**A simple powershell script to look up server hardware and OS information**

[SQLPals](https://www.sqlservercentral.com/author/SQLPals), 2019-07-23

FROM: <https://www.sqlservercentral.com/blogs/a-simple-powershell-script-to-look-up-server-hardware-and-os-information?utm_source=ssc&utm_medium=pubemail>

This is a simple powershell script to query and display hardware and OS information from a remote computer.

It uses CIM (Common Information Model) that is available since Powershell version 3 and is the recommended direction.  Please see the following article on why "we" should use CIM instead of the WMI.

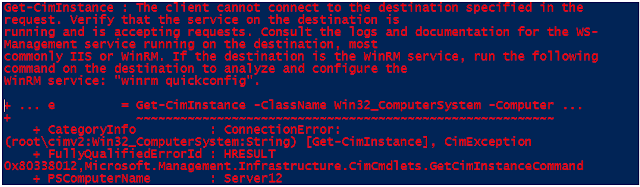
<https://devblogs.microsoft.com/scripting/should-i-use-cim-or-wmi-with-windows-powershell/>

# Specify the server name here

$server = "server1"  
  
  
# pull all the information  
$hardware = Get-CimInstance -ClassName Win32\_ComputerSystem -ComputerName $server  
$OS = Get-CimInstance -ClassName Win32\_OperatingSystem -ComputerName $server  
$CPU = Get-CimInstance -ClassName Win32\_Processor -ComputerName $server  
$PhysicalMemory = Get-CimInstance -ClassName CIM\_PhysicalMemory -ComputerName $server  
$Bios = Get-CimInstance -ClassName Win32\_BIOS -ComputerName $server  
  
$total\_memory = ($PhysicalMemory | measure-object -Property Capacity -sum).sum  
$total\_memory\_gb = $total\_memory / 1024 / 1024 / 1024  
  
# build custom array to get some key properties in a single row  
$server\_summary = New-Object PSObject  
  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name Manufacturer -value $hardware.Manufacturer  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name Model -value $hardware.Model  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name HypervisorPresent -value $hardware.HypervisorPresent  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name Bios -value $Bios.Name  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name OS -value $OS.Caption  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name OSArchitecture -value $OS.OSArchitecture  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name CPUs -value $CPU.count  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name PhySicalMemory\_GB -value $total\_memory\_gb  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name OSVersionNumber -value $OS.Version  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name ServicePackMajorVersion -value $OS.ServicePackMajorVersion  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name ServicePackMinor -value $OS.ServicePackMinorVersion  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name LastBootUpTime -value $OS.LastBootUpTime  
  
# Display the values  
  
# First, lets up the buffer size first so we can see the complete output on the screen  
$Host.UI.RawUI.BufferSize = New-Object Management.Automation.Host.Size (500, 3000)  
  
"summary"  
"======="  
  
$server\_summary | ft -AutoSize  
  
""  
"Detailed Properties"  
"==================="  
  
"Hardware:"  
$hardware | ft -Property \*  
  
"Bios:"  
$Bios | ft -Property \*   
  
"Operating System:"  
$OS | ft -Property \*  
  
"CPUs:"   
$CPU | ft -Property \*   
  
"Physical Memory:"  
$PhysicalMemory | ft -property \*

Caveat:

That worked on most servers but on some I ran into an error with the CIM.

[](https://1.bp.blogspot.com/-x_91CiTpOxM/XTcLLkCnARI/AAAAAAAABHs/aSq8MSppruwx_x8b1uhXbWCd2tWOtaAPwCLcBGAs/s1600/Capture.PNG)

So I tried the solution suggested by the error message, which is to run the **winrm quickconfig** on the remote computer.  That threw message that the "*WinRM service is already running on this machine*", so maybe there is a firewall that is blocking it.

So I decided to go back to the good old faithful WMI for those servers. The powershell methods are still interchangeable between CIM and WMI so all it took was to do a global search/ replace for  Get-CimInstance / Get-WmiObject.

# Specify the server name here

$server = "server1"  
  
  
# pull all the information  
$hardware = Get-WmiObject -ClassName Win32\_ComputerSystem -ComputerName $server  
$OS = Get-WmiObject -ClassName Win32\_OperatingSystem -ComputerName $server  
$CPU = Get-WmiObject -ClassName Win32\_Processor -ComputerName $server  
$PhysicalMemory = Get-WmiObject -ClassName CIM\_PhysicalMemory -ComputerName $server  
$Bios = Get-WmiObject -ClassName Win32\_BIOS -ComputerName $server  
  
$total\_memory = ($PhysicalMemory | measure-object -Property Capacity -sum).sum  
$total\_memory\_gb = $total\_memory / 1024 / 1024 / 1024  
  
# build custom array to get some key properties in a single row  
$server\_summary = New-Object PSObject  
  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name Manufacturer -value $hardware.Manufacturer  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name Model -value $hardware.Model  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name HypervisorPresent -value $hardware.HypervisorPresent  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name Bios -value $Bios.Name  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name OS -value $OS.Caption  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name OSArchitecture -value $OS.OSArchitecture  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name CPUs -value $CPU.count  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name PhySicalMemory\_GB -value $total\_memory\_gb  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name OSVersionNumber -value $OS.Version  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name ServicePackMajorVersion -value $OS.ServicePackMajorVersion  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name ServicePackMinor -value $OS.ServicePackMinorVersion  
Add-Member -inputObject $server\_summary -memberType NoteProperty -Name LastBootUpTime -value $OS.LastBootUpTime  
  
# Display the values  
  
# First, lets up the buffer size first so we can see the complete output on the screen  
$Host.UI.RawUI.BufferSize = New-Object Management.Automation.Host.Size (500, 3000)  
  
"summary"  
"======="  
  
$server\_summary | ft -AutoSize  
  
""  
"Detailed Properties"  
"==================="  
  
"Hardware:"  
$hardware | ft -Property \*  
  
"Bios:"  
$Bios | ft -Property \*   
  
"Operating System:"  
$OS | ft -Property \*  
  
"CPUs:"   
$CPU | ft -Property \*   
  
"Physical Memory:"  
$PhysicalMemory | ft -property \*

Simple enough?  Nah... but I am sticking with the title!