Advanced Scripting   
WMI and the CIM

Last Updated: 3/25/2021 9:58 PM Version 1  
Document Prepared for: CIT361 Student

# Name Daniel Harris ID 235868292

# Instructions

Save a copy of this document. 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 is called Windows Management Instrumentation (WMI). 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

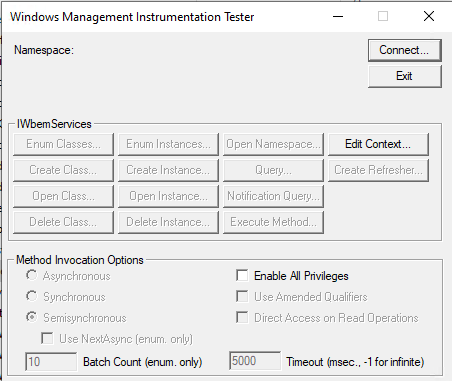
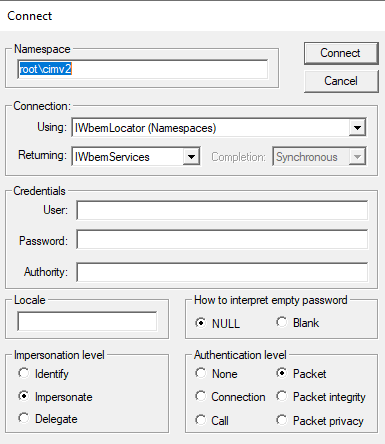
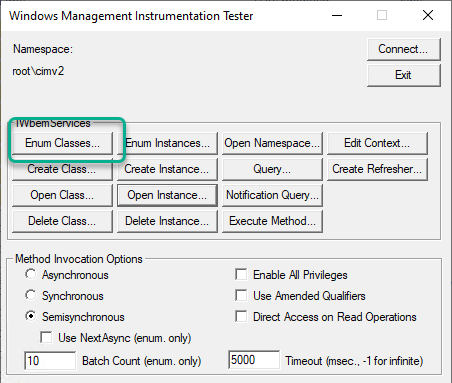
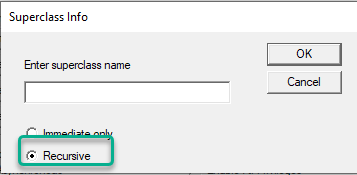
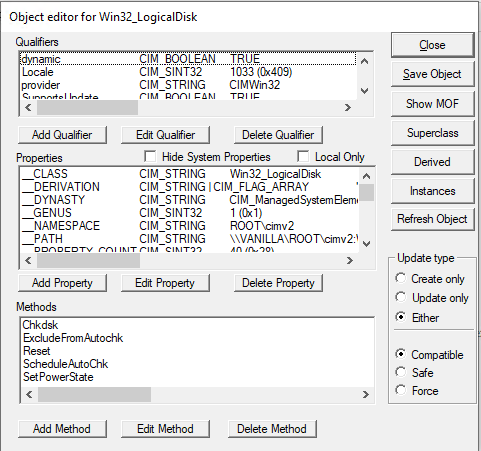
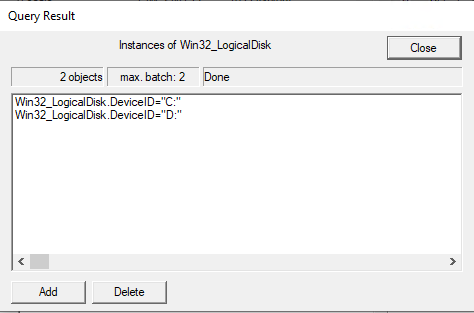
## Steps

1. Open PowerShell or a cmd prompt and start wmic, wmic is an interactive command line tool that allows 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 the name of one of the aliases in. You can get a list of aliases by typing /?, enter  
   /?
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?C:
6. Get a list of volumes  
   Volume
   1. List each volume with its blocksize and capacity. TRUE 4096 TRUE 79871414272 C:\  
       TRUE 4096 FALSE 533721088 \\?\Volume{d7daa129-79d3-40a4-b0bc-58b5432fa130}\  
       TRUE 1024 FALSE 100663296 \\?\Volume{203cc496-fd3d-4b6f-b43d-b61bad77746a}\
7. Explore 4 more aliases record your results
   1. Alias csproduct Results Computer System Product Computer System Product 3551-5169-1342-0854-0796-7811-98 Virtual Machine 621FC592-A9DA-460E-9E1F-E28BD76E1084 Microsoft Corporation Hyper-V UEFI Release v4.1
   2. Alias voltage Results No Instance(s) Available.
   3. Alias dmachannel Results No Instance(s) Available.
   4. Alias bios Results Hyper-V UEFI Release v4.1
8. Use the exit command to quit wmic  
   Exit

# Task 2—wbemtest

Windows provides a graphical tool to explore WMI as will, it has many more features than wmic

## Steps

1. Using file explorer browse to the C:\Windows\System32\wbem folder
2. Run the program wbemtest
3. You should see  
   
4. Click on the connect button, you should now see the connection dialog.  
   
5. There are a lot of options, just click connect. For our purpose just click connect. Now you should see the tester window.  
   
6. Explore the classes.
   1. To get a list of available classes click enum classes. In the superclass dialog select Recursive then click OK  
      
   2. It will 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 1252
   3. Click 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 load the instances with live data. Click the Instances Button. Now you will see the disks on your system. Mine looks like this  
      
   4. Double click on one of your disks. This will open up the editor with the data now filled in. Look at following properties and record their values.
      1. DeviceID C:
      2. DriveType 3(0x3)
      3. FileSystem NTFS
8. Launch notepad
9. Get another instance of a class, this time use the class win32\_process. Then get Instances. You should see many processes.
   1. Open a process and explore the data. Record the following information
      1. Description notepad.exe
      2. ThreadCount 7
      3. WorkingSetSize 14974976
10. What if you wanted to find the information for notepad, what would you do? You need the process id
11. Close the object editor.
12. Close the query results.
13. Clos the object editor.

# Task 3—WMI queries

WMI supports a SQL like query language. You can use the query language to return specific instances of class or portions of class data.

## Steps

1. Use a WMI query to find the notepad data
2. Click the Query button
3. Enter the query  
   **select \* from win32\_process where name='notepad.exe'**
4. Click the Apply button, this should return the notepad process(es)
   1. Open a process and explore the data. Record the following information
      1. Description notepad.exe
      2. ThreadCount 7
      3. WorkingSetSize 14974976
5. Try another query only this time you will only return the properties you are interested in. Enter the query  
   **select Description, ThreadCount, WorkingSetSize from win32\_process where name='notepad.exe'**
6. From the results look at the properties of an instance. Notice that only the three properties are returned.
7. Close wbemtest

# Task 4—How does it work

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 kind of come in pairs. There will be a .mof and .dll with the same name.

# Deliverable

Upload this document with completed answers to i-learn.