# PowerShell and WMI

TRAINSIGNAL

THE GLOBAL LEADER IN PROFESSIONAL COMPUTER TRAINING™

## Windows Management Instrumentation (WMI) Basic Concepts

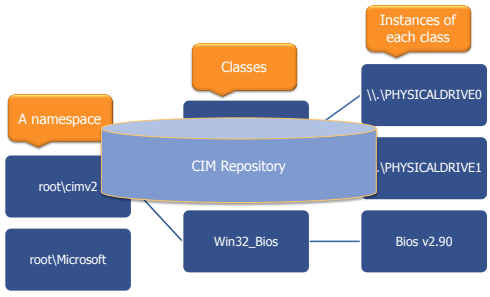
- Hardware and software management information
- WMI is Microsoft's implementation of an industry standard
  - Stored in a local database – the CIM repository
  - Managed by the Winmgmt service
  - Part of Windows since Windows 2000
- Can be accessed locally and remotely via DCOM
- Allows for alternate credentials on remote computers



## WMI Basic Concepts

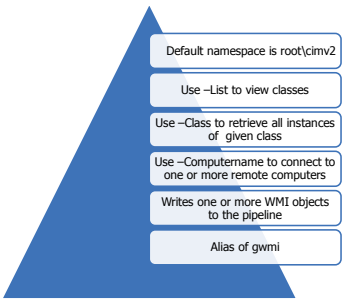
- Information is organized by *Namespace*
  - Namespaces typically align to products or technologies
- *Classes* are definitions of management objects
  - Organized by namespace
  - Have properties and methods
  - Not all classes and properties are available on all versions of Windows
- An *instance* is a real-world manifestation of a class

## WMI Basic Concepts




TRAINSIGNAL

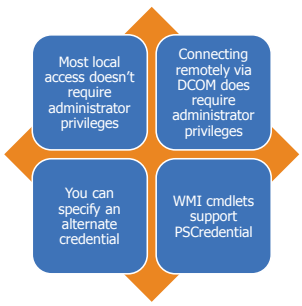
## Using Get-WmiObject




```
PS C:\> get-wmiobject win32_computersystem
```

TRAINSIGNAL

## WMI Security




TRAINSIGNAL

## WMI Credentials

Temporary credentials with password stored as a secure string

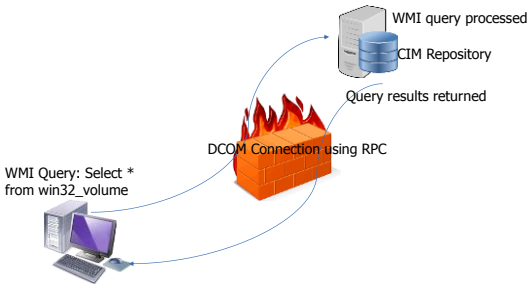
```
PS C:\> get-wmiobject win32_bios -computer
CHI-DC04 -credential
"globomantics\administrator"
```

Create a saved credential using Get-Credential:

```
PS C:\> $cred = Get-Credential
"globomantics\administrator"
PS C:\> get-wmiobject win32_process -computer
"CHI-FP01","CHI-FP02" -credential $cred
```



## WMI Queries






## Retrieving WMI Information

Default namespace is Root\cimv2

Use Select query (-Query)

Retrieve all instances and properties from a given class

```
PS C:\> get-wmiobject -query "Select * from win32_bios"
```

Retrieve instances that meet some criteria

```
PS C:\> get-wmiobject -query "Select
DeviceID,Size,Freespace from Win32__LogicalDisk Where
DriveType=3"
```

Use legacy operators for filtering comparisons

## Retrieving WMI Information

Use a filter (-Filter)
…the Where portion of a select query

```
PS C:\> get-wmiobject Win32__LogicalDisk
-filter "DriveType=3"
```

This returns all properties but you can pipe to Select-Object

```
PS C:\> get-wmiobject win32__LogicalDisk
-filter "DriveType=3" | Select
DeviceID,Size,Freespace
```



## Retrieving WMI Information

Use –Property to return a subset of properties for all instances of a given class

You still get system classes

Filter them out with Select-Object or a format cmdlet

```
PS C:\> get-wmiobject win32_logicaldisk
-Property deviceid,size,freespace
```

```
PS C:\> get-wmiobject win32_logicaldisk
-Property deviceid,size,freespace -filter
"drivetype=3" | format-table -autosize
```



## Authentication and Privileges

Set authentication level for WMI communications
Required for some WMI queries

| Value | Level | Description |
|---|---|---|
| 0 | Default | Default |
| 1 | None | No authentication |
| 2 | Connect | Client authenticates on connection |
| 3 | Call | Authentication at beginning of each call from the server |
| 4 | Packet | Authentication on all data received from client |
| 5 | PacketIntregity | All data is authenticated and verified |
| 6 | PacketPrivacy | All data is authenticated, verified and encrypted |

```
PS C:\> Get-WmiObject -Namespace root\webadministration
-Computer CHI-WEB01 –class site -Authentication
Packetprivacy
```

**Authentication and Privileges**

Some WMI operations or queries require special privileges

No easy way to know when you'll need it.

```
PS C:\> get-wmiobject win32_ntlogevent
-filter "logfile='security'" -computer
CHI-DC01 -EnableAllPrivileges -asjob
```



## Getting WMI Objects

**Invoking WMI Methods**

Methods can only be invoked on a single object

Use Invoke-WMIMethod

- Depending on the method you may need a specific WMI object
- Or a reference to a WMI class
- Use –ArgumentList to pass method parameters

The ReturnValue property indicates success or error

# Invoking WMI Methods

## WMI Shortcuts

**[wmi]**

- If you know the exact WMI object you want
- `PS C:\> [wmi]"root\cimv2:win32_service.name='bits'"`

**[wmiclass]**

- Reference an entire class
- `PS C:\> [wmiclass]"Win32_Process"`

**[wmisearcher]**

- Easily find objects in WMI
- `PS C:\> ([wmisearcher]"Select * from win32_service where startmode='disabled'").Get()`



## WMI Tips and Tricks

Convert date times with the ConvertToDateTime() method

Look for different properties to reflect computer name

PSComputername is an alias for __SERVER property

Don't be afraid to reformat properties

Run long running WMI commands as jobs

Read About_WMI_Cmdlets

Read About_WMI

# WMI Tips in Action

**Lab**

1. Get all instances of the Win32_Volume class on your computer.

2. Repeat step 1 but only fixed volumes and select its name, size, free space and the file system. If you want, format the sizes in MB.

3. Get the Win32_OperatingSystem class and display the computer name, the name of the operating system, when it was installed (in a friendly format), when it last booted (in a friendly format), a calculated property for the uptime and a property that indicates if it is 64 bit or not.

4. If you have remote computers you can query, repeat the previous steps using alternate credentials where possible. Include the computer name in your output.