

# Managing VMware with PowerShell FAQ

---

## Managing VMware with PowerShell -- Frequently Asked Questions

This document contains frequently asked questions about managing VMware with PowerShell. This document is a wiki page, so if you have contributions or corrections, feel free to make them by editing the document directly.

Note: If you are using Internet Explorer, the code samples below do not render properly. To deal with this problem you can click "Edit Document" on the left hand side of the page, switch to plain text view, and copy text out of the document.

### General VI Toolkit (for Windows) Questions:

Q: What is the VI Toolkit (for Windows)?

A: The VI Toolkit (for Windows) is a snap-in to Windows PowerShell that provides administration and automation for VMware Virtual Infrastructure.

Q: Does this FAQ cover a particular version of the VI Toolkit (for Windows)?

A: Yes, this FAQ specifically covers the VI Toolkit (for Windows) 1.0, build 103777.

Q: What VMware versions are supported?

A: Version 1.0 supports: VMware ESX 3.0.x, VMware ESX 3.5, VMware ESXi 3.5, VMware VirtualCenter 2.0 and VMware VirtualCenter 2.5. In addition, with the VI Toolkit (for Windows) you can manage multiple hosts at the same time, even if they are different versions of VI.

Q: What client platforms does the VI Toolkit (for Windows) support?

A: Windows XP, 2003 and Vista, 32 and 64 bit versions.

Q: What versions of PowerShell are compatible with the VI Toolkit (for Windows)?

A: The toolkit has been tested with PowerShell version 1, as well as the Community Technology Preview releases of PowerShell version 2.

### General PowerShell Questions:

Q: How do I stop PowerShell from prompting me for confirmation before taking action?

A: Using parameter -Confirm one can bypass the confirmation messagebox. This parameter takes input as a Boolean.

Use

```
-Confirm:$false
```

Q: Can I manage more than one VirtualCenter server at a time? Can I connect to both ESX and VC at the same time.

A: Yes, but the syntax changes when you want to manage more than one host.

```
$vcs = @()  
$vcs += connect-viserver <VC 1>  
$vcs += connect-viserver <VC 2>  
...  
$vcs += connect-viserver <VC N>  
  
# Snapshot all VMs across all VirtualCenter servers.  
get-vm -server $vcs | new-snapshot
```

## Initial Steps

Q: How do I connect to a VirtualCenter Server or Host Server?

A: Use the Connect-VIServer cmdlet and provide a value to the Server parameter.

```
connect-viserver -server  
<server> -user <user> -password <pwd>
```

Q: What if my host server listens on a non-default port?

A: Use the Port parameter to specify the port number.

```
connect-viserver -server  
<server> -user <user> -password <pwd> -port  
<port>
```

## VM Lifecycle:

Q: How do I create a VM?

A: New virtual machine can be created using New-VM cmdlet. Before, executing this cmdlet first you need to connect to server. Below script would create a new virtual machine in the same host.

```
New-VM -Name  
"Target-VM" -Host (Get-VMHost)
```

Q: How do I create a VM on a particular datastore?

A: To create a virtual machine on a particular we need the reference of the datastore. This script creates a new virtual machine on a particular datastore.

```
connect-viserver -server  
<server> -user <user> -password <pwd>  
$datastore = Get-Datastore -Name  
"Select-Datastore-Name"  
New-VM -Name  
"Target-VM" -Host (Get-VMHost -Datastore  
$datastore)
```

Q: How can I import an existing VM?

A: This script will import a VM to the vm folder.

```
connect-viserver -server  
<server> -user <user> -password <pwd>  
  
# If importing to a cluster uncomment the line below
```

```
# $rp = get-cluster <cluster name> | get-resourcepool |  
get-view  
  
# If importing to a non-cluster host uncomment the line below  
# $rp = get-vmhost <host name> | get-resourcepool | get-view  
  
$vmFolder = get-view (get-folder vm).id  
$vmFolder.RegisterVM_Task(  
"[<Datastore Name>] <Path To  
VMX>.vmx",  
"<VM Name>", $false, $rp.MoRef,  
$null)
```

Q: How can I determine the connection state of CD ROM drives and floppy drives?

A: Connection state of CD drive and Floppy drive can be determined by property "Connected:". This return Boolean value true for connected and false for disconnected state. Below script would determine the connection state of CD and floppy drive.

```
connect-viserver -server  
<server> -user <user> -password <pwd>  
Write-Output  
"FloppyDrive Status:"  
Get-vm | Select-Object @  
{ Name=  
"Status"; Expression=  
{(Get-FloppyDrive -VM  
$_).ConnectionState.Connected  
}  
}, @  
{ Name=  
"Name"; Expression=  
{$_ .Name  
}  
} | ft  
Write-Output  
"CDDrive Status:"  
Get-vm | Select-Object @  
{ Name=  
"Status"; Expression=  
{(Get-CDDrive -VM  
$_).ConnectionState.Connected  
}  
}, @  
{ Name=  
"Name"; Expression=  
{$_ .Name  
}
```

```
} | ft
```

Q: How can I set VM poweron options, such as configuring tools to automatically upgrade?

A: By setting the ToolsUpgradePolicy property to upgradeAtPowerCycle. With this setting Tools upgrade is automatically performed on the virtual machine and rebooted if necessary. Below is the script for the same.

```
connect-viserver -server  
<server> -user <user> -password <pwd>  
$vm = Get-View (Get-VM VMName).Id  
$vmConfigSpec = New-Object VMware.Vim.VirtualMachineConfigSpec  
$vmConfigSpec.Tools = New-Object VMware.Vim.ToolsConfigInfo  
$vmConfigSpec.Tools.ToolsUpgradePolicy =  
"UpgradeAtPowerCycle"  
$vm.ReconfigVM($vmConfigSpec)
```

Q: How do I set resources, such as limits and reservations, on a VM?

A: Various resources of a virtual machine can be set using the ReconfigVM\_task method on virtual machine. To setting the reservation and limits of a virtual machine an instance of VirtualMachienConfigSpec would be created and all the required resources can be set in configspec before calling the ReconfigVM\_Task. Below is the script for the same.

```
connect-viserver -Server  
<Server> -User <user> -Password <password>  
$vm = Get-VM VMName | %  
{Get-View $_.ID  
}  
$spec = new-object VMware.Vim.VirtualMachineConfigSpec  
$spec.CPUAllocation = New-Object VMware.Vim.ResourceAllocationInfo  
$spec.CpuAllocation.Shares = New-Object VMware.Vim.SharesInfo  
$spec.CpuAllocation.Shares.Level =  
"normal"  
$spec.CpuAllocation.Limit = -1  
$vm.ReconfigVM_Task($spec)
```

Q: How can I determine the version of VMware tools my VMs are using?

A: Toolversion property under the config property of virtual machine object having this information. To get this information we can use the below script.

```
connect-viserver -Server  
<Server> -User <user> -Password <password>  
get-vm VMName | %  
{ get-view $_.ID  
} | select Name, @  
{ Name=  
"ToolsVersion"; Expression=  
{$_ .config.tools.toolsVersion  
}  
}
```

Q: How can I change a VM's IP address?

A: This script can be used to change the IP setting.

```
connect-viserver -server
<server> -user <user> -password <pwd>

$vmclonespec = New-Object VMware.Vim.VirtualMachineCloneSpec
$vmclonespec.Customization = New-Object
VMware.Vim.CustomizationSpec
$vmclonespec.Customization.NicSettingMap = @(New-Object
VMware.Vim.CustomizationAdapterMapping)
$vmclonespec.Customization.NicSettingMap[0].Adapter = New-Object
VMware.Vim.CustomizationIPSettings
$vmclonespec.Customization.NicSettingMap[0].Adapter.ip = New-Object
VMware.Vim.CustomizationFixedIp
$vmclonespec.Customization.NicSettingMap[0].Adapter.Ip.IpAddress =
"<Target IP Address>"
$vmclonespec.Customization.Identity = New-Object
vmware.Vim.CustomizationIdentitySettings
$vmclonespec.Customization.GlobalIPSettings = New-Object
VMware.Vim.CustomizationGlobalIPSettings

$vmclonespec.config = New-Object
VMware.Vim.VirtualMachineConfigSpec
$vmclonespec.location = New-Object
VMware.Vim.VirtualMachineRelocateSpec

$vmclonespec.powerOn = $false
$vmclonespec.template = $false
$name =
"Test"

$target = Get-Folder -Name TARGETFOLDER | %
{Get-View $_.ID
}
$vmmor = Get-VM VMNAME | Get-View
$vmmor.CloneVM_Task($target.MoRef , $name, $vmclonespec )
```

Q: How do I VMotion a VM?

A: This script can be used to VMotion from ESSX1 to ESX2.

```
connect-viserver -Server
<Server> -User <user> -Password <password>
get-vmhost ESX1 | get-vm namevm | move-vm
&ndash;destination (get-vmhost ESX2)
```

## Snapshots:

Q: How do I snapshot a VM? Multiple VMs?

A: New-Snapshot cmdlet can be used to create the snapshot of a virtual machine. This script will create the snapshot with name "Target-Snpsht" for all the virtual machines.

To do snapshot of a particular machine use Get-VM with virtual machine. Below is the script for the same.

```
connect-viserver -server
```

```
<server> -user <user> -password <pwd>
New-Snapshot -Name &ldquo;Target-Snpsht&rdquo; -VM
(Get-VM)
```

Q: How do I set a VM to a particular snapshot?

A: A virtual machine can have multiple snapshots and any of them can be set at a time. This script can set a virtual machine to a specified snapshot.

```
connect-viserver -server
<server> -user <user> -password <pwd>
$vm = Get-VM -Name &ldquo;VMName&rdquo;
$snapshotname = $vm | Get-Snapshot -Name
&ldquo;Target-Snpsht&rdquo;
$vm | get-snapshot -Name
&ldquo;Target-Snpsht&rdquo; | where
{ $_.name -like $snapshotname.name
} | %
{ set-vm $vm -snapshot $snapshotname
}
```

Q: How can I identify all snapshots older than a particular date?

A: This script show snapshots more than 30 days old.

```
connect-viserver -server
<server> -user <user> -password <pwd>
Get-VM | Get-Snapshot | Where
{ $_.Created -lt (Get-Date).AddDays(-30)
} | select Name, Created
```

## ESX Host Configuration:

Q: How can I create virtual switches and portgroups?

A: This script creates the virtual switches.

```
connect-viserver -server
<server> -user <user> -password <pwd>
Get-VMHost YOURHOST | New-VirtualSwitch -name
"My New Switch" -nic YOURNIC (e.g. vmnic1)
Get-VMHost YOURHOST | Get-VirtualSwitch -name
"My New Switch" | New-VirtualPortGroup
-vlanid 100
```

Q: How can I configure NTP?

A: This script can be used to configure the NTP.

```
connect-viserver -server
<server> -user <user> -password <pwd>
$HS = Get-VMHost YOURHOST | get-view
$dtsystem = $HS.ConfigManager.DateTimeSystem
$mor = Get-View -MoRef $dtsystem
```

```
$dateConfig = New-Object VMware.Vim.HostDateTimeConfig
$hsntpConfig = New-Object VMware.Vim.HostNtpConfig
$dateConfig.ntpConfig = $hsntpConfig
$dateConfig.ntpConfig.server = @(
"<Your NTP Server>"
)
$dateConfig.timeZone =
"<Your Timezone>"
$smor.updateDateTimeConfig($dateConfig)
```

Q: How can I rescan Host Bus Adapters (HBAs)?

A: This script used to rescan Host Bus Adapters (HBAs).

```
connect-viserver -server
<server> -user <user> -password <pwd>
get-vmhost | Get-VMHostStorage -RescanAllHBA
```

Q: How can I display and configure the physical adapters of my ESX host.

A: This script display the list of physical adapters of an ESX host.

```
connect-viserver -server
<server> -user <user> -password <pwd>
Get-VMHost | %
{Get-View $_.ID
} | %
{$_ .name; $nics = $_.config.network.pnic;
foreach ($nic in $nics)
{$nic.device
}
}
```

This script can be used to configure the physical adapters of ESX host.

```
connect-viserver -server
<server> -user <user> -password <pwd>

# $nics will hold an array of physical NICs on the server.
$nics = (get-vmhost | get-vmhostnetwork).PhysicalNic

$nics | where
{ $_.DeviceName -eq
"NIC_TO_CONFIGURE"
} | Set-VMHostNetworkAdapter -duplex full
-bitratepersecmb 100
```

Q: How can I power off an ESX host?

A: This script can be used to power off an ESX host.

```
connect-viserver -server
<server> -user <user> -password <pwd>
$HS = Get-VMHost YOURHOST | get-view
$HS.ShutdownHost(
"false")
```

## Reporting:

Q: How can I create a tabular report of all my VMs, including their configurations and amount of disk space allocated?

A: This script can be used for tabular reporting of all VMs with their configuration.

```
connect-viserver -server
<server> -user <user> -password <pwd>
get-vm | %
{ get-view $_.ID
} | Select-Object Name, @
{ Name=
"MemoryMB"; Expression=
{$_ .Summary.Config.memorySizeMB
}
} , @
{ Name=
"MemReservation"; Expression=
{$_ .Summary.Config.memoryReservation
}
} , @
{ Name=
"numCpu"; Expression=
{$_ .Summary.Config.numCpu
}
} , @
{ Name=
"numVirtualDisks"; Expression=
{$_ .Summary.Config.numVirtualDisks
}
}
```

## Other:

Q: How can I create folders that appear under "Hosts and Clusters" in VC (a so-called "Blue Folder")?

A: A "Blue Folder" is a folder created under the VM folder.

For example:

```
connect-viserver -server
<server> -user <user> -password <pwd>
Get-Folder vm | New-Folder -Name
&quot;ldquo;Target-Folder-Name&rdquo;
```

Q: I'm concerned about the security implications of Invoke-VMScript, how can I disable it?

A: Invoke-VMscript can be disabled on a VM-by-VM basis by adding this line to the VM's VMX file: `monitor_control.restrict_backdoor = "true"` . You can add this line by hand or you can add it with the following script:

```
$vmsToDisableInvoke = Get-VM
<vms to reconfigure>
$vmsToDisableInvoke | foreach {
    $vm = Get-View $_.Id
    $vmConfigSpec = New-Object VMware.Vim.VirtualMachineConfigSpec
    $vmConfigSpec.extraconfig += New-Object VMware.Vim.optionvalue

    $vmConfigSpec.extraconfig[0].Key="monitor_control.restrict_backdoor"
    $vmConfigSpec.extraconfig[0].Value="true"
    $vm.ReconfigVM($vmConfigSpec)
}
```