**xHyperV Module – Windows PowerShell Desired State Configuration Resource Kit**

# **Introduction**

The **xHyper-V** module is a part of the Windows PowerShell Desired State Configuration (DSC) Resource Kit, which is a collection of DSC Resources produced by the PowerShell Team. This module contains the **xVhd, xVMHyperV** and **xVMSwitch** resources. These DSC resources allow configuration of Hyper-V host for Vhd, VM and VMSwitch.

**All of the resources in the DSC Resource Kit are provided AS IS, and are not supported through any Microsoft standard support program or service. The “x” in xHyper-V stands for experimental**, which means that these resources will be **fix forward** and monitored by the module owner(s).

Please leave comments, feature requests, and bug reports in the Q & A tab for this module.

If you would like to modify **xHyper-V,** feel free. When modifying, please update the module name, resource friendly name, and MOF class name (instructions below). As specified in the license, you may copy or modify this resource as long as they are used on the Windows Platform.

For more information about Windows PowerShell Desired State Configuration, check out the blog posts on the [PowerShell Blog](http://blogs.msdn.com/b/powershell/) ([this](http://blogs.msdn.com/b/powershell/archive/2013/11/01/configuration-in-a-devops-world-windows-powershell-desired-state-configuration.aspx) is a good starting point). There are also great community resources, such as [PowerShell.org](http://powershell.org/wp/tag/dsc/), or [PowerShell Magazine](http://www.powershellmagazine.com/tag/dsc/). For more information on the DSC Resource Kit, check out [this blog post](http://go.microsoft.com/fwlink/?LinkID=389546).

# Installation

To install **xHyper-V** module

* Unzip the content under $env:ProgramFiles\WindowsPowerShell\Modules folder

**To confirm installation:**

* **Run Get-DSCResource to see that xVhd, xVMHyperV and xVMSwitch are among the DSC Resources listed**

# **Requirements**

This module requires the latest version of PowerShell (v4.0, which ships in Windows 8.1 or Windows Server 2012R2). It also requires **Hyper-V** features. To easily use PowerShell 4.0 on older operating systems, [install WMF 4.0](http://www.microsoft.com/en-us/download/details.aspx?id=40855). Please read the installation instructions that are present on both the download page and the release notes for WMF 4.0.

# **Description**

The **xHyper-V** module contains the **xVhd, xVMHyperV** and **xVMSwitch** DSC Resources. These DSC resources allow you to configure Hyper-V host for its Vhd, VM and VMSwitch.

# Details

**xVhd** resource has following properties:

* **Name**: The desired VHD file name
* **Path**: The desired folder where the VHD will be created
* **ParentPath**: Parent VHD file path, for differencing disk
* **MaximumSizeBytes**: Maximum size of Vhd to be created
* **Generation**: Virtual disk format - Vhd or Vhdx
* **Ensure:** Should the VHD be present or absent

**xVMHyperV** resource has following properties:

* **Name:** The desired VM name
* **VhdPath:** The desired VHD associated with the VM
* **SwitchName:** Virtual switch associated with the VM
* **State:** State of the VM – Running,Paused,Off
* **Path:** Folder where the VM data will be stored;
* **Generation:** Associated Virtual disk format - Vhd or Vhdx
* **StartupMemory:** Startup RAM for the VM
* **MinimumMemory:** Minimum RAM for the VM. This enables dynamic memory
* **MaximumMemory:** Maximum RAM for the VM. This enable dynamic memory
* **MACAddress:** MAC address of the VM
* **ProcessorCount:** Processor count for the VM
* **WaitForIP:** If specified, waits for VM to get valid IP address
* **RestartIfNeeded:** If specified, shutdowns and restarts the VM as needed for property changes
* **Ensure:** Should the VM be present or absent

**xVMSwitch** resource has following properties:

* **Name:** The desired VM Switch name
* **Type:** The desired type of switch – External,Internal,Private
* **NetAdapterName:** Network adapter name for external switch type
* **AllowManagementOS:** Specify if the VM host has access to the physical NIC
* **Ensure:** Should the VM Switch be present or absent

# **Example: Create a new VHD**

This configuration will create a new VHD on Hyper-V host.

configuration Sample\_xVHD\_NewVHD

{

param

(

[Parameter(Mandatory)]

[string]$Name,

[Parameter(Mandatory)]

[string]$Path,

[Parameter(Mandatory)]

[Uint64]$MaximumSizeBytes,

[ValidateSet("Vhd","Vhdx")]

[string]$Generation = "Vhd",

[ValidateSet("Present","Absent")]

[string]$Ensure = "Present"

)

Import-DscResource -module xHyper-V

Node localhost

{

xVHD NewVHD

{

Ensure = $Ensure

Name = $Name

Path = $Path

Generation = $Generation

MaximumSizeBytes = $MaximumSizeBytes

}

}

}

# **Example: Create a differencing VHD**

This configuration will create a differencing VHD, given a parent VHD, on Hyper-V host.

configuration Sample\_xVhd\_DiffVHD

{

param

(

[Parameter(Mandatory)]

[string]$Name,

[Parameter(Mandatory)]

[string]$Path,

[Parameter(Mandatory)]

[string]$ParentPath,

[ValidateSet("Vhd","Vhdx")]

[string]$Generation = "Vhd",

[ValidateSet("Present","Absent")]

[string]$Ensure = "Present"

)

Import-DscResource -module xHyper-V

Node localhost

{

xVHD DiffVHD

{

Ensure = $Ensure

Name = $Name

Path = $Path

ParentPath = $ParentPath

Generation = $Generation

}

}

}

# **Example: Create a VM for a given VHD**

This configuration will create a VM, given a VHD, on Hyper-V host.

configuration Sample\_xVMHyperV\_Simple

{

param

(

[string[]]$NodeName = 'localhost',

[Parameter(Mandatory)]

[string]$VMName,

[Parameter(Mandatory)]

[string]$VhdPath

)

Import-DscResource -module xHyper-V

Node $NodeName

{

# Install HyperV feature, if not installed - Server SKU only

WindowsFeature HyperV

{

Ensure = 'Present'

Name = 'Hyper-V'

}

# Ensures a VM with default settings

xVMHyperV NewVM

{

Ensure = 'Present'

Name = $VMName

VhdPath = $VhdPath

Generation = $VhdPath.Split('.')[-1]

DependsOn = '[WindowsFeature]HyperV'

}

}

}

# **Example: Create a VM with dynamic memory for a given VHD**

This configuration will create a VM with dynamic memory settings, given a VHD, on Hyper-V host.

configuration Sample\_xVMHyperV\_DynamicMemory

{

param

(

[string[]]$NodeName = 'localhost',

[Parameter(Mandatory)]

[string]$VMName,

[Parameter(Mandatory)]

[string]$VhdPath,

[Parameter(Mandatory)]

[Uint64]$StartupMemory,

[Parameter(Mandatory)]

[Uint64]$MinimumMemory,

[Parameter(Mandatory)]

[Uint64]$MaximumMemory

)

Import-DscResource -module xHyper-V

Node $NodeName

{

# Install HyperV feature, if not installed - Server SKU only

WindowsFeature HyperV

{

Ensure = 'Present'

Name = 'Hyper-V'

}

# Ensures a VM with dynamic memory

xVMHyperV NewVM

{

Ensure = 'Present'

Name = $VMName

VhdPath = $VhdPath

Generation = $VhdPath.Split('.')[-1]

StartupMemory = $StartupMemory

MinimumMemory = $MinimumMemory

MaximumMemory = $MaximumMemory

DependsOn = '[WindowsFeature]HyperV'

}

}

}

# **Example: Create a VM with dynamic memory, network interface and processor count for a given VHD**

This configuration will create a VM with dynamic memory, network interface and processor count settings, given a VHD, on Hyper-V host.

configuration Sample\_xVMHyperV\_Complete

{

param

(

[string[]]$NodeName = 'localhost',

[Parameter(Mandatory)]

[string]$VMName,

[Parameter(Mandatory)]

[string]$VhdPath,

[Parameter(Mandatory)]

[Uint64]$StartupMemory,

[Parameter(Mandatory)]

[Uint64]$MinimumMemory,

[Parameter(Mandatory)]

[Uint64]$MaximumMemory,

[Parameter(Mandatory)]

[String]$SwitchName,

[Parameter(Mandatory)]

[String]$Path,

[Parameter(Mandatory)]

[Uint32]$ProcessorCount,

[ValidateSet('Off','Paused','Running')]

[String]$State = 'Off',

[Switch]$WaitForIP

)

Import-DscResource -module xHyper-V

Node $NodeName

{

# Install HyperV feature, if not installed - Server SKU only

WindowsFeature HyperV

{

Ensure = 'Present'

Name = 'Hyper-V'

}

# Ensures a VM with all the properties

xVMHyperV NewVM

{

Ensure = 'Present'

Name = $VMName

VhdPath = $VhdPath

SwitchName = $SwitchName

State = $State

Path = $Path

Generation = $VhdPath.Split('.')[-1]

StartupMemory = $StartupMemory

MinimumMemory = $MinimumMemory

MaximumMemory = $MaximumMemory

ProcessorCount = $ProcessorCount

MACAddress = $MACAddress

RestartIfNeeded = $true

WaitForIP = $WaitForIP

DependsOn = '[WindowsFeature]HyperV'

}

}

}

# **Example: Create an internal VM Switch**

This configuration will create an internal VM Switch, on Hyper-V host.

configuration Sample\_xVMSwitch\_Internal

{

param

(

[string[]]$NodeName = 'localhost',

[Parameter(Mandatory)]

[string]$SwitchName

)

Import-DscResource -module xHyper-V

Node $NodeName

{

# Install HyperV feature, if not installed - Server SKU only

WindowsFeature HyperV

{

Ensure = 'Present'

Name = 'Hyper-V'

}

# Ensures a VM with default settings

xVMSwitch InternalSwitch

{

Ensure = 'Present'

Name = $SwitchName

Type = 'Internal'

DependsOn = '[WindowsFeature]HyperV'

}

}

}

# **Example: Create an external VM Switch**

This configuration will create an external VM Switch, on Hyper-V host.

configuration Sample\_xVMSwitch\_External

{

param

(

[string[]]$NodeName = 'localhost',

[Parameter(Mandatory)]

[string]$SwitchName,

[Parameter(Mandatory)]

[string]$NetAdapterName

)

Import-DscResource -module xHyper-V

Node $NodeName

{

# Install HyperV feature, if not installed - Server SKU only

WindowsFeature HyperV

{

Ensure = 'Present'

Name = 'Hyper-V'

}

# Ensures a VM with default settings

xVMSwitch ExternalSwitch

{

Ensure = 'Present'

Name = $SwitchName

Type = 'External'

NetAdapterName = $NetAdapterName

DependsOn = '[WindowsFeature]HyperV'

}

}

}

# Renaming Requirements

1. Update the following names by replacing MSFT with your company/community name and replace the “x” with your own prefix (e.g. the resource name should change from MSFT\_xComputer to Contoso\_myComputer):

* **Module name**
* **Resource Name**
* **Resource Friendly Name**
* **MOF class name**
* **Filename for the <resource>.schema.mof**

1. Update module and metadata information in the module manifest
2. Update any configuration that use these resources

# Versions

1.0.0.0

* Initial release with the following resources
  + xVhd
  + xVMHyperV
  + xVMSwitch