Name Click here to enter name ID Click here to enter id

CIM

Exercise 3.2

# Instructions

Answer all questions directly in this document. You will save and upload this completed document as your homework submission.

# Overview

Working with CIM was made easier and supported better in Windows PowerShell. There are cmdlets to find CIM classes and get instances of CIM classes. They also have better support for WMI query language.

# Requirements

* Windows
* PowerShell (Desktop or Core)

# Task 1—Finding CIM classes

One of the keys to working with CIM is to find the correct classes that accomplish your task. PowerShell makes it pretty easy.

## Steps

1. Find all commands containing CIM:  
   **Get-Command \*cim\***
   1. There’s almost too many to count by hand. Let’s make PowerShell count them.
   2. **(Get-Command \*cim\* -CommandType Alias).Count**   
      How many are aliases? Click or tap here to enter text.
   3. **(Get-Command \*cim\* -CommandType Cmdlet).Count**   
      How many are cmdlets? Click or tap here to enter text.
   4. **(Get-Command \*cim\* -CommandType Application).Count**   
      How many are applications? Click or tap here to enter text.
2. Get all cim classes.  
   Get-CimClass
3. That’s a long list, how long is it?  
   Get-CimClass|Measure # you could also type (Get-CimClass).Count
   1. How many classes are there?Click or tap here to enter text. *(This number should look familiar, it should be the same number you discovered with the* **wbemtest** *utility.)*
4. Now find a class that reveals something you want to find. Example: how much memory is in your computer? You can use wildcards to search for classes. Just like discovering cmdlets, think of a good keyword then use it in your filter. This should filter your list down to a manageable scan.
5. Maybe win32\_PhysicalMemory sounds interesting.
6. Once you have found a class, get it and look at it.   
   $c= Get-CimClass win32\_physicalmemory
   1. Use the **CimClassProperties** property to view the available properties:   
      **$c.CimClassProperties|ft**
      1. How many properties are there? Click or tap here to enter text.
   2. Now look at the **CimClassMethods** property to view the available methods:   
      **$c.CimClassMethods|ft**
      1. How many methods are there? Click or tap here to enter text.
7. You try it. What class would you use to determine what version is the bios on your computer?
   1. What is the class name? Click or tap here to enter text.
   2. How many Properties does it have? Click or tap here to enter text.
   3. How many Methods does it have? Click or tap here to enter text.

# Task 2—Getting CIM Instances

Once you know the class name, you can use PowerShell to get an actual instance of the class with the appropriate data filled in. This is super easy.

## Steps

1. Get an instance of the Win32\_PhysicalMemory class  
   $m=Get-CimInstance win32\_physicalmemory
   1. How many instances were returned? Click or tap here to enter text.
2. View the data  
   $m|ft BankLabel,Manufacturer,SerialNumber,Capacity
   1. For each instance record the BankLabel, Manufacturer, SerialNumber and Capacity  
      Click or tap here to enter text.

# Task 3—WMI Queries with CIM

You can use WMI queries in PowerShell CIM cmdlets as well. Let’s get a process using a query.

## Steps

1. Make sure Notepad is running. Then enter the command (all on one line):  
   Get-CimInstance -Query "select \* from win32\_process where name='notepad.exe'"
2. You should see an instance of a Notepad process.
   1. What is its **WorkingSetSize**? Click or tap here to enter text.
3. Again there’s a convenient alias, so let’s use it. Get the Notepad instance and store it in a variable:   
   $n=gcim -Query "select \* from win32\_process where name='notepad.exe'"
4. View the Cim Methods for the Win32\_Process?  
   (Get-CimClass win32\_process).CimClassMethods
   1. What methods does it have? Click or tap here to enter text. *Reminder: use the* **-join** *operator to combine all of those method names, separated by commas, to one line of output.*
5. Use the terminate method to stop Notepad (this will stop all the retrieved instances):  
   $n|Invoke-CimMethod -MethodName Terminate

# Deliverable

Upload this document with completed answers to I-Learn Canvas.