Microsoft

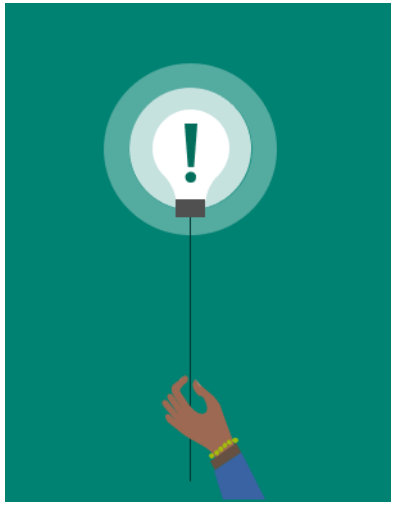
WMI & CIM

Module 2

Objectives

Learnings covered:

- Introduction to WMI & CIM
- Understanding CIM and WMI
- Querying data by using CIM and WMI
- Making changes by using CIM and WMI



Introduction to WMI & CIM

Architecture and technologies

	CIM	WMI
Requires WMF 2.0 or newer	Yes	No
Requires that remoting be enabled	Yes	No
Offers cross-platform compatibility—that is, supports non-Windows computers	Yes	No
Requires a single firewall port exception	Yes	No
Supports session-based connections	Yes	No
Supports ad-hoc connections	Yes	Yes
Supports Windows 11, Windows 10, Windows 8.1, Windows 8, and Windows 7	Yes	Yes
Supports Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2, and Windows Server 2008	Yes	Yes
Requires **the remote administration** firewall exceptions	Yes (one port)	Yes (RPC & High ports)

Understanding the repository

The repository used by CIM and WMI is organized into namespaces

Namespaces organize related classes:

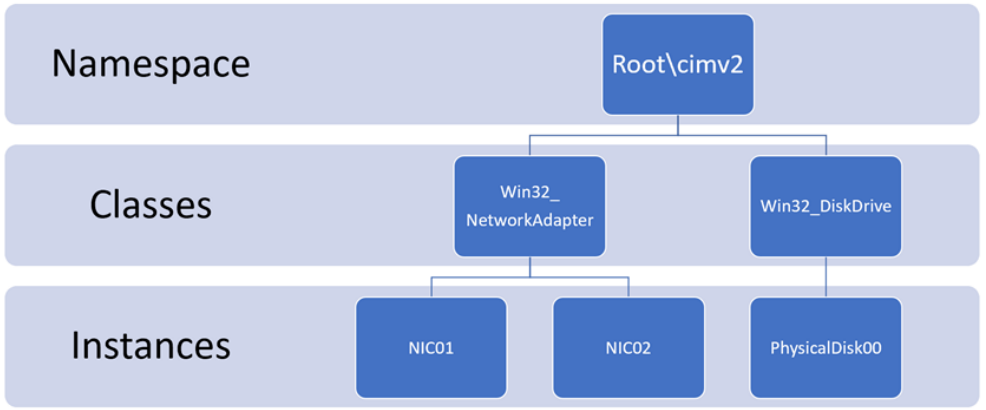
- Toggle through the namespaces by using tab completion in the **Get-CimInstance -Namespace** cmdlet

Classes represent manageable components

An instances is an actual occurrence of a class

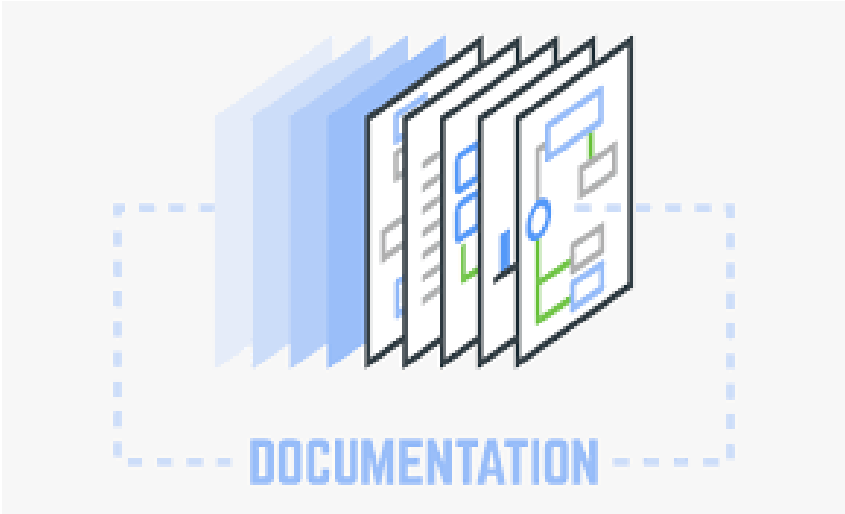
An instances has:

- Properties that describe the instance's attributes
- Methods that cause the instance perform an action



Finding documentation

- The fastest way to find documentation is to type a repository class name into an Internet search engine.
- Classes in the root\CIMv2 namespace are typically well documented.
- Classes from other namespaces are typically not well documented.



Listing Namespaces

Listing namespaces helps you discover what the repository on your computer contains.

To list the namespaces, run the following command:

```
Get-WmiObject -Namespace root -List -Recurse | Select -Unique __NAMESPACE
```

```
Get-CimInstance -Namespace Root -ClassName __Namespace
```

CIM commands offer tab completion for the
-Namespace parameter

Listing Classes

Listing classes in alphabetical order can make it easier to decide whether the class you need exists.

To produce an alphabetical list of classes in the root\CIMv2 namespace, run either of following commands:

- Get-WmiObject -Namespace root\cimv2 –List |
Sort Name
- Get-CimClass –Namespace root\CIMv2 |
Sort CimClassName

Demonstration

WMI Explorer

Querying instances

- Query by specifying a class name
- Include *-Namespace* if the class is not in **root\CIMv2**
- Include *-Filter* to restrict the instances that the command returns

To retrieve only the instances of **Win32_LogicalDisk** for which the **DriveType** property is 3, run either of the following commands:

```
Get-WmiObject -Class Win32_LogicalDisk -Filter "DriveType=3“
Get-CimInstance -ClassName Win32_LogicalDisk –Filter "DriveType=3"
```

Note that filter operators differ from Windows PowerShell comparison operators

Using WQL (SQL for WMI)

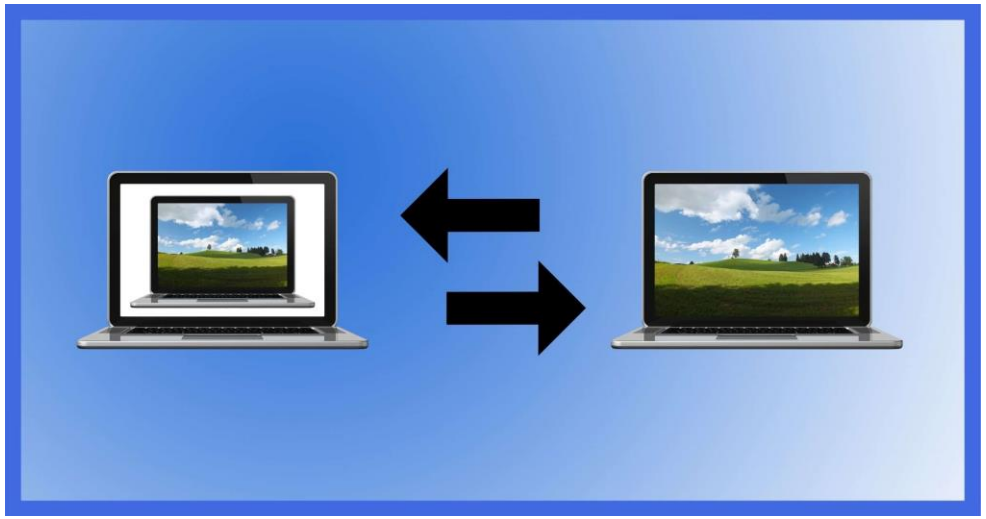
- WQL is a SQL like querying language.
- Use the -Query parameter to provide a WQL query

```
Get-CimInstance -Query "SELECT * FROM Win32_LogicalDisk WHERE DriveType = 3“
```

```
Get-WmiObject -Query "SELECT * FROM Win32_LogicalDisk WHERE DriveType = 3"
```

Connecting to remote computers

- You use -ComputerName to query from a remote computer
- You use -Credential to specify an alternate credential for remote connections using **WmiObject only**
- Get-WmiObject -ComputerName Contoso-DC1
  -Credential Contoso\Administrator
  -Class Win32_BIOS
- The CIM equivalent **does not support** -Credential
- Get-CimInstance -ClassName Win32_BIOS -ComputerName Contoso-DC1 – **THIS WILL NOT WORK !!!!**
  Use Invoke-Command instead !
- WMI uses DCOM
- CIM uses WinRM for ad-hoc connections



Using CIM sessions

- CIM sessions:
- Are reusable, persistent, and authenticated connections to a remote computer
- Can be created and stored in a variable
- Allow you to pass a CIM session object in the -CimSession parameter instead of using -ComputerName to target the computer in the specified session
- Can be manually closed when no longer needed

Discovering Methods

- Many repository classes include methods
- A method tells an object to perform a task or action
- Repository class methods typically reconfigure the manageable component that the class represents
- Use Get-Member to discover the methods of a class
- Note that the output does not explain how to use a method, so you need documentation for that

Finding documentation for methods

Methods

The **Win32_Service** class has these methods.

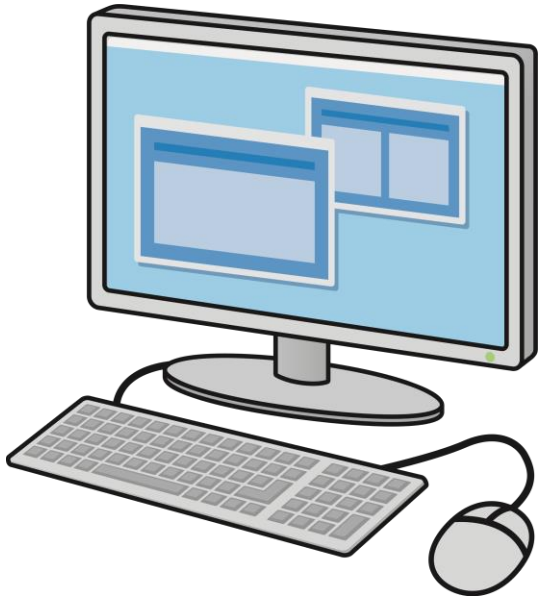
Method	Description
Change	Modifies a service.
ChangeStartMode	Modifies the start mode of a service.
Create	Creates a new service.
Delete	Deletes an existing service.
GetSecurityDescriptor	Returns the security descriptor that controls access to the service.
InterrogateService	Requests that a service update its state to the service manager.
PauseService	Attempts to place a service in the paused state.
ResumeService	Attempts to place a service in the resumed state.
SetSecurityDescriptor	Writes an updated version of the security descriptor that controls access to the service.
StartService	Attempts to place a service into the startup state.
StopService	Places a service in the stopped state.
UserControlService	Attempts to send a user-defined control code to a service.

Invoking Methods

- The ways to invoke a method are:
  - Invoke-WmiMethod
  - Invoke-CimMethod
- Both Invoke techniques can be used with or can accept a pipeline object from the corresponding Get command
- The returned object includes a *ReturnValue* parameter:
  - Zero typically means success
  - For other values, see the documentation

RPC & Dynamic RPC

Client Port 49500



RPC 135 TCP

Listening on Port 51000

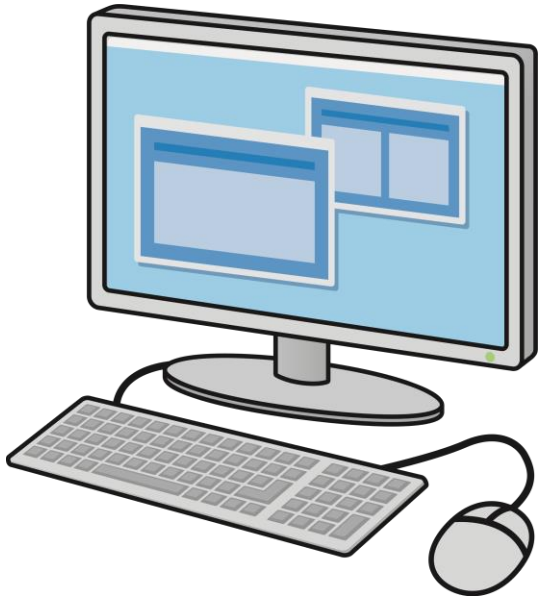
Source Port 49500
Destination Port 51000

Server Port 51000

Windows XP and below
1024 – 5000
Windows Vista and up
49152 - 65535

Windows 2003 and below
1024 – 5000
Windows 2008 and up
49152 - 65535

Windows Remote Management
WIN-RM



5985 TCP

Windows PowerShell 3.0
.NET Framework 4.0
Windows Remote Management 3.0

Questions?

Demonstration

WMI & CIM