Filter "DriveType=3"

OutputCopy

DeviceID DriveType ProviderName VolumeName Size FreeSpace PSComputerName

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

C: 3 Local Disk 203912880128 65541357568 .

Q: 3 New Volume 122934034432 44298250240 .

PowerShellCopy

Get-CimInstance -ClassName Win32\_LogicalDisk -Filter "DriveType=3" |

Measure-Object -Property FreeSpace,Size -Sum |

Select-Object -Property Property,Sum

OutputCopy

Property Sum

-------- ---

FreeSpace 109839607808

Size 326846914560

**Getting Logon Session Information**

You can get general information about logon sessions associated with users through the **Win32\_LogonSession** WMI class:

PowerShellCopy

Get-CimInstance -ClassName Win32\_LogonSession

**Getting the User Logged on to a Computer**

You can display the user logged on to a particular computer system using Win32\_ComputerSystem. This command returns only the user logged on to the system desktop:

PowerShellCopy

Get-CimInstance -ClassName Win32\_ComputerSystem -Property UserName

**Getting Local Time from a Computer**

You can retrieve the current local time on a specific computer by using the **Win32\_LocalTime** WMI class.

PowerShellCopy

Get-CimInstance -ClassName Win32\_LocalTime

OutputCopy

Day : 23

DayOfWeek : 1

Hour : 8

Milliseconds :

Minute : 52

Month : 12

Quarter : 4

Second : 55

WeekInMonth : 4

Year : 2019

PSComputerName :

**Displaying Service Status**

To view the status of all services on a specific computer, you can locally use the Get-Service cmdlet. For remote systems, you can use the **Win32\_Service** WMI class. If you also use Select-Object to filter the results to **Status**, **Name**, and **DisplayName**, the output format will be almost identical to that from Get-Service:

PowerShellCopy

Get-CimInstance -ClassName Win32\_Service |

Select-Object -Property Status,Name,DisplayName

To allow the complete display of names for the occasional services with extremely long names, you may want to use Format-Table with the **AutoSize** and **Wrap** parameters, to optimize column width and allow long names to wrap instead of being truncated:

PowerShellCopy

Get-CimInstance -ClassName Win32\_Service |

Format-Table -Property Status,Name,DisplayName -AutoSize -Wrap