Advanced Scripting   
WMI Legacy Tools

Last Updated: 4/18/2024 9:07 AM Version 1  
Document Prepared for: CYBER360 Student

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

# Instructions

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

# Overview

Microsoft’s implementation of the Distributed Management Taskforce’s web based management and common information model (CIM) is called Windows Management Instrumentation (WMI). It relies on Windows RPC, which is no longer supported in cross-platform PowerShell Core, but is still accessible using Windows PowerShell 5.1 (Desktop edition). In this exercise you will briefly explore WMI.

# Requirements

Windows

# Task 1—Exploring WMI with **wmic**

Use the command line tool **wmic** to explore WMI.

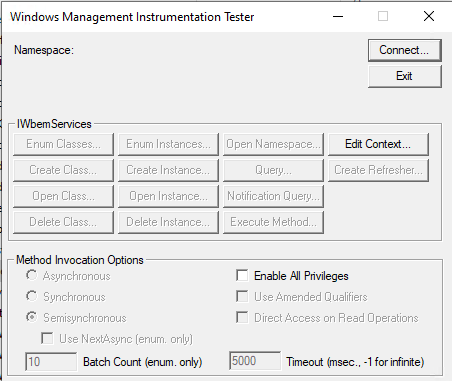
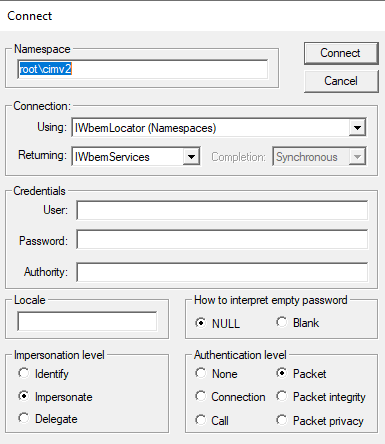
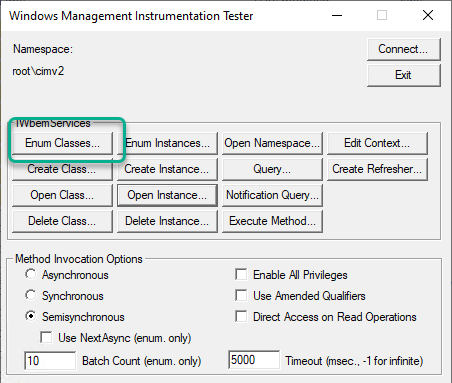
## Steps

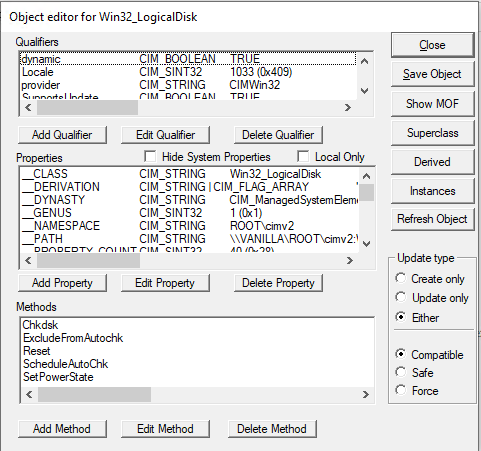
1. Open a CMD.EXE command prompt (or Windows PowerShell) and start **wmic**, which is an interactive command line tool that enables access to WMI.  
   wmic
2. You should see a prompt like  
   wmic:root\cli>
3. The easiest way to retrieve data using wmic is to just type in the name of one of the aliases. You can get a list of aliases by typing: /?  
   /?   
   (*Note: if* /? *doesn’t work, try starting again at step 1, with an* administrator *privileged command prompt.*)
4. Get a list of services running on your computer  
   Service
5. Get a list of disks  
   Logicaldisk
   1. What logical disks are on your system?Click or tap here to enter text.
6. Get a list of volumes  
   Volume
   1. List each volume with its block size and capacity: Click or tap here to enter text.
7. Explore 4 more aliases of your choice, and record your results.
   1. Alias Click or tap here to enter text. Results Click or tap here to enter text.
   2. Alias Click or tap here to enter text. Results Click or tap here to enter text.
   3. Alias Click or tap here to enter text. Results Click or tap here to enter text.
   4. Alias Click or tap here to enter text. Results Click or tap here to enter text.
8. Use the exit command to quit **wmic**:  
   exit

# Task 2—wbemtest

Windows provides a graphical tool to explore WMI as well. It has many more features than **wmic**.

## Steps

1. Using File Explorer, browse to C:\Windows\System32\wbem
2. Run the program **wbemtest**
3. You should see:  
   
4. Tap the connect button. You should now see the connection dialog.  
   
5. There are a lot of options, but for this exercise just tap connect. Now you should see the tester window:  
   
6. Explore the classes.
   1. To get a list of available classes, tap the [Enum Classes…] button. In the Superclass Info dialog, select Recursive, then click [OK]:  
      A screenshot of a computer login box

      Description automatically generated
   2. It may take a minute to run the query. Eventually you will see a list of classes. Browse the list for a minute.
      1. Record the number of objects found Click or tap here to enter text.
   3. Tap [Close]
7. Get an instance of a class
   1. Click the [Open instance…] button
   2. For the class name enter **win32\_logicaldisk** then click [OK]. The description of the class is loaded and displayed. You should see something like this:  
      
   3. You can see the properties and methods of the class. Now, to load the instances with live data, click the [Instances] button in the right column. Now you will see the disks on your system. For example, mine looks like this:  
      A screenshot of a computer program

      Description automatically generated
   4. Double click on one of your disks. This will open up an Object Editor with the data now filled in. Look at the following properties and record their values.
      1. DeviceID Click or tap here to enter text.
      2. DriveType Click or tap here to enter text.
      3. FileSystem Click or tap here to enter text.
   5. Close the Object Editor and Query Result dialogs.
8. Launch Notepad.
9. Get another instance of a class, like you did in step 7, but this time use the class win32\_process. Then get Instances, like you did in step 7.3. You should see many processes.
   1. Find the process with Handle value “0” (it should be first process in the list). Open its Object Editor, like you did in step 7.4, and explore the data. Record the following information:
      1. Description Click or tap here to enter text.
      2. ThreadCount Click or tap here to enter text.
      3. WorkingSetSize Click or tap here to enter text.
   2. Choose a different process (perhaps one nearer the bottom of your list), open its Object Editor, and record the following information:
      1. Handle Click or tap here to enter text.
      2. Description Click or tap here to enter text.
      3. ThreadCount Click or tap here to enter text.
      4. WorkingSetSize Click or tap here to enter text.
10. If you wanted to find and record these properties for the Notepad process, **wbemtest** by itself is kind of a clumsy interface for getting it. What else could you do? Click or tap here to enter text. *(Hint: what PowerShell commands have you used for exploring processes?)*
11. Close each of the Object Editor and Query Result dialogs. Leave the Tester open for the next task.

# Task 3—WMI queries

WMI supports a SQL-style query language. You can use the query language to return specific instances of class or portions of class data. Let’s use a WMI query to find the properties of the Notepad process.

## Steps

1. Click the [Query…] button.
2. Enter the query:  
   **select \* from win32\_process where name='notepad.exe'**
3. Click the Apply button, this should return just the Notepad process(es).
   1. Open (double-click) a process and explore the data. Record the following information:
      1. Handle Click or tap here to enter text.
      2. Description Click or tap here to enter text.
      3. ThreadCount Click or tap here to enter text.
      4. WorkingSetSize Click or tap here to enter text.
4. Try another query, but this time you will only return the properties you are interested in. Enter the query (all on one line):  
   **select Handle, Description, ThreadCount, WorkingSetSize from win32\_process where name='notepad.exe'**
5. From the results look at the properties of an instance. Notice that only those four properties are returned.
6. Close **wbemtest**. (You opened a Notepad app in task 2 step 8; leave it running for upcoming tasks.)

# Task 4—How WMI works

How does WMI know how to create instances of classes and what the classes look like? The answer lies in the **.mof** and **.dll** files. The .mof files are the description of the class, the .dll files are the providers that are called to fill the instances or execute the methods.

## Steps

1. Open your File Explorer and navigate to C:\Windows\System32\wbem
2. Notice there are many MOF files and .dll (Application Extension) files. Notice they tend to come in pairs; there will be a .mof and .dll with the same name.

Note: WMI or CIM classes are NOT .NET classes, nor are they PowerShell classes; they have their own administration-specific object-oriented structure.

# Task 5—WMI cmdlets in Windows PowerShell 5.1

## Steps

1. Launch Windows Powershell (Desktop edition, version 5.1, the default PowerShell that comes bundled in Windows 10 and Windows 11).
2. Enter:  
   **Get-Command -Name \*wmi\***
   1. How many of the listed items are aliases? Click or tap here to enter text.
   2. How many of the listed items are cmdlets? Click or tap here to enter text.
   3. How many of the listed items are applications? Click or tap here to enter text.
3. Enter:  
   **Get-WmiObject -List | more** # (it may take some time before you start to see output)
   1. What is the NameSpace you see at the top of the output? Click or tap here to enter text.
   2. Press [Q] to quit the **more** pager and get back to the PowerShell prompt.
4. Enter:  
   **Get-WmiObject -Class win32\_process | more**
   1. Hit the spacebar a few times to explore the output, then press [Q] to quit **more**.
5. Let’s use this command’s convenient alias. Enter (all on one command line):  
   **gwmi -Class win32\_process -Property Handle, Description, ThreadCount, WorkingSetSize | sort -Descending -Property Description**
   1. What is the Description of the last process in that sorted list? Click or tap here to enter text.
6. Let’s use a query to find the Notepad process.
   1. Enter:  
      **gwmi -Query "select \* from win32\_process where name='notepad.exe'"**   
      What is the value of its ExecutablePath property? Click or tap here to enter text.
   2. Enter (all on one line):  
      **gwmi -Query "select Handle, Description, ThreadCount, WorkingSetSize from win32\_process where name='notepad.exe'"**   
      Have any of these values changed from what you recorded in Task 3 step 3? If so, what changed? Click or tap here to enter text.

# Deliverable

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