# HP Operations Orchestration

for the Windows operating system

Software Version: 9.00.06

## Microsoft System Center Virtual Machine Manager Integration Guide

Document Release Date: October 2011

Software Release Date: October 2011

# Legal Notices

## Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

## Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

## Copyright Notices



## Trademark Notices

For information on open-source and third-party software acknowledgements, see *Open-Source and Third-Party Software Acknowledgements* (HPOO_OpenSrc_3rd-PartyAcks.pdf) in the documentation set for this release.

## Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

**http:\\h20230.www2.hp.com\selfsolve\manuals**

This site requires that you register for an HP Passport and sign-in. To register for an HP Passport ID, go to:

**http:\\h20229.www2.hp.com\passport-registration.html**

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

# Support

Visit the HP Software Support Web site at:

**www.hp.com\go\hpsoftwaresupport**

This Web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support Web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

**http:\\h20229.www2.hp.com\passport-registration.html**

To find more information about access levels, go to:

**http:\\h20230.www2.hp.com\new_access_levels.jsp**

# Contents

# 1 Introduction

This section includes the following topics:

# Purpose of the Microsoft System Center Virtual Machine Manager (SCVMM) Integration

SCVMM helps you enable centralized management of physical and virtual IT infrastructure and dynamic resource optimization across multiple virtualization platforms such as Hyper-V. This includes planning, deploying, managing, and optimizing the virtual infrastructure.

The key benefits of SCVMM are that it:

- Centrally creates and manages virtual machines across the entire datacenter.
- Easily consolidates multiple physical servers onto virtual hosts.
- Rapidly provisions and optimizes new and existing virtual machines.

With this integration, administrators can create HP Operations Orchestration (OO) flows that are integrated with SCVMM. Performance and Resource Optimization (PRO) enables the dynamic management of virtual resources through its integration with Operation Manager.

To learn how to create OO flows, see the *Studio Guide to Authoring Operations Orchestration Flows* (Studio_AuthorsGuide.pdf) in the documentation set for the current OO release,

The SCVMM integration uses Windows PowerShell cmdlets to integrate with OO.

# Audience

This guide is intended for system administrators who establish and maintain the implementation of the integration between SCVMM and HP OO. This guide assumes that you have administrative access to both systems.

# Prerequisites

To use this integration successfully, you should have administrator-level knowledge of the SCVMM and the Windows PowerShell cmdlets used by the integration. See *SCVMM PowerShell cmdlets* for more information.

# Supported Versions

**Table 1 Supported Versions**

| Operations Orchestration Version | SCVMM Version |
|---|---|
| 9.00.06 | 2008 R2 |

# Downloading OO Releases and Documents on HP Live Network

HP Live Network provides an **Operations Orchestration Community** page where you can find and download supported releases of OO and associated documents.

To download OO releases and documents, visit the following site:

**https:\\www.www2.hp.com**

This site requires that you register for an HP Passport and sign-in. To register for an HP Passport ID, go to:

**http:\\h20229.www2.hp.com\passport-registration.html**

Or click the **New users - please register** link on the HP Passport login page.

On the **HP Live Network** page, click **Operations Orchestration Community**.

**The Operations Orchestration Community** page contains links to announcements, discussions, downloads, documentation, help, and support.

# Related Documents

This section provides a list of documents that are related to this document and that may be useful to consult along with it. Use the following links to read the following documents

- Microsoft TechNet document *System Center Virtual Machine Manager 2008 and Virtual Machine Manager 2008 R2.*
- *Microsoft Virtual Machine Manager 2008 R2 Deployment Guide*
- *Microsoft Virtual Machine Manager 2008 R2 Guide to Operations Manager Integration*.
- *Microsoft Virtual Machine Manager 2008 R2 Security Guide*
- *Microsoft Virtual Machine Manager 2008 R2 Operations Guide*
- *Microsoft Virtual Machine Manager 2008 R2 Scripting Guide*
- *Microsoft System Center Virtual Machine Manager 2008 R2 Cmdlet Reference*
- *Microsoft System Center Virtual Machine Manager Authoring PRO-Enabled Management Packs*

# 2 Getting Started with the SCVMM Integration

This section includes the following topics:

# Installing and Configuring the Integration

The following are some PowerShell requirements and useful notes for configuring the integration:

1. PowerShell 2.0 with WinRM 2.0 must be installed on your RAS machine. You can download the kit from **http:\\support.microsoft.com\kb\968930**.
2. Before using the SCVMM integration operations, make sure you have enabled PowerShell remoting, by using the **Enable-PSRemoting** cmdlet.
3. In workgroup environments, you may need to enable classic mode authentication for network logons. To do this, open the **Local Security Policy** from the Control Panel and select **Administrative Tools**. Navigate to **Local Policies -> Security Options**, double-click **Network Access: Sharing and Security Model for local accounts** and set it to **classic**.
4. Modify the **WSMan trusted hosts** setting by adding the IP addresses of all remoting clients to the list of trusted hosts. Do this by using one of the following commands:
   a. `Set-item wsman:localhost\client\trustedhosts -value *` **(adds all computers as trusted hosts).**
   b. Set-item wsman:localhost\client\trustedhosts -value Computer (only adds Computer to the trusted hosts).
   c. Set-item wsman:localhost\client\trustedhosts -value *.domain.com (adds all computers in the specified domain).
   d. Set-item wsman:localhost\client\trustedhosts -value 10.10.10.1 (adds the remote computer with the IP address 10.10.10.1 to the trusted hosts list).
5. To enable CredSSP authentication, follow the steps described in *About SCVMM Security*.
6. To use Kerberos authentication, the client and server machine must be in the same domain.

## Hyper-V Management Through SCVMM

- Managed Hyper-V servers must be members of the same domain as the SCVMM server that **manages** them.
- SCVMM provides a Hyper-V agent that must be installed on managed Hyper-V servers.
- Add Hyper-V servers into SCVMM using one of the following methods:
  - From the SCVMM Administrator console, as a host on an Active Directory domain or on a perimeter network
  - From PowerShell, using the **Add-VMHost cmdlet**

# SCVMM — OO Integration Architecture

System Center Virtual Machine Manager allows you to administer a set of virtual machine hosts using:

- Windows PowerShell
- SCVMM Administrator console
- Web-based Delegated Provisioning UI

The SCVMM integration uses the capabilities of PowerShell to remotely execute SCVMM-specific cmdlets.

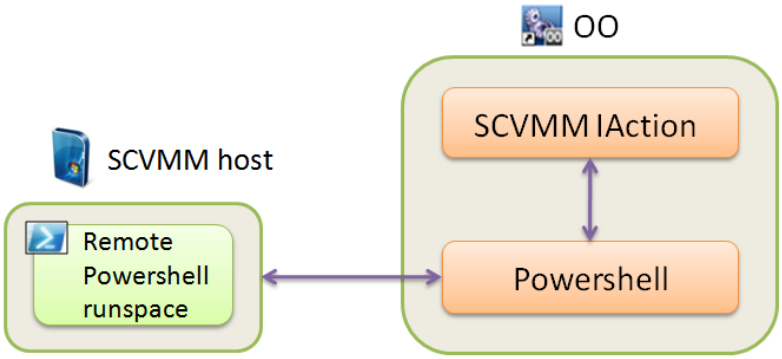


**Figure 1 - SCVMM - OO Integration Architecture**

The SCVMM integration contains a generic SCVMM operation named **Run SCVMM Commands**, and a set of flows that use this operation to execute specific tasks.

The **Run SCVMM Commands** operation:

1. Creates a remote session to the target host using the **New-PSSession** command.
2. Imports the **Microsoft.SystemCenter.VirtualMachineManager** snap-in using the **Add-PSSnapin** command.
3. Sequentially executes the specified commands.
4. Serializes the result as a table.
5. Closes the remote session.

Most of the SCVMM cmdlets require that you run the **Get-VMMServer** command before you run them. The **Get-VMMServer** command connects to a Virtual Machine Manager server and retrieves the object that represents the server from the Virtual Machine Manager database.

Because of this, the generic **Run SCVMM Commands** operation runs the following command before any other commands you specify:

```
get-vmmserver -computerName "10.10.10.100"
```

The commands you specify in the **Run SCVMM Commands** operation are executed sequentially, not in a pipeline, so you must enter each command on a separate row. The operation initializes the remote session, runs the commands one by one, and then closes the session. The result of the last command is returned.

The generic operation is used to create flows that execute specific tasks. A flow contains the set of common inputs described in *Common Inputs in the Integration* and a number of specific inputs. The value of these inputs is used to build the PowerShell commands that are executed on the SCVMM host using the **Run SCVMM Commands** operation.

# SCVMM Use Cases

Following are the major use cases for the SCVMM integration, and the operations and flows that implement them.

1. As an SCVMM administrator I want a generic OO operation or flow so that I will be able to **execute any SCVMM PowerShell cmdlets** on a remote SCVMM server:
   - Run SCVMM Commands
2. As an SCVMM administrator, I want a set of OO operations and flows so that I will be able to **manage and provision virtual machines** on Hyper-V servers:
   - Disable Virtual Machine Undo Disk
   - Discard Virtual Machine State
   - Get Virtual Machine
   - Move Virtual Machine
   - New Virtual Machine
   - Refresh Virtual Machine
   - Register Virtual Machine
   - Remove Virtual Machine
   - Repair Virtual Machine
   - Resume Virtual Machine
   - Save Virtual Machine State
   - Set Virtual Machine
   - Shut Down Virtual Machine
   - Start Virtual Machine
   - Stop Virtual Machine
   - Store Virtual Machine
   - Suspend Virtual Machine
3. As an SCVMM administrator, I want a set of OO operations and flows so that I will be able to **deploy virtual machines from templates**:
   - Get Template
   - New Template
   - Remove Template
   - Set Template
4. As an SCVMM administrator, I want a set of OO operations and flows so that I will be able to **manage and organize virtual disk drives**:
   - Compress Virtual Disk Drive
   - Convert Virtual Disk Drive
   - Expand Virtual Disk Drive
   - Get Virtual Disk Drive

- New Virtual Disk Drive
- Remove Virtual Disk Drive
- Set Virtual Disk Drive

5 As an SCVMM administrator, I want a set of OO operations and flows so that I will be able to **manage and organize virtual DVD drives**:

- Get Virtual DVD Drive
- New Virtual DVD Drive
- Remove Virtual DVD Drive
- Set Virtual DVD Drive

6 As an SCVMM administrator, I want a set of OO operations and flows so that I will be able to **manage and organize virtual hard disks**:

- Get Virtual Hard Disk
- Move Virtual Hard Disk
- Remove Virtual Hard Disk
- Set Virtual Hard Disk

7 As an SCVMM administrator, I want a set of OO operations and flows so that I will be able to **manage and organize virtual networks**:

- Get Virtual Network
- New Virtual Network
- Remove Virtual Network
- Set Virtual Network

8 As an SCVMM administrator, I want a set of OO operations and flows so that I will be able to **manage and organize virtual SCSI adapters**:

- Get Virtual SCSI Adapters
- New Virtual SCSI Adapter
- Remove Virtual SCSI Adapter

9 As an SCVMM administrator, I want a set of OO operations and flows so that I will be able to **manage and organize virtual network adapters**:

- Get Virtual Network Adapters
- New Virtual Network Adapters
- Remove Virtual Network Adapters
- Set Virtual Network Adapters

10 As an SCVMM administrator, I want a set of OO operations and flows so that I will be able to **provision hardware profiles**:

- Get CPU Type
- Get Hardware Profile
- Get Operating System
- New Hardware Profile
- Remove Hardware Profile

- Set Hardware Profile

11 As an SCVMM administrator, I want a set of OO operations and flows so that I will be able to **manage jobs**:

- Get Job
- Get Step
- Restart Job
- Stop Job

# 3 Using the SCVMM – OO Integration

This section includes the following topics:

# Location of SCVMM Integration Operations and Flows in OO Studio

The Microsoft SCVMM integration includes the following operations and flows in the OO Studio Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.

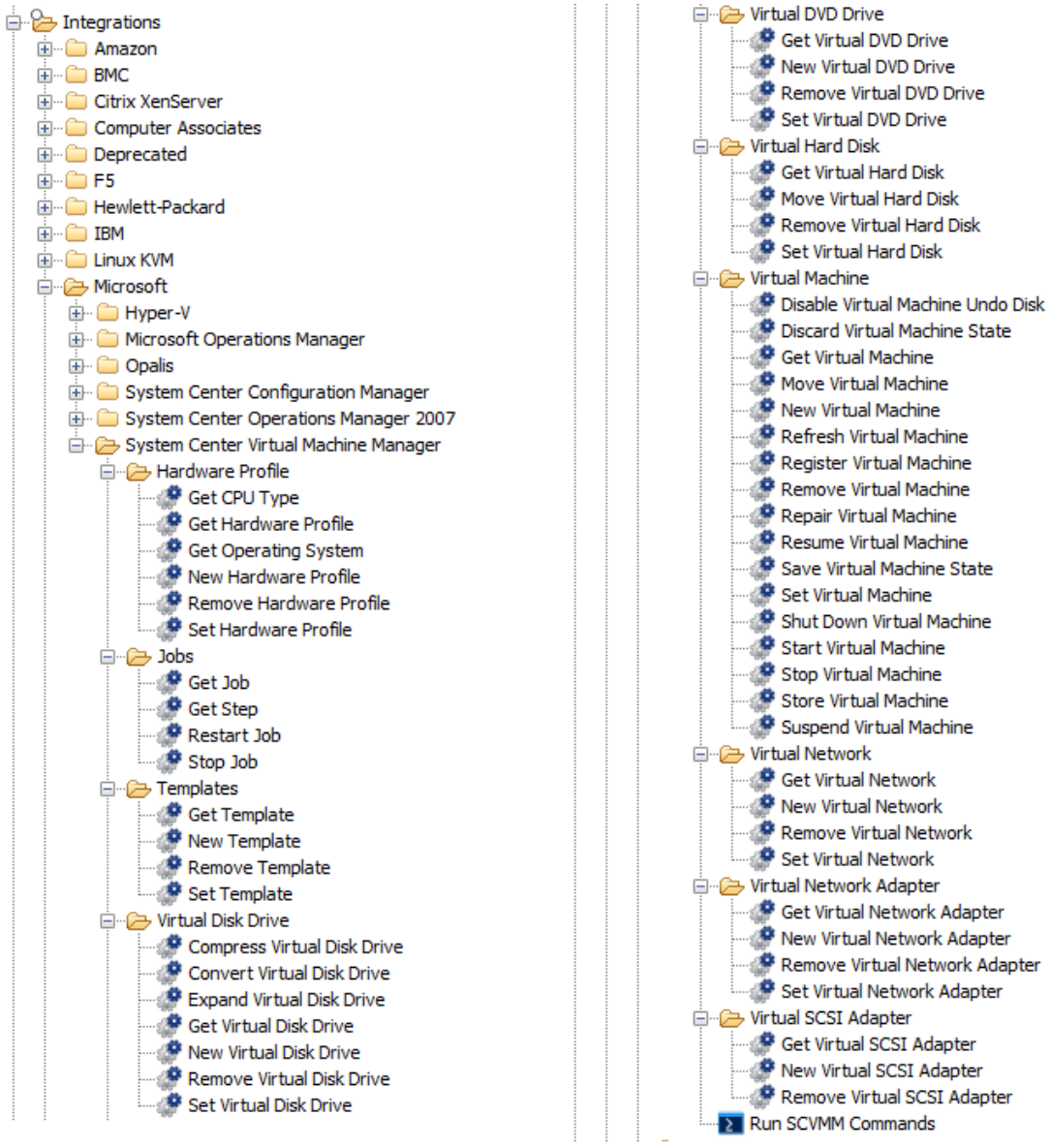


**Figure 2 - Location of the SCVMM Operations and Flows**

# Common Inputs in the Integration

OO operations and flows use inputs to specify how they obtain the data that they need and when the data is obtained. The following inputs are used in most of the SCVMM integration's operation and flows.

### host

The hostname or IP address of the SCVMM server.

### username

The username to use when connecting to the server. Use the format **`username@domain`** or **`domain\username`**.

### password

The password to use when connecting to the server.

### authType

Specifies the mechanism used to authenticate the user's credentials. The valid values (which are case-insensitive) are **Default**, **Negotiate**, **Credssp**, and **Kerberos**. The default value is **Default**.

### delimiter

The delimiter used to separate each property name from the property value in the output table. The default value is a colon (:).

### colDelimiter

The delimiter used to separate the columns in the output table. The default value is a comma (,).

### rowDelimiter

The delimiter used to separate the rows in the output table. The default value is the newline character.

# Descriptions of SCVMM Integration Operations and Flows

This section describes the SCVMM integration's operation and flows, including their inputs, results, responses, examples, and other important information.

## Run SCVMM Commands

The **Run SCVMM Commands** operation runs System Center Virtual Machine Manager (SCVMM) cmdlets using Windows PowerShell. The complete syntax and descriptions of all cmdlets that can be used with this operation can be found at **http:\\www.microsoft.com\download\en\details.aspx?displaylang=en&id=20419**. The operation creates a remote PowerShell session to the target host, loads the **Microsoft.SystemCenter.VirtualMachineManager** snap-in and executes the specified commands.

## Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

### commands

The PowerShell cmdlets to execute.

## Results

The operation returns the following results:

### returnResult

A table containing a row for each PSObject that the cmdlets emit. The table's columns represent the properties of these PSObjects, in the format `propertyName<delimiter>propertyValue`.

### warnings

A list of warnings produced by the PowerShell commands. If the commands were executed without warnings, this result is empty.

## Responses

The operation returns one of the following responses:

### Success

The cmdlets were executed successfully.

### Failure

The cmdlets could not be executed.

**Notes:**

- PowerShell 2.0 with WinRM 2.0 must be installed on your client. You can download the kit from *http:\\support.microsoft.com\kb\968930*.
- Before using this operation, make sure you have enabled PowerShell remoting, by using the **Enable-PSRemoting** cmdlet.
- The Windows Remote Management service must be running.
- In workgroup environments, you may need to enable classic mode authentication for network logons. To do this, open the **Local Security Policy** from the Control Panel and selecting **Administrative Tools**. Navigate to **Local Policies** -> **Security Options**, double-click **Network Access: Sharing and Security Model for local accounts** and set it to **classic**.
- Modify the WSMan trusted hosts setting by adding the IP addresses of all remoting clients to the list of trusted hosts. You can do this using one of the following commands:
  - Set-item wsman:localhost\client\trustedhosts -value * (adds all computers as trusted hosts).
  - Set-item wsman:localhost\client\trustedhosts -value Computer (only adds Computer to the trusted hosts).
  - Set-item wsman:localhost\client\trustedhosts -value *.domain.com (adds all

computers in the specified domain).
  - Set-item wsman:localhost\client\trustedhosts -value 10.10.10.1 (adds the remote computer with the IP address 10.10.10.1 to the trusted hosts list).
- Read the SCVMM\ folder's description for information on how to enable CredSSP authentication.

# Hardware Profile

## Get CPU Type

The **Get CPU Type** flow runs the **Get-CPUType** PowerShell command that gets a list of available CPU types that can be used to define the processor in a virtual machine, template or hardware profile.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### id

The identifier of the CPU type whose details will be returned. The valid format for the identifier is **`xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx`**. For example, **h7js87df-637h-dg7s-0jd7-jd745gsb65**.

### Results

The flow returns the following results:

#### returnResult

A table containing a row for each CPU type retrieved or an exception message if the flow failed. The table's columns represent the properties of the CPU type, in the format `propertyName<delimiter>propertyValue`. If there are no CPU types that match the given criteria the result will be empty.

#### warnings

A list of warnings produced by the PowerShell commands. If the commands were executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The flow executed successfully and the result contains a list of available CPU types.

#### Failure

The flow could not retrieve the list of CPU types.

**Note:** PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.

## Get Hardware Profile

The **Get Hardware Profile** flow runs the **Get-HardwareProfile** PowerShell command that gets details about the hardware profiles available in the Virtual Machine Manager library.

### Inputs

All of the *flow's inputs except the following are described in Common Inputs in the Integration*.

#### id

The identifier of the hardware profile whose details will be returned. The valid format for the identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **h7js87df-637h-dg7s-0jd7-jd745gsb65**.

#### all

Set to **true** to retrieve a list of all hardware profiles regardless of their parent object. If you do not specify a value, the flag is not set.

### Results

The flow returns the following results:

#### returnResult

A table containing a row for each hardware profile retrieved or an exception message if the flow failed. The table's columns represent the properties of the hardware profile, in the format `propertyName<delimiter>propertyValue`. If there are no profiles that match the given criteria, the result is empty.

#### warnings

A list of warnings produced by the PowerShell commands. If the commands were executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The flow executed successfully and the result contains the list of hardware profiles that matched the given criteria.

#### Failure

The flow could not retrieve the list of hardware profiles.

**Notes:**

- Not all of the available filters can be applied at the same time. Use the **Get-Help Get-HardwareProfile -Detailed** PowerShell command to see the valid filter combinations and a set of examples.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder and the **Run SCVMM Commands** operation's **Description** tab.

### Get Operating System

The **Get Operating System** flow runs the **Get-OperatingSystem** PowerShell command that gets a list of valid operating systems from the Virtual Machine Manager database.

#### Inputs

The flow has only the inputs described in *Common Inputs in the Integration*.

#### Results

The flow returns the following results:

##### returnResult

A table containing a row for each operating system retrieved or an exception message if the flow failed. The table's columns represent the properties of the operating system, in the format `propertyName<delimiter>propertyValue`.

##### warnings

A list of warnings produced by the PowerShell commands. If the commands were executed without warnings, this result is empty.

#### Responses

The flow returns one of the following responses:

##### Success

The flow executed successfully and the result contains a list of valid operating systems.

##### Failure

The flow could not retrieve the list of operating systems.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Get-OperatingSystem -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.

## New Hardware Profile

The **New Hardware Profile** flow runs the **New-HardwareProfile** PowerShell cmdlet that creates a hardware profile in the Virtual Machine Manager library. You can create a hardware profile based on defaults or you can customize an existing hardware profile.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### name

The name of the new hardware profile.

#### hardwareProfile

The name or identifier of the hardware profile to be cloned. The name you specify must match the **Name** property of the hardware profile object exactly. The valid format for the hardware profile identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **MyHardwareProfile** or **h7js87df-637h-dg7s-0jd7-jd745gsb65**.

#### description

A description of the new hardware profile.

#### owner

The owner of the hardware profile in the form of a valid domain user account **domain\username**).

#### bootOrder

Specifies the order of devices that a virtual machine uses to start up. The valid values are **CD**, **IDEHardDrive**, **PXEBoot** and **Floppy**. For example, **PXEBoot,IDEHardDrive,CD,Floppy**.

#### numLock

Set to **true** to enable the BIOS value for NumLock on a virtual machine on a Hyper-V host. The valid values are **true** and **false**. If you specify an invalid value, the property is not set.

#### cpuType

The type of CPU. To retrieve a list with all the CPU types available use the **Get CPU Type** flow. For example, **3.40 GHz Pentium D (dual core)**, **2.40 GHz Opteron**, or **3.07 GHz Xeon**.

#### cpuCount

The number of CPUs to set. The maximum number of processors you can set on a virtual machine depends on the type of virtual machine host.

#### limitCpuForMigration

Specifies whether to limit the processor features to enable migration to a physical computer that has a different version of the same processor as the source computer. The valid values are **true** and **false**. If you specify any other value, the property is not set.

#### limitCpuFunctionality

Specifies whether to enable running an older operating system (such as Windows NT 4.0) by providing only limited CPU functionality for the virtual machine. The valid values are **true** and **false**. If you specify any other value, the property is not set.

#### memoryMb

The total amount of memory in megabytes assigned to the virtual machine. The maximum host memory assignable to the virtual machine depends on the type of virtual machine host.

### Results

The flow returns the following results:

#### returnResult

A table containing a row with details about the new hardware profile or an exception message if the flow failed. The table's columns represent the properties of the hardware profile, in the format `propertyName<delimiter>propertyValue`.

#### warnings

A list of warnings produced by the PowerShell commands. If the commands were executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The operation executed successfully and the result contains details about the new hardware profile.

#### Failure

The flow could not create the hardware profile.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help New-HardwareProfile -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder and the **Run SCVMM Commands** operation's **Description** tab.

## Remove Hardware Profile

The **Remove Hardware Profile** flow runs the **Remove-HardwareProfile** PowerShell command that removes a hardware profile from the Virtual Machine Manager library.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### hardwareProfile

The name or identifier of the hardware profile to be removed. The name you specify must match the **Name** property of the hardware profile object exactly. The valid format for the hardware profile identifier is **`xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx`**. For example, **MyHardwareProfile** or **h7js87df-637h-dg7s-0jd7-jd745gsb65**.

### Results

The flow returns the following results:

#### returnResult

A table containing a row with details about the removed hardware profile or an exception message if the flow failed. The table's columns represent the properties of the hardware profile, in the format `propertyName<delimiter>propertyValue`.

#### warnings

A list of warnings produced by the PowerShell commands. If the commands were executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The flow executed successfully and the result contains details about the removed hardware profile.

#### Failure

The flow could not remove the hardware profile.

**Notes:**

- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.

## Set Hardware Profile

The **Set Hardware Profile** flow runs the **Set-HardwareProfile** PowerShell cmdlet that changes the properties of a hardware profile in the Virtual Machine Manager library.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

### hardwareProfile

The name or identifier of the hardware profile to be updated. The name you specify must match the **Name** property of the hardware profile object exactly. The valid format for the hardware profile identifier is **`xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx`**. For example, **MyHardwareProfile** or **h7js87df-637h-dg7s-0jd7-jd745gsb65**.

### name

The new name of the new hardware profile.

### description

A description of the hardware profile.

### owner

The owner of the hardware profile in the form of a valid domain user account (**`domain\username`**).

### bootOrder

Specifies the order of devices that a virtual machine uses to start up. The valid values are **CD**, **IDEHardDrive**, **PXEBoot** and **Floppy**. For example, **PXEBoot,IDEHardDrive,CD,Floppy**.

### numLock

Set to **true** to enable the BIOS value for NumLock for a virtual machine on a Hyper-V host. The valid values are **true** and **false**. If you specify an invalid value, the property is not set.

### cpuType

The type of CPU. To retrieve a list with all the CPU types available, use the **Get CPU Type** flow. For example, **3.40 GHz Pentium D (dual core)**, **2.40 GHz Opteron**, or **3.07 GHz Xeon**.

### cpuCount

The number of CPUs to set. The maximum number of processors you can set on a virtual machine depends on the type of virtual machine host.

### limitCpuForMigration

Specifies whether to limit the processor features to enable migration to a physical computer that has a different version of the same processor as the source computer. The valid values are **true** and **false**. If you specify any other value, the property is not set.

### limitCpuFunctionality

Specifies whether to enable running an older operating system (such as Windows NT 4.0) by providing only limited CPU functionality for the virtual machine. The valid values are **true** and **false**. If you specify any other value, the property is not set.

### memoryMb

The total amount of memory in megabytes assigned to the virtual machine. The maximum host memory assignable to the virtual machine depends on the type of virtual machine host.

### Results

The flow returns the following results:

#### returnResult

A table containing a row with details about the updated hardware profile or an exception message if the flow failed. The table's columns represent the properties of the hardware profile, in the format `propertyName<delimiter>propertyValue`.

#### warnings

A list of warnings produced by the PowerShell commands. If the commands were executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The operation executed successfully and the result contains details about the updated hardware profile.

#### Failure

The flow could not update the hardware profile.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Set-HardwareProfile -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder and the **Run SCVMM Commands** operation's **Description** tab.

## Jobs

### Get Job

The **Get Job** flow executes the **Microsoft.SystemCenter.VirtualMachineManager\Get-Job** PowerShell cmdlet that gets job objects on the Virtual Machine Manager server.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### id

The ID of the job object to retrieve.

#### name

The name of the job object to retrieve.

#### full

Specifies whether to return the job object with an audit record.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the job object that was retrieved.

#### warnings

A list of warnings produced by the cmdlet. If the cmdlet was executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The flow did not complete successfully.

Notes:

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Microsoft.SystemCenter.VirtualMachineManager\Get-Job -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs in the inputs that accept both names and IDs.

## Get Step

The **Get Step** flow executes the **Get-Step** PowerShell cmdlet that gets the steps for the specified job on a Virtual Machine Manager server.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### job

The ID or name of the job for which you want to get the steps.

#### name

The name of the step.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the step object that was retrieved.

#### warnings

A list of warnings produced by the cmdlet. If the cmdlet was executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The flow did not complete successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help  Get-Step -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs in the inputs that accept both names and IDs.

## Restart Job

The **Restart Job** flow executes the **Restart-Job** PowerShell cmdlet that restarts a failed or canceled Virtual Machine Manager job.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### job

The ID or name of the job you want to restart.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the job object that was restarted.

#### warnings

A list of warnings produced by the cmdlet. If the cmdlet was executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The flow did not complete successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help  Restart-Job -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs in the inputs that accept both names and IDs.

## Stop Job

The **Stop Job** flow executes the **Stop-Job** PowerShell cmdlet that stops a Virtual Machine Manager job that is running.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### job

The ID or name of the job you want to stop.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the job object that was stopped.

#### warnings

A list of warnings produced by the cmdlet. If the cmdlet was executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The flow did not complete successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help  Stop-Job -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs in the inputs that accept both names and IDs.

## Templates

### Get Template

The **Get Template** flow executes the **Get-Template** PowerShell cmdlet that retrieves the details of virtual machine template objects from the Virtual Machine Manager library.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### template

The ID or name of the template object to be retrieved. If you do not specify a value for this input, all available template objects in the VMM library are retrieved.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the template object that was retrieved.

#### warnings

A list of warnings produced by the cmdlet. If the cmdlet was executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The flow did not complete successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Get-Template -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs in the inputs that accept both names and IDs.

## New Template

The **New Template** flow executes the **New-Template** PowerShell cmdlet that creates a new template object in the Virtual Machine Manager library.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### newTemplateName

The name of the template to create.

#### libraryServer

The name or ID of a VMM library server object (for example, **SCVMM.battleground.ad**).

#### sharePath

A path to a valid library share on an existing library server that uses a Universal Naming Convention (UNC) path (for example, **FileServer01****LibShare**).

#### hardwareProfile

The name of a hardware profile object.

#### operatingSystem

The name of an operating system object (for example, **Windows 7** or **Suse Linux Enterprise Server 10 (64 bit)**).

### virtualHardDisk

The name or ID of a virtual hard disk object.

### template

The name or ID of a template object.

### vm

The name or ID of a virtual machine object.

### cpuType

The name or ID of a CPU type object (for example, **3.20 GHz Xeon** or **2.40 GHz Pentium 4**).

### cpuCount

The number of CPUs on the new template.

### description

The description of the new template.

### owner

The owner of the new template in the form of a valid domain user account. The valid formats are **username@domain** or **domain\username**.

### orgName

The name of the organization whose member is specified in the **fullName** input.

### fullName

The name of the person whose name in which the template is registered.

### computerName

The name of a computer that VMM can uniquely identify on the network. The valid formats are **FQDN**, **IPv4** or **IPv6** address, or the **NetBIOS** name.

### highlyAvailable

Specifies whether the virtual machine created from this template will be placed on a Hyper-V host that is part of a cluster. The valid values are **true** and **false**.

### memoryMb

The total amount of memory in MB on the host that is assigned to a virtual machine created from this template.

### timeZone

A number/index that identifies a geographical region that shares the same standard time. For a list of time zone indexes, see *Microsoft Time Zone Index Values* at *http://go.microsoft.com/fwlink/?LinkId=120935*.

#### noCustomization

Specifies whether the guest operating system settings on this template cannot be customized. The valid values are **true** and **false**.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the template object that was created.

#### warnings

A list of warnings produced by the cmdlet. If the cmdlet was executed without warnings, his result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The flow did not complete successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help New-Template -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## Remove Template

The **Remove Template** flow executes the **Remove-Template** PowerShell cmdlet that removes a template object from Virtual Machine Manager and deletes all the files associated with it.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### template

The ID or name of the template object to be removed. To avoid unwanted behavior caused by duplicate template names, you should retrieve the template object by its ID (for example, **template-21** or **47aff0a3-dc75-41e4-a598-fbee8f81b719**).

#### force

Specifies whether to force the removal of the template object and of all its associations with other objects. The valid values are **true** and **false**.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the template object that was deleted.

#### warnings

A list of warnings produced by the cmdlet. If the cmdlet was executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The flow did not complete successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Remove-Template -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## Set Template

The **Set Template** flow executes the **Set-Template** PowerShell cmdlet that changes the properties of a template object used in a Virtual Machine Manager environment.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### template

The name or ID of the template object to update.

### operatingSystem

The name of an operating system object (for example, **Windows 7** or **Suse Linux Enterprise Server 10 (64 bit)**).

### name

The new name of the template.

### cpuType

The name or ID of a CPU type object (for example, **3.20 GHz Xeon** or **2.40 GHz Pentium 4**).

### cpuCount

The number of CPUs on the template.

### cpuMax

A numeric value that specifies the highest percentage of the total resources of a single CPU on the host that can be used by a specific virtual machine (created from this template) at any given time.

### cpuReserve

A numeric value that specifies the minimum percentage of the resources of a single CPU on the host to allocate to a virtual machine (created from this template).

### description

The new description of the template.

### owner

The new owner of the template in the form of a valid domain user account. The valid formats are **`username@domain`** or **`domain\username`**.

### orgName

The name of the organization whose member is specified in the **fullName** input.

### fullName

The name of the person in whose name the template is registered.

### computerName

The name of a computer that VMM can uniquely identify on the network. The valid formats are **FQDN**, **IPv4** or **IPv6** address, or the **NetBIOS** name.

### highlyAvailable

Specifies whether the virtual machine created from this template will be placed on a Hyper-V host that is part of a cluster. The valid values are **true** and **false**.

### memoryMb

The total amount of memory in MB on the host that is assigned to a virtual machine created from this template.

### timeZone

A number/index that identifies a geographical region that shares the same standard time. For a list of time zone indexes, see *Microsoft Time Zone Index Values* at *http://go.microsoft.com/fwlink/?LinkId=120935*.

### enabled

Specifies whether users are prevented from using the template object. The valid values are **true** and **false**.

### tag

A word or phrase associated to the template.

### networkUtilization

The amount of bandwidth specified in MB on the host's network that can be used by a specific virtual machine (created from this template).

### limitCpuFunctionality

Specifies whether an older operating system (such as Windows NT 4.0) can be run on a virtual machine created from this template deployed on a Hyper-V host by providing only limited CPU functionality for the virtual machine. The valid values are **true** and **false**.

### limitCpuForMigration

Specifies whether to limit processor features for a virtual machine created from this template to enable migration to a physical computer that has a different version of the same processor as the source computer. The valid values are **true** and **false**.

### quotaPoint

A numeric value that specifies a quota that limits the number of virtual machines self-service users can create.

### numLock

Specifies whether to enable the BIOS value for NumLock on a virtual machine created from this template. The valid values are **true** and **false**.

### bootOrder

An enumeration that specifies the order of devices that a virtual machine (created from this template) on a Hyper-V host uses to start up. This input must contain all of the following values: **CD**, **IDEHardDrive**, **PXEBoot**, or **Floppy**. You can specify them in any order you want. For example, **PXEBoot, IDEHardDrive, CD, Floppy** or **CD, IDEHardDrive, Floppy, PXEBoot**.

## Results

The flow returns the following results:

### returnResult

A table containing the details of the template object that was updated.

#### warnings

A list of warnings produced by the cmdlet. If the cmdlet was executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The flow did not complete successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Set-Template -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## Virtual Disk Drive

### Compress Virtual Disk Drive

The **Compress Virtual Disk Drive** flow executes the **Compress-VirtualDiskDrive** PowerShell cmdlet in order to compress a dynamically expanding virtual hard disk attached to a virtual disk drive on a virtual machine on Windows-based host managed by Virtual Machine Manager. It reduces the size of the virtual hard disk. The virtual machine must be stopped before the virtual hard disk can be compressed.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### virtualDiskDrive

The name or ID of the virtual disk drive object to be compressed (for example **vhd-21**, **47aff0a3-dc75-41e4-a598-fbee8f81b719**).

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual disk drive object that was compressed.

#### warnings

A list of warnings produced by the cmdlet. If the cmdlet was executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The flow did not complete successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Compress-VirtualDiskDrive -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs in the inputs that accept both names and IDs.
- In case a virtual disk drive is configured on a virtual machine, that virtual disk drive can only be compressed if the virtual machine is in one of the **PowerOff** or **Stored** states. To bring the virtual machine in a state from which it is possible to run the task successfully, you can use the **Repair Virtual Machine**, **Stop Virtual Machine**, or **Store Virtual Ma**chine flows.

## Convert Virtual Disk Drive

The **Convert Virtual Disk Drive** flow executes the **Convert-VirtualDiskDrive** PowerShell cmdlet in order to convert a virtual hard disk attached to a virtual disk drive from fixed to dynamic or from dynamic to fixed. It can also convert a pass-through disk attached to a virtual disk drive to a virtual hard disk.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### virtualDiskDrive

The name or ID of the virtual disk drive object to be converted (for example **vhd-21**, **47aff0a3-dc75-41e4-a598-fbee8f81b719**).

#### fixed

Specifies whether the converted virtual hard disk will be of fixed size. The valid values are **true** and **false**.

#### dynamic

This specifies that the converted virtual hard disk will expand dynamically. The valid values are **true** and **false**.

#### path

The name of a hardware profile object.

#### fileName

The name of an operating system object (for example, **Windows 7** or **Suse Linux Enterprise Server 10 (64 bit)**).

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual disk drive object that was converted.

#### warnings

A list of warnings produced by the cmdlet. If the cmdlet was executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The flow did not complete successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Convert-VirtualDiskDrive -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.
- In case a virtual disk drive is configured on a virtual machine, that virtual disk drive can only be converted if the virtual machine is in one of the **PowerOff** or **Stored** states. To bring the virtual machine in a state from which it is possible to run the task successfully, you can use the **Repair Virtual Machine**, **Stop Virtual Machine**, or **Store Virtual Machine** flows.

## Expand Virtual Disk Drive

The **Expand Virtual Disk Drive** flow executes the **Expand-VirtualDiskDrive** PowerShell cmdlet to expand a virtual hard disk attached to a virtual disk drive on a virtual

machine on Windows-based host managed by Virtual Machine Manager. It increases the total capacity of the virtual hard disk. The virtual machine must be stopped before the virtual hard disk can be expanded.

## Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

### virtualDiskDrive

The name or ID of the virtual disk drive object to be expanded (for example **vhd-21**, **47aff0a3-dc75-41e4-a598-fbee8f81b719**).

### size

The size in megabytes of a fixed virtual hard disk object or the maximum size of a dynamically expanding virtual hard disk.

## Results

The flow returns the following results:

### returnResult

A table containing the details of the virtual disk drive object that was expanded.

### warnings

A list of warnings produced by the cmdlet. If the cmdlet was executed without warnings, this result is empty.

## Responses

The flow returns one of the following responses:

### Success

The PowerShell cmdlets were executed successfully.

### Failure

The flow did not complete successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Expand-VirtualDiskDrive -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.
- In case a virtual disk drive is configured on a virtual machine, that virtual disk drive can only be expanded if the virtual machine is in one of the **PowerOff** or **Stored** states. To bring the virtual machine in a state from which it is possible to run the task

successfully, you can use the **Repair Virtual Machine**, **Stop Virtual Machine**, or **Store Virtual Machine** flows.

## Get Virtual Disk Drive

This **Get Virtual Disk Drive** flow executes the **Get-VirtualDiskDrive** PowerShell cmdlet to get the details of the virtual disk drive objects configured on a virtual machine or on a template.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### vm

The name or ID of the virtual machine object whose virtual disk drives will be retrieved (for example **vm-21**, **47aff0a3-dc75-41e4-a598-fbee8f81b719**).

#### template

The name or ID of the virtual machine object whose virtual disk drives will be retrieved (for example **template-21**, **47aff0a3-dc75-41e4-a598-fbee8f81b719**).

#### all

Specifies whether all of the available virtual disk drive objects should be retrieved. The valid values are **true** and **false**.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual disk drive object that was retrieved.

#### warnings

A list of warnings produced by the cmdlet. If the cmdlet was executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The flow did not complete successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Get-VirtualDiskDrive -Detailed** PowerShell command and inspecting

the **SYNTAX** and **PARAMETERS** sections in the resulting output.

- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## New Virtual Disk Drive

The **New Virtual Disk Drive** flow executes the **New-VirtualDiskDrive** PowerShell cmdlet to create a virtual disk drive on a virtual machine deployed on a host managed by Virtual Machine Manager. It can also create a virtual disk drive on a template stored in the Virtual Machine Manager library.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### vm

The name or ID of the virtual machine to which the new virtual disk drive will be attached (for example **vm-21**, **47aff0a3-dc75-41e4-a598-fbee8f81b719**).

#### template

The name or ID of the template object to which the virtual disk drive will be attached (for example **template-21**, **47aff0a3-dc75-41e4-a598-fbee8f81b719**).

#### ide

Specifies IDE as the bus type to which the virtual disk drive will be attached. The valid values are **true** and **false**.

#### scsi

Specifies SCSI as the bus type to which the virtual disk drive will be attached. The valid values are **true** and **false**.

#### bus

Specifies the IDE or SCSI bus to which to attach the virtual disk drive.

#### lun

The logical unit number (LUN) for the virtual disk drive on an IDE or SCSI bus.

#### virtualHardDisk

The name or ID of the virtual hard disk object that will be associated to the new virtual disk drive (for example **vhd-21**, **47aff0a3-dc75-41e4-a598-fbee8f81b719**).

#### dynamic

This specifies that the new virtual hard disk will expand dynamically. The valid values are **true** and **false**.

### fixed

Specifies whether the new virtual hard disk will be of fixed size. The valid values are **true** and **false**.

### size

The size in megabytes of the fixed virtual hard disk object or the maximum size of a dynamically expanding virtual hard disk.

### path

A destination path specifying the location of the virtual hard disk file attached to the new virtual disk drive. This may be a local path (for example, **D:\my_vhds**) or a UNC path (for example, **LibraryServer\Shared**).

### fileName

The name of the virtual hard disk file attached to the new virtual disk drive.

### bootVolume

Specifies that the volume attached to the new virtual disk drive will be a boot volume. The valid values are **true** and **false**.

### systemVolume

Specifies that the volume attached to the new virtual disk drive will be a system volume. The valid values are **true** and **false**.

## Results

The flow returns the following results:

### returnResult

A table containing the details of the virtual disk drive object that was created.

### warnings

A list of warnings produced by the cmdlet. If the cmdlet was executed without warnings, this result is empty.

## Responses

The flow returns one of the following responses:

### Success

The PowerShell cmdlets were executed successfully.

### Failure

The flow did not complete successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help New-VirtualDiskDrive -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the

description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.

- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.
- In case a virtual disk needs to be configured on a virtual machine, that virtual machine must be in one of the **PowerOff** or **Stored** states. To bring the virtual machine in a state from which it is possible to run the task successfully, you can use the **Repair Virtual Machine**, **Stop Virtual Machine**, or **Store Virtual Machine** flows.

## Remove Virtual Disk Drive

The **Remove Virtual Disk Drive** flow executes the **Remove-VirtualDiskDrive** PowerShell cmdlet to remove a virtual disk drive object from a virtual machine or from a template in the Virtual Machine Manager environment.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### virtualDiskDrive

The name or ID of the virtual disk drive object to be expanded (for example **vhd-21**, **47aff0a3-dc75-41e4-a598-fbee8f81b719**).

#### force

Specifies whether to force the removal of the virtual disk drive object and of all its associations with other objects. The valid values are **true** and **false**.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual disk drive object that was deleted.

#### warnings

A list of warnings produced by the cmdlet. If the cmdlet was executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The flow did not complete successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Remove-VirtualDiskDrive -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.
- In case a virtual disk drive is configured on a virtual machine, that virtual disk drive can only be removed if the virtual machine is in one of the **PowerOff** or **Stored** states. To bring the virtual machine in a state from which it is possible to run the task successfully, you can use the **Repair Virtual Machine**, **Stop Virtual Machine**, or **Store Virtual Machine** flows.

## Set Virtual Disk Drive

The **Set Virtual Disk Drive** flow executes the **Set-VirtualDiskDrive** PowerShell cmdlet to change the properties of a virtual disk drive object on a virtual machine or template. This flow can change the Bus type (IDE or SCSI) of a virtual disk drive and its Bus and LUN settings.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### virtualDiskDrive

The name or ID of the virtual disk drive object to be updated (for example **vhd-21**, **47aff0a3-dc75-41e4-a598-fbee8f81b719**).

#### ide

Specifies IDE as the bus type to which the virtual disk drive will be attached. The valid values are **true** and **false**.

#### scsi

Specifies SCSI as the bus type to which the virtual disk drive will be attached. The valid values are **true** and **false**.

#### bus

Specifies the IDE or SCSI bus to which to attach the virtual disk drive.

#### lun

The logical unit number (LUN) for the virtual disk drive on an IDE or SCSI bus.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual disk drive object that was updated.

#### warnings

A list of warnings produced by the cmdlet. If the cmdlet was executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The flow did not complete successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Set-VirtualDiskDrive -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.
- In case a virtual disk drive is configured on a virtual machine, that virtual disk drive can only be updated if the virtual machine is in one of the **PowerOff** or **Stored** states. To bring the virtual machine in a state from which it is possible to run the task successfully, you can use the **Repair Virtual Machine**, **Stop Virtual Machine**, or **Store Virtual Machine** flows.

## Virtual DVD Drive

### Get Virtual DVD Drive

The **Get Virtual DVD Drive** flow runs the **Get-VirtualDVDDrive** PowerShell cmdlet that gets details about the DVD drive objects from a virtual machine, template, or hardware profile.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

### vm

The name or identifier of the virtual machine whose virtual DVD drives are to be retrieved. The virtual machine can be deployed on a virtual machine host or can be stored in the Virtual Machine Manager library. If you specify a name, it must match the **Name** property of the virtual machine object exactly. To avoid unwanted behavior caused by duplicate names, retrieve the virtual machine by its identifier. The valid format for the identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **MyVM** or **12345678-abcd-efgh-ijkl-1234567890**.

### template

The name or identifier of the template whose virtual DVD drives are to be retrieved. If you specify a name, it must match the **Name** property of the target template exactly. To avoid unwanted behavior caused by duplicate names, provide the template identifier. The valid format for the virtual machine identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **MyTemplate** or **12345678-abcd-efgh-ijkl-1234567890**.

### hardwareProfile

The name of the hardware profile whose virtual DVD drives are to be retrieved. The valid values are **true** and **false**. If you specify a value of **true**, the flow retrieves a list of all of the virtual DVD drives regardless of their parent object. If you specify a value of **false**, the flag is not set.

## Results

The flow returns the following results:

### returnResult

A table containing a row for each virtual DVD drive retrieved or an exception message if the flow failed. The table's columns represent the properties of the DVD drive, in the format `propertyName<delimiter>propertyValue`. If there are no drives that match the given criteria, the result is empty.

### warnings

A list of warnings produced by the PowerShell commands. If the commands were executed without warnings, this result is an empty string.

## Responses

The flow returns one of the following responses:

### Success

The flow executed successfully, and the result contains the virtual DVD drives that matched the given criteria.

### Failure

The flow could not retrieve the list of virtual DVD drives.

**Notes:**

- Not all of the available filters can be applied at the same time. Use the **Get-Help Get-VirtualDVDDrive -Detailed** PowerShell command to see the valid filter combinations and a set of examples.

- PowerShell-related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.

## New Virtual DVD Drive

The **New Virtual DVD Drive** flow runs the **New-VirtualDVDDrive** PowerShell cmdlet that creates a virtual DVD drive on a virtual machine, template, or hardware profile.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### bus

The IDE bus number to which the virtual DVD drive is to be attached.

#### lun

The logical unit number (LUN) for the virtual DVD drive on the IDE bus.

#### vm

The name or identifier of the virtual machine to which the new virtual DVD drive are to be attached. The virtual machine can be deployed on a virtual machine host or can be stored in the Virtual Machine Manager library. If you specify a name, it must match the **Name** property of the virtual machine object exactly. To avoid unwanted behavior caused by duplicate names, provide the virtual machine identifier. The valid format for the virtual machine identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **MyVM** or **12345678-abcd-efgh-ijkl-1234567890**.

#### template

The name or identifier of the template to which the new virtual DVD drive is to be attached. If you specify a name, it must match the **Name** property of the target template exactly. To avoid unwanted behavior caused by duplicate names, specify the template identifier. The valid format for the template identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **MyTemplate** or **12345678-abcd-efgh-ijkl-1234567890**.

#### hardwareProfile

The name of the hardware profile to which the new virtual DVD drive is to be attached.

#### iso

The name or identifier of the ISO file from the Virtual Machine Manager library that is to be attached to the new virtual DVD drive. If you specify a name, it must match the **Name** property of the ISO object exactly. To avoid unwanted behavior caused by duplicate names, specify the ISO identifier. The valid format for the ISO identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **12345678-abcd-efgh-ijkl-1234567890** or **MyIsoFile.iso**.

#### anyHostDrive

Specifies whether the virtual DVD is to be connected to any corresponding physical drive on the host. If you specify any other value except **true**, the flag is considered to be false.

#### hostDrive

The drive on the virtual machine host to connect the new virtual DVD drive from the virtual machine (for example, **D:** or **E:**).

### Results

The flow returns the following results:

#### returnResult

A table containing a row with details about the new virtual DVD drive or an exception message if the flow failed. The table's columns represent the properties of the DVD drive, in the format `propertyName<delimiter>propertyValue`.

#### warnings

A list of warnings produced by the PowerShell commands. If the commands were executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The flow executed successfully and the result contains details about the new virtual DVD drive.

#### Failure

The flow could not create the DVD drive.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help New-VirtualDVDDrive -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell-related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder and the **Run SCVMM Commands** operation **Description** tab.
- If the flow fails to create a virtual DVD drive on a virtual machine, its status might be changed to **Update Failed**. To rerun the operation you must repair the virtual machine from the SCVMM Admin Console or by using the **Repair Virtual Machine** operation.

## Remove Virtual DVD Drive

The **Remove Virtual DVD Drive** flow runs the **Remove-VirtualDVDDrive** PowerShell cmdlet that removes a virtual DVD drive from a virtual machine, template, or hardware profile.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### virtualDvdDrive

The identifier of the virtual DVD drive to be removed. The valid format is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **12345678-abcd-efgh-ijkl-1234567890**.

### Results

The flow returns the following results:

#### returnResult

A table containing a row with details about the removed virtual DVD drive or an exception message if the flow failed. The table's columns represent the properties of the virtual DVD drive, in the format **propertyName<delimiter>propertyValue**.

#### warnings

A list of warnings produced by PowerShell commands. If the commands were executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The flow was successful and the result contains details about the removed virtual DVD drive.

#### Failure

The flow could not remove the virtual DVD drive.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help New-VirtualDVDDrive -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.

## Set Virtual DVD Drive

The **Set Virtual DVD Drive** flow runs the **Set-VirtualDVDDrive** PowerShell cmdlet that changes the properties of a virtual DVD drive on a virtual machine, template, or hardware profile.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### virtualDvdDrive

The identifier of the virtual DVD drive whose properties to be updated. The valid format is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **12345678-abcd-efgh-ijkl-1234567890**.

### bus

The IDE bus number where you want to move the virtual DVD drive.

### lun

The logical unit number (LUN) where you want to move the virtual DVD drive on the IDE bus.

### iso

The name or identifier of the ISO file from the Virtual Machine Manager library to be attached to the virtual DVD drive. If you specify a name, it must match the **Name** property of the ISO object exactly. To avoid unwanted behavior caused by duplicate names, specify the ISO identifier. The valid format for the ISO identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **12345678-abcd-efgh-ijkl-1234567890** or **MyIsoFile.iso**.

### anyHostDrive

Specifies whether the virtual DVD will be connected to any corresponding physical drive on the host. For any other value except **true,** the flag is considered to be **false**.

### hostDrive

The drive on the virtual machine host to which the flow is to connect the virtual DVD drive from the virtual machine. For example, **D:** or **E:**.

### noMedia

Specifies whether to disconnect the virtual DVD drive from the host drive or ISO to which it is connected. If you specify any value other than **true**, the virtual DVD drive is not disconnected.

## Results

The flow returns the following results:

### returnResult

A table containing a row with details about the updated virtual DVD drive or an exception message if the flow failed. The table's columns represent the properties of the virtual DVD drive, in the format `propertyName<delimiter>propertyValue.`

### warnings

A list of warnings produced by the PowerShell commands. If the commands were executed without warnings, this result is empty.

## Responses

The flow returns one of the following responses:

### Success

The flow executed successfully and the result contains details about the virtual DVD drive.

### Failure

The flow could not update the virtual DVD drive.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Set-VirtualDVDDrive -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell-related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder and the **Run SCVMM Commands** operation's **Description** tab.
- If the flow fails to update a virtual DVD drive on a virtual machine, its status might be changed to **Update Failed**. To retry the operation, you must repair the virtual machine from the SCVMM Admin Console or by using the **Repair Virtual Machine** flow.

# Virtual Hard Disk

## Get Virtual Hard Disk

The **Get Virtual Hard Disk** flow executes the **Get-VirtualHardDisk** PowerShell cmdlet to get the details of the virtual hard disk objects attached to a virtual machine or to a template.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### vm

The name or id of the virtual machine object whose virtual hard disks are to be retrieved. For example, **vm-21** or **47aff0a3-dc75-41e4-a598-fbee8f81b719**.

#### template

The name or ID of the virtual machine object whose virtual hard disks are to be retrieved. For example, **template-21** or **47aff0a3-dc75-41e4-a598-fbee8f81b719**.

#### all

Specifies whether you want to retrieve all the available virtual hard disk objects. The valid values are **true** and **false**.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual hard disk objects that were retrieved.

#### warnings

A list of warnings that the cmdlet produced. If the cmdlet was executed without warnings, this result is an empty string.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The PowerShell cmdlets were not executed successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Get-VirtualHardDisk -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell-related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder and the **Run SCVMM Commands** operation's **Description** tab.

## Move Virtual Hard Disk

The **Move Virtual Hard Disk** flow executes the **Move-VirtualHardDisk** PowerShell cmdlet to move a virtual hard disk object in the Virtual Machine Manager library.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### virtualHardDisk

The name or ID of the virtual hard disk object to be moved. For example, **vhd-21** or **47aff0a3-dc75-41e4-a598-fbee8f81b719**.

#### path

The destination path where you want to move the virtual hard disk. This should be a local path (for example, **D:\my_vhds**).

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual hard disk object that was moved.

#### warnings

A list of warnings that the cmdlet produced. If the cmdlet was executed without warnings, this result is an empty string.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The PowerShell cmdlets were not executed successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Move-VirtualHardDisk -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell-related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object ID in the inputs that accept both names and IDs.

### Remove Virtual Hard Disk

The **Remove Virtual Hard Disk** flow executes the **Remove-VirtualHardDisk** PowerShell cmdlet to remove a virtual hard disk object from the Virtual Machine Manager library.

#### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

##### virtualHardDisk

The name or ID of the virtual hard disk object to be removed. For example, **vhd-21** or **47aff0a3-dc75-41e4-a598-fbee8f81b719**.

##### force

Specifies whether to force the removal of the virtual hard disk object and of all its associations with other objects. The valid values are **true** and **false**.

#### Results

The flow returns the following results:

##### returnResult

A table containing the details of the virtual hard disk object that was removed.

##### warnings

A list of warnings that the cmdlet produced. If the cmdlet was executed without warnings, this result is an empty string.

#### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The PowerShell cmdlets were not executed successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Remove-VirtualHardDisk -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell-related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object ID in the inputs that accept both names and IDs.

## Set Virtual Hard Disk

The **Set Virtual Hard Disk** flow executes the **Set-VirtualHardDisk** PowerShell cmdlet to change the properties of a virtual hard disk object used in a Virtual Machine Manager environment.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### virtualHardDisk

The name or ID of the virtual hard disk object to be updated. For example, **vhd-21** or **47aff0a3-dc75-41e4-a598-fbee8f81b719**.

#### name

The new name of the virtual hard disk.

#### description

The new description of the virtual hard disk.

#### enabled

Specifies whether you want the virtual hard disk to be enabled. The valid values are **true** and **false**. If you specify a value other than **true**, the virtual hard disk is disabled. For null values this property does not change.

#### operatingSystem

The name of the new operating system object associated to this virtual hard disk. For example**, Windows 7** or **Suse Linux Enterprise Server 10 (64 bit)**.

#### owner

The new owner of the virtual hard disk in the form of a valid domain user account. The valid format is **`username@domain or domain\username`**.

#### sharePath

A path to a valid library share on an existing library server that uses a Universal Naming Convention (UNC) path. For example, **FileServer01****LibShare**.

#### virtualizationPlatform

The virtualization platform of the virtual hard disk. The valid values are **VirtualServer**, **HyperV**, and **VMWareESX**.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual hard disk object that was updated.

#### warnings

A list of warnings that the cmdlet produced. If the cmdlet was executed without warnings, this result is an empty string.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The PowerShell cmdlets were not executed successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Set-VirtualHardDisk -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell-related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object ID in the inputs that accept both names and IDs.

## Virtual Machine

### Disable Virtual Machine Undo Disk

The **Disable Virtual Machine Undo** Disk flow executes the **DisableUndoDisk-VM** PowerShell cmdlet that merges or discards undo disks associated with a virtual machine on a Virtual Server host managed by Virtual Machine Manager.

## Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

### vm

The name or ID of the virtual machine whose undo disk was merged or discarded. The valid format for the VM identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example (of VM identifier), **12345678-abcd-efgh-ijkl-1234567890**.

### discard

Specifies whether you want to discard an undo disk associated with the virtual machine. The valid values are **true** and **false**.

### merge

Specifies whether you want to merge an undo disk associated with the virtual machine. The valid values are **true** and **false**

## Results

The flow returns the following results:

### returnResult

A table containing the details of the virtual machine whose undo disk was merged or discarded.

### warnings

A list of warnings produced by the cmdlet. If the cmdlet was executed without warnings, this result is empty.

## Responses

The flow returns one of the following responses:

### Success

The PowerShell cmdlets executed successfully.

### Failure

The flow was unsuccessful.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help DisableUndoDisk-VM -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell-related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## Discard Virtual Machine State

The **Discard Virtual Machine State** flow executes the **DiscardSavedState-VM** PowerShell cmdlet that discards the state of a virtual machine. Discarding the saved state of a virtual machine returns its object in a stopped state.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### vm

The name or ID of the virtual machine whose state is to be discarded. The valid format for the VM identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example (of VM identifier), **12345678-abcd-efgh-ijkl-1234567890**.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual machine whose state was discarded.

#### warnings

A list of warnings produced by the cmdlet. If the cmdlet was executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The flow was unsuccessful.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help DiscardSavedState-VM -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell-related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## Get Virtual Machine

The **Get Virtual Machine** flow runs the **Get-VM** PowerShell command that gets details about the virtual machines from the Virtual Machine Manager database. A virtual machine can be deployed on a virtual machine host or can be stored in the Virtual Machine Manager library.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### name

The name of the virtual machine.

#### id

The identifier of the virtual machine. The valid format is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **12345678-abcd-efgh-ijkl-1234567890**.

#### vmHost

The name of the virtual machine host as it appears in the **Name** property of the object. This can be a managed Hyper-V or Virtual Server host. For example, **myVmHost.domain** or **1.2.3.4**.

#### all

Specify the value as **true** to retrieve a list of virtual machines regardless of their parent object. If you specify any value except **true**, the value is set to **false**.

### Results

The flow returns the following results:

#### returnResult

A table containing a row for each virtual machine retrieved or an exception message if the flow failed. The table's columns represent the properties of the virtual machines, in the format `propertyName<delimiter>propertyValue`. If there are no virtual machines that match the given criteria, the result is empty.

#### warnings

A list of warnings produced by the PowerShell commands If the commands were executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The flow executed successfully and the result contains the virtual machines that matched the given criteria.

#### Failure

The flow could not retrieve the list of virtual machines.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Get-VM -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell-related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.

## Move Virtual Machine

The **Move Virtual Machine** flow runs the **Move-VM** PowerShell cmdlet that moves a virtual machine stored in the Virtual Machine Manager library or deployed on a host server to a new location on a host server.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### vm

The name or identifier of the virtual machine. This can be deployed on a virtual machine host or can be stored in the Virtual Machine Manager library. The name you specify must match the **Name** property of the virtual machine object exactly. To avoid unwanted behavior caused by duplicate names, provide the virtual machine identifier. The valid format for the virtual machine identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx.** For example, **MyVM** or **12345678-abcd-efgh-ijkl-1234567890**.

#### vmHost

The name or identifier of the virtual machine host. This can be a managed Hyper-V or Virtual Server host. If you specify a name, it must match the **Name** property of the virtual machine host object exactly. To avoid unwanted behavior caused by duplicate names, specify the virtual machine host identifier. The valid format for the virtual machine host identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **myVmHost.domain** or **12345678-abcd-efgh-ijkl-1234567890**.

#### path

The destination path for the operation. This should be a local path relative to the given virtual machine host (for examples, **C:\MyVMFolder**.)

#### blockLmIfHostBusy

Specifies whether to block retrying a Hyper-V live migration if the migration failed because the source host or the destination host is already participating in another live migration. The valid values are **true** and **false**. If you do not specify a value of **true**, it is set to **false**. If you do not specify a value for this input, the property is not changed.

### startVmOnTarget

Specify a value of **true** to start a virtual machine as soon as it reaches its destination host. If you specify any value other than **true**, the value is set to **false**. If you do not specify a value for this input, the property is not changed.

### useCluster

Specify a value of **true** to force the use of Windows Server 2008 Cluster Migration for the transfer of a virtual machine that is in a saved state to a host, even if the cluster supports Hyper-V live migration. If you specify any value other than **true**, the valuet is set to **false**. If you do not specify a value for this input, the property is not changed.

### useLan

Specify a value of **true** to force a transfer over the local area network (LAN) even if a faster transfer mechanism, such as a storage area network (SAN) transfer, is available. If you specify any value other than **true**, the value is set to **false**. If you do not specify a value for this input, the property is not changed.

## Results

The flow returns the following results:

### returnResult

A table containing a row with details about the moved virtual machine or an exception message if the flow failed. The table's columns represent the properties of the virtual machines, in the format `propertyName<delimiter>propertyValue`.

### warnings

A list of warnings produced by the PowerShell commands. If the commands were executed without warnings, this result is an empty string.

## Responses

The flow returns one of the following responses:

### Success

The operation executed successfully and the result contains details about the virtual machine.

### Failure

The flow could not move the virtual machine.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Move-VM -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell-related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder and the **Run SCVMM Commands** operation's **Description** tab.

## New Virtual Machine

The **New Virtual Machine** flow runs the **New-VM** PowerShell cmdlet that creates a virtual machine to be managed by Virtual Machine Manager.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### name

The name of the new virtual machine.

#### vmHost

The name or identifier of the virtual machine host. This can be a managed Hyper-V or Virtual Server host. If you specify a name, it must match the **Name** property of the virtual machine host object exactly. To avoid unwanted behavior caused by duplicate names, specify the virtual machine host identifier. The valid format for the virtual machine host identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **myVmHost.domain** or **12345678-abcd-efgh-ijkl-1234567890**.

#### path

The destination path for the operation. This should be a local path relative to the given virtual machine host.(for example, **C:\MyVMFolder**).

#### virtualHardDisk

The name or identifier of the virtual hard disk to be attached to the new virtual machine. This should be a VHD file from the Virtual Machine Manager library. If you specify a name, it must match the **Name** property of the hard disk object exactly. To avoid unwanted behavior caused by duplicate names, specify the virtual hard disk identifier. The valid format for the virtual hard disk identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **myVirtualHardDisk**, **Blank Disk - Small**, **Blank Disk - Large**, or **12345678-abcd-efgh-ijkl-1234567890**.

#### template

The name or identifier of the template used to create the virtual machine. If you specify a name, it must match the **Name** property of the target template exactly. To avoid unwanted behavior caused by duplicate names, specify the template identifier. The valid format for the template identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **MyTemplate** or **12345678-abcd-efgh-ijkl-1234567890**.

#### productKey

The product key to use for the operating system to be installed on the virtual machine. The valid format is **xxxxx-xxxxx-xxxxx-xxxxx-xxxxx**.

### libraryServer

The name or identifier of the Virtual Machine Manager library server. The name you specify must match the **Name** property of the library server object exactly. To avoid unwanted behavior caused by duplicate names, specify the library server identifier. The valid format for the virtual server identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **SCVMMLib.domain** or **12345678-abcd-efgh-ijkl-1234567890**.

### sharePath

The path to a valid library share on the provided library server. The valid format is the UNC path (for example, **SCVMM.domain****MyShare**).

### vm

The name or identifier of a virtual machine to be cloned into the new virtual machine. This can be deployed on a virtual machine host or can be stored in the Virtual Machine Manager library. The name you specify must match the **Name** property of the virtual machine object exactly. To avoid unwanted behavior caused by duplicate names, specify the virtual machine identifier. The valid format for the virtual machine identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **MyVM** or **12345678-abcd-efgh-ijkl-1234567890**.

### description

The description of the new virtual machine.

### owner

The owner of the virtual machine in the form of a valid domain user account (**domain\username**).

### hardwareProfile

The name of the hardware profile to be applied on the new virtual machine.

### cpuType

The type of CPU for the virtual machine. To retrieve a list with all the CPU types available use the **Get-CPUType** command. For example, **3.40 GHz Pentium D (dual core)**, **2.40 GHz Opteron**, or **3.07 GHz Xeon**.

### cpuCount

The number of CPUs to set on the virtual machine. The maximum number of processors you can set on a virtual machine depends on the type of virtual machine host.

### limitCpuForMigration

Specifies whether to limit the processor features for the specified virtual machine to enable migration to a physical computer that has a different version of the same processor as the source computer. If you specify any value other than **true**, the input value is set to **false**. If you do not specify a value, the property is not changed.

### limitCpuFunctionality

Specifies whether to enable running an older operating system (such as Windows NT 4.0) on a virtual machine by providing only limited CPU functionality for the virtual machine. If you specify any value other than **true**, the input value is set to **false**. If you do not specify a value, the property is not changed.

### memoryMb

The total amount of memory in MB assigned to the virtual machine. The maximum host memory assignable to the virtual machine depends on the type of virtual machine host.

### startAction

Specifies the behavior of a virtual machine when the virtualization service starts. The valid values are **AlwaysAutoTurnOnVM**, **NeverAutoTurnOnVM**, and **TurnOnVMIfRunningWhenVSStopped**.

### delayStart

The number of seconds to wait after the virtualization service starts before automatically starting the virtual machine.

### stopAction

Specifies the behavior of the virtual machine when the virtualization service stops. The valid values are **SaveVM**, **TurnOffVM**, and **ShutdownGuestOS**.

### operatingSystem

The type of operating system for the virtual machine (for example, **Windows 7** or **Ubuntu Linux (64 bit)).** To retrieve a list with all the available operating systems, use the **Get-OperatingSystem** command. If you specify a value that is not in the list of available operating systems, the operating system is set to **Unknown**.

## Results

The flow returns the following results:

### returnResult

A table containing a row with details about the new virtual machine or an exception message if the flow failed. The table's columns represent the properties of the virtual machine, in the format `propertyName<delimiter>propertyValue`.

### warnings

A list of warnings produced by the PowerShell commands. If the commands were executed without warnings,  this result is empty.

## Responses

The flow returns one of the following responses:

### Success

The operation executed successfully and the result contains details about the new virtual machine.

### Failure

The flow could not create the virtual machine.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help New-VM -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell-related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder and the **Run SCVMM Commands** operation's **Description** tab.

## Refresh Virtual Machine

The **Refresh Virtual Machine** flow executes the **Refresh-VM** PowerShell cmdlet that refreshes a virtual machine in Virtual Machine Manager.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### vm

The name or ID of the virtual machine to be refreshed. The valid format for the VM identifier is xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx. Example (of the VM identifier), **12345678-abcd-efgh-ijkl-1234567890**.

#### force

Specifies whether to force the refresh operation on the virtual machine object. The valid values are **true** and **false**.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual machine that was refreshed.

#### warnings

A list of warnings produced by the cmdlet. If the cmdlet was executed without warnings, this result is an empty string.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The flow could not refresh the virtual machine.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Refresh-Template -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## Register Virtual Machine

The **Register Virtual Machine** flow executes the **Register-VM** PowerShell cmdlet to register an existing virtual machine with Virtual Machine Manager that is currently not registered with any host managed by Virtual Machine Manager and is not stored in the Virtual Machine Manager library.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### vmHost

The name or identifier of the virtual machine host. This can be a managed Hyper-V or Virtual Server host. If you provide a name it must exactly match the **Name** property of the virtual machine host object. The valid format for the virtual machine host identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **myVmHost.domain**, **12345678-abcd-efgh-ijkl-1234567890**.

#### path

The path to a folder where the configuration files for the virtual machine are stored (on the host's file system or on shared storage).

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual machine that was registered.

#### warnings

A list of warnings that the cmdlet produced. If the cmdlet was executed without warnings, this result is an empty string.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The flow could not register the virtual machine.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Register-VM -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## Remove Virtual Machine

The **Remove Virtual Machine** flow executes the **Remove-VM** PowerShell cmdlet to remove a virtual machine from Virtual Machine Manager.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### vm

The name or ID of the virtual machine to be removed. The valid format for the VM identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example (of VM identifier), **12345678-abcd-efgh-ijkl-1234567890**.

#### force

Specifies whether you want to force the removal of the virtual machine object and of all its associations with other objects. The valid values are **true** and **false**.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual machine that was removed.

#### warnings

A list of warnings that the cmdlet produced. If the cmdlet was executed without warnings, this result is an empty string.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The flow was unable to remove the virtual machine.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Remove-VM -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## Repair Virtual Machine

The **Repair Virtual Machine** flow executes the **Repair-VM** PowerShell cmdlet to repair a virtual machine (if it is in a failed state) on a host managed by Virtual Machine Manager.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### vm

The name or ID of the virtual machine to be repaired. The valid format for the VM identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example (of the VM identifier, **12345678-abcd-efgh-ijkl-1234567890**.

#### dismiss

Specifies whether you want to dismiss the error on the virtual machine object and refresh it. The valid values are **true** and **false**.

#### undo

Specifies whether you want to cancel the last job run on a virtual machine object and reverse any changes that were made to it. The valid values are **true** and **false**.

#### retry

Specifies whether you want to retry the latest job on the virtual machine in an attempt to complete it successfully. The valid values are **true** and **false**.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual machine that was repaired.

#### warnings

A list of warnings that the cmdlet produced. If the cmdlet was executed without warnings, this result is an empty string.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully

#### Failure

The flow was unable to repair the virtual machine.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Repair-VM -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## Resume Virtual Machine

The **Resume Virtual Machine** flow resumes paused virtual machines managed by the Virtual Machine Manager.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### vm

The ID or name of the virtual machine to resume. The valid format for the VM identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example (of VM identifier), **12345678-abcd-efgh-ijkl-1234567890**.

### Results

The flow returns the following results:

#### returnResult

The properties of the virtual machine that was resumed. The properties are displayed in the `propertyName<delimiter>propertyValue` format.

#### warnings

A list of warnings that the flow returns. If the **Resume-VM** cmdlet was executed without warnings, this result is an empty string.

### Responses

The flow returns one of the following responses:

#### Success

The **Resume-VM** cmdlet was executed successfully.

#### Failure

The **Resume-VM** cmdlet could not be executed.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Resume-VM -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## Save Virtual Machine State

The **Save Virtual Machine State** flow executes the **SaveState-VM** PowerShell cmdlet to save the state (status and configuration changes) of a virtual machine. To enable the saving of its state, the virtual machine must be in a running state.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### vm

The name or ID of the virtual machine whose state will be saved. The valid format for the VM identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example (of the VM identifier) is **12345678-abcd-efgh-ijkl-1234567890**.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual machine whose state was saved.

#### warnings

A list of warnings that the cmdlet produced. If the cmdlet was executed without warnings, this result is an empty string.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The PowerShell cmdlets were not executed successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help SaveState-VM -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## Set Virtual Machine

The **Set Virtual Machine** flow runs the PowerShell command **Set-VM** to change the properties of a virtual machine managed by Virtual Machine Manager.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### vm

The name or identifier of the virtual machine . This can be deployed on a virtual machine host or in the Virtual Machine Manager library. The name must exactly match the **Name** property of the virtual machine object . To avoid unwanted behavior caused by duplicate names, specify the virtual machine identifier. The valid format for the virtual machine identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **MyVM**, **12345678-abcd-efgh-ijkl-1234567890**.

#### name

The name of the virtual machine.

#### description

The description of the virtual machine.

#### tag

A word or phrase that will be associated with the virtual machine.

#### owner

The owner of the virtual machine in the form of a valid domain user account. The valid format is `domain\username`.

### bootOrder

The order of devices that a virtual machine uses to start up. The valid values are **CD**, **IDEHardDrive**, **PXEBoot,** and **Floppy**. For example, **PXEBoot,IDEHardDrive,CD,Floppy**.

### numLock

Specify a value of **true** to enable the BIOS value for NumLock on a virtual machine on a Hyper-V host. If this input has an empty value, the property is not changed. If you specify any value other than **true**, the value is set to **false**.

### cpuType

The type of CPU for the virtual machine. To retrieve a list with all the CPU types available, use the **Get-CPUType** PowerShell command. For example, **3.40 GHz Pentium D (dual core)**, **2.40 GHz Opteron**, or **3.07 GHz Xeon**.

### cpuCount

The number of CPUs to set on the virtual machine. The maximum number of processors you can set on a virtual machine depends on the type of virtual machine host.

### limitCpuForMigration

Specifies whether to limit the processor features for the specified virtual machine to enable migration to a physical computer that has a different version of the same processor as the source computer. If the value is empty, the property is not changed. For any other value except **true**, it is set to **false**.

### limitCpuFunctionality

Specifies if running an older operating system (such as Windows NT 4.0) on a virtual machine by providing only limited CPU functionality for the virtual machine is enabled. If the value is empty, the property is not changed. For any other value except **true**, it is set to **false**.

### memoryMb

The total amount of memory (in megabytes) assigned to the virtual machine. The maximum host memory assignable to the virtual machine depends on the type of virtual machine host.

### startAction

Specifies the behavior of a virtual machine when the virtualization service starts. The valid values are **AlwaysAutoTurnOnVM**, **NeverAutoTurnOnVM**, and **TurnOnVMIfRunningWhenVSStopped**.

### delayStart

The number of seconds to wait after the virtualization service starts before automatically starting the virtual machine.

### stopAction

Specifies the behavior of the virtual machine when the virtualization service stops. The valid values are **SaveVM**, **TurnOffVM**, and **ShutdownGuestOS**.

### operatingSystem

The type of operating system for the virtual machine. To retrieve a list of all the available operating systems, use the **Get-OperatingSystem** command. For any other values the operating system is set to **Unknown**. For example, **Windows 7** or **Ubuntu Linux (64 bit)**.

### enableBackup

Specifies whether the use of the **Volume Shadow Copy** service is enabled to back up a virtual machine if the virtual machine is deployed on a Hyper-V host. If the value is empty, the property is not changed. For any other value except **true,** it is set to **false**.

### enableDataExchange

Specifies whether the use of a key/value pair for the exchange of data between the virtual machine and the host operating system is enabled if the virtual machine is deployed on a Hyper-V host. If the value is empty, the property is not changed. For any value except **true**, it sets the value to **false**.

### enableHeartbeat

Specifies whether the use of a heartbeat to monitor the health of the virtual machine is enabled if it is deployed on a Hyper-V host. If the value is empty, the property is not changed. For any value except **true**, it is set to **false**.

### enableOperatingSystemShutdown

Specifies whether to shut down the operating system on the virtual machine from Hyper-V's management interfaces on the host if the virtual machine is deployed on a Hyper-V host. If the value is empty, the property is not changed. For any value except **true**, it is set to false.

### enableTimeSynchronization

Specifies whether  synchronizing the system time of a virtual machine with the system time of the operating system running on the host is enabled if the virtual machine is deployed on a Hyper-V host. If the value is empty, the property is not changed. For any value except **true**, it is set to **false**.

### highlyAvailable

Specifies whether  a virtual machine is placed on a Hyper-V host that is part of a host cluster. If the value is empty, the property is not changed. For any value except **true**, it is set to **false**.

## Results

The flow returns the following results:

### returnResult

A table containing a row with details about the virtual machine or an exception message if the operation failed. The table's columns represent the properties of the virtual machine, in the `propertyName<delimiter>propertyValue` format.

### warnings

A list of warnings that the PowerShell commands emit. If the commands were executed without warnings, this result is an empty string.

### Responses

The flow returns one of the following responses:

#### Success

The flow executed with success and the result contains details about the virtual machine.

#### Failure

The flow could not update the virtual machine.

**Notes:**

- Not all of the available inputs can be applied at the same time. Use the **Get-Help Set-VM -Detailed** PowerShell command to see the valid input combinations and a set of examples.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.

## Shut Down Virtual Machine

The **Shut Down Virtual Machine** flow executes the **ShutDown-VM** PowerShell cmdlet to shut down a virtual machine on a host managed by Virtual Machine Manager. A Hyper-V virtual machine can only be shut down from SCVMM if it has Virtual Guest Services installed.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### vm

The name or ID of the virtual machine to be shut down. The valid format for the VM identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example (of VM identifier),: **12345678-abcd-efgh-ijkl-1234567890**.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual machine that was shut down.

#### warnings

A list of warnings that the cmdlet produced. If the cmdlet was executed without warnings, this result is an empty string.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The PowerShell cmdlets were not executed successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Shutdown-VM -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## Start Virtual Machine

The **Start Virtual Machine** flow executes the **Start-VM** PowerShell cmdlet to start a virtual machine on a host managed by Virtual Machine Manager. The flow will also succeed if the targeted virtual machine is already running but no changes will occur.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### vm

The name or ID of the virtual machine to be started. The valid format for the VM identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example (of VM identifier), **12345678-abcd-efgh-ijkl-1234567890**.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual machine that was started.

#### warnings

A list of warnings that the cmdlet produced. If the cmdlet was executed without warnings, this result is an empty string.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The PowerShell cmdlets did not execute successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Start-VM -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## Stop Virtual Machine

The **Stop Virtual Machine** flow executes the **Stop-VM** PowerShell cmdlet to stop a virtual machine on a host managed by Virtual Machine Manager. The flow also succeeds if the targeted virtual machine is already stopped, but no changes will occur.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### vm

The name or ID of the virtual machine to be stopped. The valid format for the VM identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example (of VM identifier), **12345678-abcd-efgh-ijkl-1234567890**.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual machine that was stopped.

#### warnings

A list of warnings that the cmdlet produced. If the cmdlet was executed without warnings, this result is an empty string.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The PowerShell cmdlets did not execute successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Stop-VM -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## Store Virtual Machine

The **Store Virtual Machine** flow executes the **Store-VM** PowerShell cmdlet to store a virtual machine currently deployed on a virtual machine host by migrating it from the host to the Virtual Machine Manager library.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### vm

The name or ID of the virtual machine to be stored. The valid format for the VM identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example (of VM identifier), **12345678-abcd-efgh-ijkl-1234567890**..

#### libraryServer

The name or id of the library server, where the virtual machine will be stored.

#### sharePath

The path on the library server, where the virtual machine will be stored.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual machine that was stored.

#### warnings

A list of warnings that the cmdlet produced. If the cmdlet was executed without warnings, then this result will be an empty string.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The PowerShell cmdlets did not execute successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Store-VM -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## Suspend Virtual Machine

The **Suspend Virtual Machine** flow suspends the execution on a virtual machine managed by the Virtual Machine Manager.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### vm

The name or ID of the virtual machine to be stopped. The valid format for the VM identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example (of VM identifier), **12345678-abcd-efgh-ijkl-1234567890**..

### Results

The flow returns the following results:

#### returnResult

The properties of the virtual machine that was suspended. The properties are displayed in the `propertyName<delimiter>propertyValue` format.

#### warnings

A list of warnings that the flow returns. If the **Suspend-VM** cmdlet was executed without warnings, this result is an empty string.

### Responses

The flow returns one of the following responses:

#### Success

The **Suspend-VM** cmdlet was executed successfully.

#### Failure

The **Suspend-VM** cmdlet could not be executed.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Suspend-VM -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

# Virtual Network

## Get Virtual Network

The **Get Virtual Network** flow executes the **Get-VirtualNetwork** PowerShell cmdlet to get virtual network objects configured on a host managed by Virtual Machine Manager.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### name

The name of the virtual network to get.

#### vmHost

The ID or name of the virtual machine host.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual network objects that were retrieved.

#### warnings

A list of warnings that the cmdlet produced. If the cmdlet was executed without warnings, this result is an empty string.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The PowerShell cmdlets did not execute successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Get-VirtualNetwork -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## New Virtual Network

The **New Virtual Networ**k flow executes the **New-VirtualNetwork** PowerShell cmdlet to create a virtual network object on a host managed by Virtual Machine Manager that enables virtual machines on that host to communicate over that virtual network..

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### name

The name of the virtual network.

#### vmHost

The ID or name of the virtual machine host.

#### boundToVMHost

Specifies whether you want to enable virtual machines connected to the virtual network on a Hyper-V host to access the host operating system. The valid values are **true** and **false**.

#### description

The description of the virtual network.

#### hostBoundVLanId

A numerical identifier for the virtual LAN (VLAN) that virtual machines use to access the host. The valid values are **1-4094**.

#### networkTag

The name of the network tag to associate with the virtual network.

#### path

The destination path for the operation.

#### vmHostNetworkAdapter

The ID or name of the physical network adapter associated with a host to which virtual machines deployed on that host can connect.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual network object that was created.

#### warnings

A list of warnings that the cmdlet produced. If the cmdlet was executed without warnings, his result is an empty string.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The PowerShell cmdlets did not execute successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help New-VirtualNetwork -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## Remove Virtual Network

The **Remove Virtual Network** flow executes the **Remove-VirtualNetwork** PowerShell cmdlet to remove a virtual network object from a host managed by Virtual Machine Manager.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### virtualNetwork

The ID or name of the virtual network whose properties you want to change.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual network object that was removed.

#### warnings

A list of warnings that the cmdlet produced. If the cmdlet was executed without warnings, this result is an empty string.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The PowerShell cmdlets did not execute successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Remove-VirtualNetwork -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## Set Virtual Network

The **Set Virtual Network** flow executes the **Set-Network** PowerShell cmdlet to change properties of a virtual network on a virtual machine host managed by Virtual Machine Manager.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### virtualNetwork

The ID or name of the virtual network whose properties you want to change.

#### boundToVMHost

Specifies whether you want to enable virtual machines connected to the virtual network on a Hyper-V host to access the host operating system. The valid values are **true** and **false**.

#### description

The description of the virtual network.

#### hostBoundVLanId

A numerical identifier for the virtual LAN (VLAN) that virtual machines use to access the host. The valid values are **1-4094.**

#### name

The ID or name of the virtual network.

#### networkTag

The name of the network tag to associate with the virtual network.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual network object that was changed.

#### warnings

A list of warnings that the cmdlet produced. If the cmdlet was executed without warnings, this result is an empty string.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The PowerShell cmdlets did not execute successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Set-VirtualNetwork -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## Virtual Network Adapter

### Get Virtual Network Adapter

The **Get Virtual Network Adapter** flow runs the **Get-VirtualNetworkAdapter** PowerShell command to get Virtual Machine Manager virtual network adapter objects from a virtual machine, template, or hardware profile.

## Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

### vm

The name or ID of the virtual machine from which to get the virtual network adapters. The valid format for the VM identifier **`is xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx`**. For example (of the VM identifier), **12345678-abcd-efgh-ijkl-1234567890**.

### template

The name or ID of the template from which to get the virtual network adapters. The valid format for the template identifier is **`xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx`**. For example, **MyTemplate** or **12345678-abcd-efgh-ijkl-1234567890.**

### hardwareProfile

The name or ID of the hardware profile from which to get the virtual network adapters. The valid format for the hardware profile identifier is **`xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx`**. For example, **MyHardwareProfile** or **12345678-abcd-efgh-ijkl-1234567890**.

### all

Specifies whether you want to get all virtual network adapters. The valid values are **true** and **false**.

## Results

The flow returns the following results:

### returnResult

A table containing the details of the virtual network adapter objects that were retrieved.

### warnings

A list of warnings that the cmdlet produced. If the cmdlet was executed without warnings, this result is an empty string.

## Responses

The flow returns one of the following responses:

### Success

The Powershell cmdlets were executed successfully.

### Failure

The Powershell cmdlets were not executed successfully.

**Notes:**

- Only one of the **vm**, **template**, **hardwareProfile**, or **all** inputs must have a value assigned in the flow.

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Get-VirtualNetworkAdapter -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## New Virtual Network Adapter

The **New Virtual Network Adapter** flow runs the **New-VirtualNetworkAdapter** PowerShell command to create a virtual network adapter on a virtual machine, template, or hardware profile used in Virtual Machine Manager.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### vm

The name or ID of the virtual machine where you want to create the virtual network adapters. The valid format for the VM identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example (of VM identifier), **12345678-abcd-efgh-ijkl-1234567890**.

#### template

The ID or name of the template where you want to create the virtual network adapters. The valid format for the template identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **MyTemplate** or **12345678-abcd-efgh-ijkl-1234567890**.

#### hardwareProfile

The ID or name of the hardware profile where you want to create the virtual network adapters. The valid format for the hardware profile identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **MyHardwareProfile** or **12345678-abcd-efgh-ijkl-1234567890.**

#### virtualNetwork

The ID or name of the virtual network.

#### macAddressesSpoofingEnabled

Specifies whether you want to enable MAC address spoofing. The valid values are **true** and **false**.

#### networkLocation

The name of the network location.

#### networkTag

The name of the network tag to associate with the virtual network adapter.

### noConnection

Specifies whether you want to disconnect the virtual network adapter from the virtual network. The valid values are **true** and **false**.

### physicalAddress

The physical address (MAC address) of the virtual network adapter.

### physicalAddressType

The type of physical address to use for the virtual network adapter. The valid values are **Static** and **Dynamic**.

### synthetic

Specifies whether the virtual network adapter is a high performance synthetic device. The valid values are **true** and **false**.

### vlanEnabled

Specifies whether you want to enable virtual LAN for the virtual network adapter. The valid values are **true** and **false**.

### vlanId

The numerical VLAN identifier for the virtual network adapter. The valid values are **1**-**4094**.

### vmNetworkOptimizationEnabled

Specifies whether you want to enable virtual machine network optimization. The valid values are **true** and **false**.

### vmwarePortGroup

The name of the VMware port group.

## Results

The flow returns the following results:

### returnResult

A table containing the details of the virtual network adapter object that was created.

### warnings

A list of warnings that the cmdlet produced. If the cmdlet was executed without warnings, his result is an empty string.

## Responses

The flow returns one of the following responses:

### Success

The Powershell cmdlets were executed successfully.

#### Failure

The Powershell cmdlets were not executed successfully.

**Notes:**

- Only one of the **vm**, **template**, or **hardwareProfile** inputs must have a value assigned in the flow.
- More details on valid input formats and combinations for this flow can be found using the **Get-Help New-VirtualNetworkAdapter -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## Remove Virtual Network Adapter

The **Remove Virtual Network Adapter** flow runs the **Remove-VirtualNetworkAdapter** PowerShell command to remove a virtual network adapter used in Virtual Machine Manager.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### virtualNetworkAdapter

The ID of the virtual network adapter to remove.

### Results

The flow returns the following results:

#### returnResult

A table containing the details of the virtual network adapter object that was removed.

#### warnings

A list of warnings that the cmdlet produced. If the cmdlet was executed without warnings, this result is an empty string.

### Responses

The flow returns one of the following responses:

#### Success

The PowerShell cmdlets were executed successfully.

#### Failure

The PowerShell cmdlets were not executed successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Remove-VirtualNetworkAdapter -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

## Set Virtual Network Adapter

The **Set Virtual Network Adapter** flow runs the **Set-VirtualNetworkAdapter** PowerShell command to change the properties of a virtual network adapter used in Virtual Machine Manager.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### virtualNetworkAdapter

The ID of the virtual network adapter whose properties you want to change.

#### virtualNetwork

The ID or name of the virtual network.

#### mACAddressesSpoofingEnabled

Specifies whether you want to enable MAC address spoofing. The valid values are **true** and **false.**

#### networkLocation

The name of the network location.

#### networkTag

The name of the network tag to associate with the virtual network adapter.

#### noConnection

Specifies whether you want to disconnect the virtual network adapter from the virtual network. The valid values are **true** and **false.**

#### physicalAddress

The physical address (MAC address) of the virtual network adapter.

#### physicalAddressType

The type of physical address to use for the virtual network adapter. The valid values are **Static** and **Dynamic**.

### vlanEnabled

Specifies whether you want to enable virtual LAN for the virtual network adapter. The valid values are **true** and **false.**

### vlanId

The numerical VLAN identifier for the virtual network adapter. The valid values are **1**-**4094**.

### vmNetworkOptimizationEnabled

Specifies whether you want to enable virtual machine network optimization. The valid values are **true** and **false.**

### vmwarePortGroup

The name of the VMware port group.

## Results

The flow returns the following results:

### returnResult

A table containing the details of the virtual network adapter object that was changed.

### warnings

A list of warnings that the cmdlet produced. If the cmdlet was executed without warnings, this result is an empty string.

## Responses

The flow returns one of the following responses:

### Success

The Powershell cmdlets were executed successfully.

### Failure

The Powershell cmdlets were not executed successfully.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Set-VirtualNetworkAdapter -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.
- To avoid unwanted behavior caused by multiple objects sharing the same name, specify object IDs for the inputs that accept both names and IDs.

# Virtual SCSI Adapter

## Get Virtual SCSI Adapter

The **Get Virtual SCSI Adapter** flow runs the **Get-VirtualSCSIAdapter** Powershell command that gets details about the virtual SCSI adapters from a virtual machine, template, or hardware profile.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### vm

The name or identifier of the virtual machine whose virtual SCSI adapters are to be retrieved. This can be deployed on a virtual machine host or can be stored in the Virtual Machine Manager library. The name you specify must match the **Name** property of the virtual machine object exactly. To avoid unwanted behavior caused by duplicate names, specify the virtual machine identifier. The valid format for the virtual machine identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **MyVM** or **12345678-abcd-efgh-ijkl-1234567890**.

#### template

The name or identifier of the template whose virtual SCSI adapters are to be retrieved. The name you specify must match the **Name** property of the target template exactly. To avoid unwanted behavior caused by duplicate names, specify the template identifier. The valid format for the template identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **MyTemplate** or **12345678-abcd-efgh-ijkl-1234567890**.

#### hardwareProfile

The name of the hardware profile whose virtual SCSI adapters are to be retrieved.

#### all

Specifies whether to retrieve a list of virtual SCSI adapters regardless of their parent object. The valid values are **true** and **false**. If you do not specify a value, the flag is not set.

### Results

The flow returns the following results:

#### returnResult

A table containing a row for each virtual SCSI adapter retrieved or an exception message if the flow failed. The table's columns represent the properties of the SCSI adapter, in the format `propertyName<delimiter>propertyValue`. If there are no SCSI adapters that match the given criteria, the result will be empty.

#### warnings

A list of warnings produced by the PowerShell commands. If the commands were executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The flow executed successfully and the result contains the SCSI adapters that matched the given criteria.

#### Failure

The flow could not retrieve the list of SCSI adapters.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help Get-VirtualSCSIAdapter -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.

## New Virtual SCSI Adapter

The **New Virtual SCSI Adapter** flow runs the **New-VirtualSCSIAdapter** Powershell command to create a virtual SCSI adapter on a virtual machine, template or hardware profile.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### vm

The name or identifier of the virtual machine to which the new virtual SCSI adapter is to be attached. This can be deployed on a virtual machine host or can be stored in the Virtual Machine Manager library. The name you specify must match the **Name** property of the virtual machine object exactly. To avoid unwanted behavior caused by duplicate names, specify the virtual machine identifier. The valid format for the virtual machine identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **MyVM** or **12345678-abcd-efgh-ijkl-1234567890**.

#### template

The name or identifier of the template to which the new virtual SCSI adapter is to be attached. The name you specify must match the **Name** property of the target template exactly. To avoid unwanted behavior caused by duplicate names, specify the template identifier. The valid format for the template identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **MyTemplate** or **12345678-abcd-efgh-ijkl-1234567890**.

#### hardwareProfile

The name of the hardware profile to which the new virtual SCSI adapter is to be attached.

### Results

The flow returns the following results:

#### returnResult

A table containing a row with details about the new SCSI adapter or an exception message if the flow failed. The table's columns represent the properties of the SCSI adapter, in the format `propertyName<delimiter>propertyValue`.

#### warnings

A list of warnings produced by the PowerShell commands. If the commands were executed without warnings, this result is empty.

### Responses

The flow returns one of the following responses:

#### Success

The flow executed successfully and the result contains details about the new virtual SCSI adapter.

#### Failure

The flow could not create the virtual SCSI adapter.

**Notes:**

- More details on valid input formats and combinations for this flow can be found using the **Get-Help New-VirtualSCSIAdapter -Detailed** PowerShell command and inspecting the **SYNTAX** and **PARAMETERS** sections in the resulting output.
- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.

## Remove Virtual SCSI Adapter

The **Remove Virtual SCSI Adapter** flow runs the **Remove-VirtualSCSIAdapter** PowerShell command to remove a virtual SCSI adapter from a virtual machine, template, or hardware profile.

### Inputs

All of the flow's inputs except the following are described in *Common Inputs in the Integration*.

#### virtualScsiAdapter

The identifier of the virtual SCSI adapter to be removed. The valid format for the identifier is **xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx**. For example, **12345678-abcd-efgh-ijkl-1234567890**.

### Results

The flow returns the following results:

### returnResult

A table containing a row with details about the removed virtual SCSI adapter or an exception message if the flow failed. The table's columns represent the properties of the virtual SCSI adapter, in the format `propertyName<delimiter>propertyValue`.

### warnings

A list of warnings produced by the PowerShell commands, If the commands were executed without warnings, this result is empty.

## Responses

The flow returns one of the following responses:

### Success

The flow executed successfully and the result contains details about the virtual SCSI adapter.

### Failure

The flow could not remove the virtual SCSI adapter.

**Notes:**

- PowerShell related requirements and settings needed to run this flow are listed in the description of the Library/Integrations/Microsoft/System Center Virtual Machine Manager/ folder.

# 4 Launching Integration Flows Using Other Tools

This section includes the following topics:

# Ways of Launching Integration Flows Using Other Tools

You can launch an integration flow from other tools if it has a console or interface that can accept a command. In order to launch an integration flow, you must obtain its URL.

**To obtain a flow's URL**

1. Open OO Central.
2. Click the **Flow Library** tab.
3. Click the flow in the **Flow Library** tab, and then copy the **Guided Run** or **Run All** URL in the **Execution Links** pane.

There are a number of ways to launch a flow using other tools:

- Use the REST service to launch a flow from a command line. The REST command-line tools you can use are:
  - Wget
  - RSFlowInvoke.exe or the Java version JRSFlowInvoke.jar
- Use the WSCentralService SOAP API to access Central features programmatically.

For instructions on using these methods, see the OO *Software Development Kit Guide* (SDKGuide.pdf) in the documentation set for the current OO release.

## Using Wget

Wget is a command-line tool that you can use to download and run flows from the Internet. You can download Wget from the **GNU Wget** Web page. Wget runs the flow you specify in the URL contained on the command line. It can use the HTTP, HTTPS, and FTP protocols.

## Using RSFlowInvoke or JRSFlowInvoke

RSFlowInvoke (RSFlowInvoke.exe) or the Java version JRSFlowInvoke (JRSFlowInvoke.jar) is a command-line utility that allows you to start a flow without using OO Central (although the Central service must be running). These tools are available in the Operations Orchestration home directory under the Studio\tools\ folder.

## Using the WSCentralService SOAP API

Use the WSCentralService SOAP API to launch and control the execution of integration flows programmatically. This allows you to control the flow execution—including running, pausing, resuming, and canceling a flow, and viewing the status of a flow run.

The WSCentralService SOAP API Java and .NET classes and interfaces are located in the OO SDK home directory, in the lib/ folder. The certificates, keystore, WSDL, and sample code are located in the OO SDK home directory, in the samples/ folder.

# 5 Creating Custom SCVMM Integration Operations

This section includes the following topics:

# Ways of Creating SCVMM Integration Operations

If you need to perform tasks that are not currently supported by the SCVMM integration's operations, you can use OO IActions to create new operations. This requires that you have knowledge of the SCVMM API.

## Using OO IActions to Create OO Operations

An IAction is the code that implements an OO operation through a RAS (Remote Action Service). To create custom operations using IActions, you need knowledge of the SCVMM API as well as Java or a .NET language.

For information on creating custom IActions, see the *OO Software Development Kit Guide* (SDKGuide.pdf) in the documentation set for the current OO release.

# 6 Troubleshooting

This section includes the following topics:

# Troubleshooting Overview

This section provides troubleshooting procedures and tools that you can use to solve problems you may encounter while using this integration. It also includes a list of the error messages you may receive while using the integration and offers descriptions and possible fixes for the errors.

# General Troubleshooting Procedures and Tools

If a Microsoft SCVMM operation or flow fails unexpectedly, verify that you have properly applied the settings described in the *Installing and Configuring the Integration* section in this document on the RAS host and on the SCVMM server host. Also check that the credential inputs provided in the flow are correct.

You can also verify that the SCVMM task you wanted to perform succeeds when executed in PowerShell by following the below steps:

1. Open the PowerShell console on the RAS host.
2. Create and use a new PowerShell session on the SCVMM server with the following commands:

   ```
   $s = New-PSSession <SCVMM_server_ip_or_hostname> -Credential <SCVMM_domain>\<SCVMM_user>
   ```

   ```
   Enter-PSSession $s
   ```
3. Load the SCVMM PowerShell snap-in with the command below:

   ```
   Add-PSSnapin Microsoft.SystemCenter.VirtualMachineManager
   ```
4. Connect to the Virtual Machine Manager server:

   ```
   Get-VMMServer <SCVMM_server_ip_or_hostname>
   ```
5. Once you are in the session and connected to the server, run the command that the OO flow or operation tried to execute and inspect the output.

# Error Messages

This section lists the error messages you may receive while using this integration. Each error message includes possible causes and fixes for the error.

The following errors are thrown by the SCVMM content and indicate problems that prevent the execution of the PowerShell cmdlets which are intended to carry out the SCVMM tasks:

### Can not add snap-in to the current powershell session.

The provided snap-in could not be loaded in the current Powershell session. A needed library might be missing.

### Can not connect to host.

A connection to the host could not be established. Possible reasons might be: the host is unreachable, Powershell is not installed etc.

### Can not execute powershell command.

The snap-in containing the needed command is not loaded or syntax of the command is not valid.

### You must provide a host in order to execute the operation.

All the flows need a host input to be provided in order to be executed.

### Invalid value for authentication type.

Only Default, Kerberos, CredSSP and Negotiate authentication types are supported.

There are also a number of SCVMM specific errors that can occur when the OO SCVMM flows executes SCVMM cmdlets via PowerShell. An extensive list containing these errors and fixes is available at *http://social.technet.microsoft.com/wiki/contents/articles/virtual-machine-manager-vmm-2008-r2-error-codes.aspx*.

# 7 Security

This section includes the following topics:

# About SCVMM Security

The OO integration flows and operation create a remote Windows PowerShell session on the target SCVMM computer. In order to successfully execute the commands, you must provide each user with credentials that have enough privileges and an authentication mechanism to authenticate the user on the remote host.

The operation and flows supports the following authentication mechanisms:

- Default
- Negotiate
- Kerberos
- Credssp

**Default** and **Negotiate** work even if the RAS and SCVMM hosts are not in the same domain with default configuration if the given user has enough privileges to run PowerShell commands.

For **Kerberos** authentication, both the RAS and SCVMM machine should be in the same domain and should be able to access the domain controller.

The **Credential Security Support Provider** protocol (**CredSSP**) lets an application delegate the user's credentials from the client to the target server for remote authentication. The client is authenticated over the encrypted channel by using the Simple and Protected Negotiate (SPNEGO) protocol with either Microsoft Kerberos or Microsoft NTLM.

### To enable CredSSP authentication

1. Server side steps:
   a. Set CredSSP authentication to **true** by using the **Enable-WSManCredSSP -Role Server** PowerShell command, or by using `the` **`winrm set winrm/config/client/auth '@{CredSSP="true"}'`** command line alternative.
   b. Create a new HTTPS listener by using the following command:

      **`winrm create winrm/config/Listener?Address=+Transport=HTTPS`**

      If you encounter the following error `Cannot create a WinRM listener on HTTPS because this machine does not have an appropriate certificate. To be used for SSL, a certificate must have a CN matching the hostname, be appropriate for Server Authentication, and not be expired, revoked, or self-signed.`, and all the properties are set correctly, you can try to give the exact path to your valid certificate using:

      **`winrm create winrm/config/Listener?Address=+Transport=HTTPS @{Hostname="myhost";CertificateThumbprint="97ec43f84ec447f0d88de55bcf91227069680d2e"`**

      where **`myhost`** is your actual hostname. **`CertificateThumbprint`** can be retrieved following these steps:

      i. Open the Microsoft Management Console (MMC) snap-in for certificates.
      ii. In the Console Root window's left pane, click **Certificates (Local Computer)**.
      iii. Click the **Personal** -> **Certificates** folders to expand them.

iv. In the list of certificates, note the **Intended Purposes** heading. Find a certificate that lists **Server Authentication** as an intended purpose and double-click the certificate.

v. In the **Certificate** dialog box, click the **Details** tab, scroll through the list of fields, and then click **Thumbprint**. Copy the hexadecimal characters from the box. If this thumbprint is used in code for the X509FindType, remove the spaces between the hexadecimal numbers. For example, the thumbprint **a9 09 50 2d d8 2a e4 14 33 e6 f8 38 86 b0 0d 42 77 a3 2a 7b** should be specified as **a909502dd82ae41433e6f83886b00d4277a32a7b** in code.

2 Client side steps:

a Enable CredSSP authentication on the client, by running the command **`Enable-WSManCredSSP -Role Client -DelegateComputer WSMAN/*`**.

c Allow delegating fresh credentials by performing the following steps:

i. Open **gpedit.msc** and go to **Computer Configuration** -> **Administrative Templates** -> **System** -> **Credentials Delegation**.

ii. Enable **Allow Delegating Fresh Credentials** and add the **wsman** hosts to the server list.

iii. Run **`gpupdate /force`** from command line to force policy update.

d Domain Controller side steps:

– If the NETWORK SERVICE doesn't have "Validated write to service principal name", try running the following command: **`dsacls "CN=AdminSDHolder,CN=System,DC=domain,DC=com" /G "S-1-5-20:WS;Validated write to service principal name"`** or open ADUC, go to **Computers** -> **DC object** -> **Security**, select **Network Service**, and then give it "**Validated write to SPN**".