# VirtualBox List Installed VMs

A cool feature of VirtualBox is the COM Objects that it exposes for use with external applications such as PowerShell. I am still playing around with the basics here so bear with me.

My first challenge for myself was to get a list of all the installed VMS on my computer using the VirtualBox COM Object. This was made simple with PowerShell thanks to the Get-Member (gm) command, which enumerates all the methods and properties of any object (in this case the VirtualBox.VirtualBox object).

To get a list of all the methods and properties available I ran this command.

```
New-Object -ComObject VirtualBox.VirtualBox | gm
```

Which gave me the following back.

```
   TypeName: System.__ComObject#{3f36e024-7fed-4f20-a02c-9158a82b44e6}

Name                        MemberType Definition
----                        ---------- ----------
CheckFirmwarePresent        Method     int CheckFirmwarePresent (FirmwareType,...
CreateAppliance             Method     IAppliance CreateAppliance ()
CreateDHCPServer            Method     IDHCPServer CreateDHCPServer (string)
CreateHardDisk              Method     IMedium CreateHardDisk (string, string)
CreateLegacyMachine         Method     IMachine CreateLegacyMachine (string, s...
CreateMachine               Method     IMachine CreateMachine (string, string,...
CreateSharedFolder          Method     void CreateSharedFolder (string, string...
FindDHCPServerByNetworkName Method     IDHCPServer FindDHCPServerByNetworkName...
FindDVDImage                Method     IMedium FindDVDImage (string)
FindFloppyImage             Method     IMedium FindFloppyImage (string)
FindHardDisk                Method     IMedium FindHardDisk (string)
FindMachine                 Method     IMachine FindMachine (string)
GetDVDImage                 Method     IMedium GetDVDImage (string)
GetExtraData                Method     string GetExtraData (string)
GetExtraDataKeys            Method     SAFEARRAY(string) GetExtraDataKeys ()
GetFloppyImage              Method     IMedium GetFloppyImage (string)
GetGuestOSType              Method     IGuestOSType GetGuestOSType (string)
GetHardDisk                 Method     IMedium GetHardDisk (string)
GetMachine                  Method     IMachine GetMachine (string)
OpenDVDImage                Method     IMedium OpenDVDImage (string, string)
OpenExistingSession         Method     void OpenExistingSession (ISession, str...
OpenFloppyImage             Method     IMedium OpenFloppyImage (string, string)
OpenHardDisk                Method     IMedium OpenHardDisk (string, AccessMod...
OpenMachine                 Method     IMachine OpenMachine (string)
OpenRemoteSession           Method     IProgress OpenRemoteSession (ISession, ...
OpenSession                 Method     void OpenSession (ISession, string)
RegisterCallback            Method     void RegisterCallback (IVirtualBoxCallb...
RegisterMachine             Method     void RegisterMachine (IMachine)
```

```
RemoveDHCPServer            Method     void RemoveDHCPServer (IDHCPServer)
RemoveSharedFolder          Method     void RemoveSharedFolder (string)
SetExtraData                Method     void SetExtraData (string, string)
UnregisterCallback          Method     void UnregisterCallback (IVirtualBoxCal...
UnregisterMachine           Method     IMachine UnregisterMachine (string)
WaitForPropertyChange       Method     void WaitForPropertyChange (string, uin...
DHCPServers                 Property   SAFEARRAY(IDHCPServer) DHCPServers () {...
DVDImages                   Property   SAFEARRAY(IMedium) DVDImages () {get}
FloppyImages                Property   SAFEARRAY(IMedium) FloppyImages () {get}
GuestOSTypes                Property   SAFEARRAY(IGuestOSType) GuestOSTypes ()...
HardDisks                   Property   SAFEARRAY(IMedium) HardDisks () {get}
HomeFolder                  Property   string HomeFolder () {get}
Host                        Property   IHost Host () {get}
Machines                    Property   SAFEARRAY(IMachine) Machines () {get}
PackageType                 Property   string PackageType () {get}
PerformanceCollector        Property   IPerformanceCollector PerformanceCollec...
ProgressOperations          Property   SAFEARRAY(IProgress) ProgressOperations...
Revision                    Property   uint Revision () {get}
SettingsFilePath            Property   string SettingsFilePath () {get}
SharedFolders               Property   SAFEARRAY(ISharedFolder) SharedFolders ...
SystemProperties            Property   ISystemProperties SystemProperties () {...
Version                     Property   string Version () {get}
```

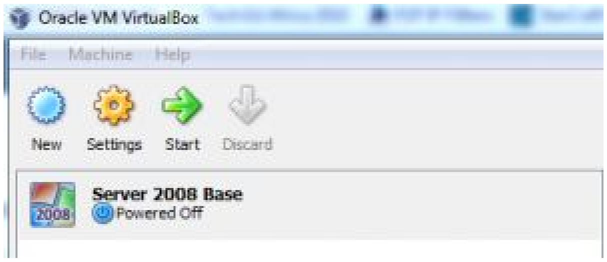
From there I made use of the Machines property to get a collection of all the virtual machines on my computer, the finished script looks something like this.

```
# Check if VBox is installed
if( (Test-Path -Path hklm:\SOFTWARE\Oracle\VirtualBox) -eq $false ){
    Write-Host "Virtual Box is not installed!"
    Exit
}

# Create the VBox COM Object
$comVB   = New-Object -ComObject VirtualBox.VirtualBox
$arrVMS  = $comVB.Machines
$counter = 1

# Count found VMs
if( $arrVMS.Length -eq 0 ){
    Write-Host "There are no VMs installed"
    Exit
}

# List the names of the VMs
foreach( $vm in $arrVMS ){
    $info = "VM" + $counter + ": " + $vm.Name + " (" + $vm.OSTypeId + ")"
    Write-Host $info
    $counter++
}
```

The following was written to my console.

```
        VM1: Server 2008 Base (Windows2008)
```

Just checking to ensure this is valid...



I am starting to like PowerShell :)