

# Managing Computers using CIM Commands

---



**Liam Cleary**

CEO / MICROSOFT MVP / MICROSOFT CERTIFIED TRAINER

@shareplicity [www.shareplicity.com](http://www.shareplicity.com) | @helloitsliam [www.helloitsliam.com](http://www.helloitsliam.com)



# Overview



## Available CIM Commands

### Connecting to Remote Computers

- Session versus Computer Name
- Execute Commands



# Available CIM Commands

---



All CIM Commands are  
available within the  
*"CimCmdlets"* module



# Available CIM Commands: Get

**Get-CimAssociatedInstance**

**Get-CimClass**

**Get-CimInstance**

**Get-CimSession**



# Available CIM Commands: New

**New-CimInstance**

**New-CimSession**

**New-CimSessionOption**



# Available CIM Commands: Import and Invoke

**Import-BinaryMiLog**

**Invoke-CimMethod**



# Available CIM Commands: Other

**Register-CmdIndicationEvent**

**Remove-CimInstance**

**Remove-CimSession**

**Set-CimInstance**





# View all CIM Commands

**# Retrieve all CIM Commands**

```
Get-Command -Module CimCmdlets | Select-Object -Property Name
```

**# View help about specific CIM Command**

```
Get-Help Get-CimInstance -ShowWindow
```



# Demo



**View all CIM Commands**

**Retrieve help for specific CIM Command**



# Connecting to Remote Computers

---



# Connecting to Remote Computers



**Get-CimInstance**  
Creates a temporary session



**New-CimSession**  
Create a session that is open until  
you close it

# Creating Remote Sessions

**# Create a Temporary CIM Session and return Information**

```
$computer = "Trainer"
Get-CimInstance `
    -ComputerName $computer `
    -Namespace "root/cimv2/TerminalServices" `
    -ClassName Win32_TerminalServiceSetting
```

**# Create a CIM Session and Close the CIM Session**

```
$computer = "Trainer"
$sessionName = "TrainingSession"
$session = New-CimSession -Name $sessionName -ComputerName $computer
Remove-CimSession -CimSession $session
```



# Session versus Computer Name

---



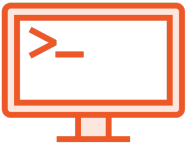
# The CIM Computer Name Parameter



Create single connection to target client or server



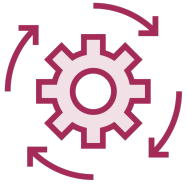
Starts and Stops the session automatically



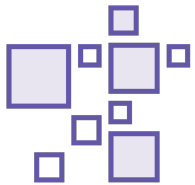
Usually for executing single commands against client or server



# The CIM Session Command



Can control the sessions, either temporary or permanent (until close)



Used when needing to run multiple commands against remote targets



Faster Performance for execution of commands



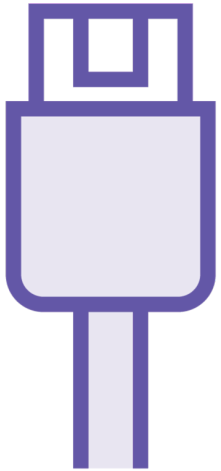


# Execute Commands

---



# CIM Session Options



Connection  
Protocol



Proxy  
Authentication



Proxy Credential



Use SSL



# Create CIM Session with Options

**# Set CIM Session using DCOM as the Protocol**

```
New-CimSessionOption -Protocol DCOM
```

**# Set CIM Session using WSMAN as the Protocol**

```
New-CimSessionOption -Protocol Wsman
```

**# Set CIM Session Properties**

```
$cred = Get-Credential
```

```
New-CimSessionOption `
    -ProxyAuthentication Kerberos `
    -ProxyCredential $cred `
    -SkipCACheck `
    -SkipCNCheck `
    -UseSsl
```



# Combining CIM Commands



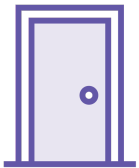
Execute **New-CimSession** to initiate a CIM Session



Call **Get-CimSession** to retrieve the created CIM Session, then retrieve the specified CIM Instance, by using **Get-CimInstance**



Execute CIM commands by calling the **Invoke-CimMethod**



Retrieve the created CIM Session using **Get-CimSession**, then remove the CIM Session using **Remove-CimSession**



# Example Script: Remote Desktop Configuration

```
$computer = "Trainer"
$computertest = Test-Connection $computer -Quiet -Count 1 -TimeToLive 10

if($computertest -eq "True" )
{
    $session = New-CimSession `
        -Name "TerminalServices" `
        -ComputerName $computer

    # Code to execute
}
```



# Example Script: Remote Desktop Configuration

```
$tsSetting = Get-CimSession `
    -Name $session.Name | Get-CimInstance `
    -Namespace root/cimv2/TerminalServices `
    -ClassName Win32_TerminalServiceSetting

$tsGeneralSetting = Get-CimSession `
    -Name $session.Name | Get-CimInstance `
    -Namespace root/cimv2/TerminalServices `
    -ClassName Win32_TSGeneralSetting
```



# Example Script: Remote Desktop Configuration

```
$tsSetting | Invoke-CimMethod `
    -MethodName SetAllowTSConnections `
    -Arguments @{AllowTSConnections=1;ModifyFirewallException=1} | `
        Format-Table -Property PSComputerName,ReturnValue -AutoSize

$tsGeneralSetting | Invoke-CimMethod `
    -MethodName SetUserAuthenticationRequired `
    -Arguments @{UserAuthenticationRequired=1} | Out-Null

Get-CimSession -Name $session.Name | Remove-CimSession
```



# Demo



**Modify CIM Session using Options**

**Execute CIM Commands**





# Summary



**Reviewed the available CIM Commands**

**Understood how to connect to Remote Computers**

**Executed commands on Remote Computers**

- Used the Session Command
- Using the Computer Name Parameter



Up Next:  
Performing Tasks on Remote Machines  
using CIM

---

