

# HP Operations Orchestration

for the Windows and Linux operating systems

Software Version: OO Content Pack 7

---

## VMware Virtual Infrastructure and vSphere Integration Guide

Document Release Date: January 2012

Software Release Date: January 2012



## 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

© Copyright 2009-2011 Hewlett-Packard Development Company, L.P.

### 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](http://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](http://h20230.www2.hp.com/new_access_levels.jsp)**

# Contents

<b>1</b>	<b>Introduction .....</b>	<b>9</b>
	Purpose of the VMware Virtual Infrastructure and vSphere Integration.....	10
	Supported Versions .....	10
	Downloading OO Releases and Documents on HP Live Network.....	10
<b>2</b>	<b>Getting Started with the VMware Virtual Infrastructure and vSphere Integration</b>	<b>12</b>
	Installing and Configuring the Integration .....	13
	VMware Virtual Infrastructure and vSphere — OO Integration Architecture .....	13
	VMware Virtual Infrastructure and vSphere Use Cases.....	13
<b>3</b>	<b>Using the VMware Virtual Infrastructure – OO Integration.....</b>	<b>18</b>
	Location of VMware Virtual Infrastructure Integration Operations and Flows in OO Studio.....	19
	Common Inputs in the Integration.....	20
	Descriptions of VMware Virtual Infrastructure Integration Operations and Flows .....	21
	Cluster .....	22
	Get Hosts in Cluster .....	22
	Get VM Port Groups from Cluster .....	23
	Get VMs in Cluster.....	23
	Move Host Into Cluster .....	24
	Remove Host from Cluster .....	25
	Datastore .....	25
	Copy DS Path.....	25
	Create DS Directory .....	26
	Delete DS Path .....	27
	Download DS Directory.....	27
	Download DS Directory Multi Thread .....	28
	Download DS File .....	28
	FS Get Children Enhanced .....	29
	Generate DS Url .....	30
	Get DS Children .....	30
	Is DS Directory .....	31
	Move DS Path .....	31
	Upload DS Directory .....	32
	Upload DS Directory Multi Thread .....	33
	Upload DS File.....	34
	Guest.....	34
	Apply Guest Customization Spec to VM .....	34
	Export Guest Customization Spec.....	36
	Get Guest Info.....	37
	Import Guest Customization Spec.....	39
	Mount Tools .....	40
	Reboot Guest .....	41

Shutdown Guest .....	41
Standby Guest .....	42
Upgrade Tools .....	42
VM Tools Running Status .....	43
VM Tools Version Status.....	44
Host .....	45
Add Host.....	45
Disconnect Host .....	45
Get VM Port Groups from Host .....	46
Reconnect Host .....	47
Remove Host .....	48
Set Maintenance Mode .....	49
Network .....	49
Add Distributed Port Group .....	49
Add Host to vNetwork Distributed Switch .....	50
Add Port Group.....	51
Add Virtual NIC .....	52
Add Virtual Switch .....	53
Add vNetwork Distributed Switch .....	53
Delete Distributed Port Group .....	54
Delete Network Entity .....	54
Delete vNetwork Distributed Switch .....	55
Edit Physical NIC Link .....	56
Edit Virtual NIC .....	56
Remove Host from vNetwork Distributed Switch .....	58
Power .....	59
Get Power State .....	59
Restart .....	59
Set Power State .....	60
Samples.....	61
Get Freespace On Datastores .....	61
Get List of Powered On VMs.....	61
Get VMs Powered On Longer Than By Event .....	62
Get VMs Powered On Longer Than By Task.....	63
Guided Search.....	64
Host Maintenance With No Downtime .....	65
Quick Migrate .....	65
Samples/Subflows.....	66
Get Powered On VM List As IDs .....	66
Snapshots.....	67
Create Snapshot .....	67
Delete Snapshot .....	68
Get Snapshot.....	69
Revert to Snapshot .....	69
Tasks .....	70
Cancel Task.....	70
Get All Running Tasks .....	71
Get Task State .....	71
Utilities .....	72
Advanced Search .....	72
Destroy Object.....	73
Get Latest Event.....	74
Get Latest Task .....	75
Get OS Descriptors .....	76

List Custom Fields.....	77
Simple Search .....	77
Virtual Apps/Samples .....	78
Import OVF to Cluster .....	78
Import OVF to Host.....	79
Virtual Apps/Samples/Subflows.....	80
Prompt For Each Key .....	80
Prompt For Each Key With Selection List.....	80
Virtual Apps .....	81
Export vApp as OVF.....	81
Get vApp.....	81
Get VMs in vApp.....	82
Import OVF.....	83
Parse OVF.....	85
Power Off vApp.....	85
Power On vApp .....	86
Shutdown vApp.....	87
Suspend vApp .....	87
Virtual Machines/Configuration .....	88
Add Existing Disk to VM .....	88
Add Floppy Drive To VM.....	89
Add New Disk To VM.....	90
Add NIC to VM .....	91
Add Optical Drive To VM.....	92
Configure NIC on VM.....	93
Edit Floppy Drive on VM .....	94
Edit Optical Drive on VM.....	96
Edit VM Boot Options .....	97
Edit VM CPUID Masks.....	98
Edit VM Power Options .....	100
Edit VM Settings .....	101
Edit VM Tools Options .....	103
Get Details of Disk on VM .....	104
Get Details of NIC on VM .....	105
Get Details of SCSI Controller on VM .....	107
Get VM CPU Count.....	108
Get VM Memory Size .....	108
Remove Disk From VM .....	109
Remove Floppy Drive From VM .....	110
Remove NIC From VM.....	110
Remove Optical Drive From VM.....	111
Set VM CPU Count.....	112
Set VM Memory Size .....	113
Virtual Machines.....	114
Clone Virtual Machine .....	114
Clone Virtual Machine to Template .....	115
Create Linked VM .....	116
Create Virtual Machine.....	117
Create VM Folder .....	118
Delete VM Folder.....	119
Deploy Virtual Machine from Template .....	119
Export VM as OVF .....	121
Get Virtual Machine.....	122
Mark VM As Template.....	123
Migrate Virtual Machine .....	124

Move Virtual Machine.....	125
Promote VM Disk .....	125
Register Virtual Machine Template.....	126
Register Virtual Machine.....	127
Relocate Virtual Machine.....	128
Rename VM Folder.....	129
Unregister Virtual Machine.....	130
VM is template.....	130
Hidden Inputs in the Integration .....	131
Get Task State.....	131
Create Virtual Machine .....	131
<b>4 Creating Custom VMware Virtual Infrastructure Integration.....</b>	<b>132</b>
Customizing the Integration.....	133
Using the Web Services Wizard to Create OO Operations for a Web Service .....	133
Using OO IActions to Create OO Operations.....	133
Using Third-Party Tools to Create OO Operations .....	133
<b>5 Troubleshooting.....</b>	<b>134</b>
Troubleshooting Overview .....	135
General Troubleshooting Procedures and Tools .....	135
Error Messages .....	135
<b>6 Security.....</b>	<b>139</b>
About VMware Virtual Infrastructure Security .....	140
<b>7 OO Tools .....</b>	<b>141</b>
OO Tools You Can Use with the VMware Virtual Infrastructure Integration .....	142



# 1 Introduction

This section includes the following topics:

- [Purpose of the VMware Virtual Infrastructure and vSphere Integration](#)
- [Supported Versions](#)
- [Downloading OO Releases and Documents on HP Live Network](#)

# Purpose of the VMware Virtual Infrastructure and vSphere Integration

With this integration, administrators can build HP Operations Orchestration (OO) flows that are integrated into the VMware Virtual Infrastructure (VI) and vSphere.

This document will explain how this integration has been implemented and how the operations included communicate between OO and VI.

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

## Supported Versions

**Table 1 Supported Versions**

Operations Orchestration Version	VMware Virtual Infrastructure Version	VMware vSphere Version
Content Pack 7	2.5, 3.0, 3.5	4.0, 4.1

## 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://hpln.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.

1. On the left-hand side, click **Operations Orchestration Content Packs**.
2. In the **Operations Orchestration Content Packs** box, click **Content**. The HP Passport and sign-in page appears.
3. Enter your user ID and Password to access to continue.
4. Click **HP Operations Orchestration 9.00**.

## 5. Search for HP Operations Orchestration Content Pack 7

## 2 Getting Started with the VMware Virtual Infrastructure and vSphere Integration

This section includes the following topics:

- [Installing and Configuring the Integration](#)
- [VMware Virtual Infrastructure and vSphere – OO Architecture](#)
- [VMware Virtual Infrastructure and vSphere Use Cases](#)

## Installing and Configuring the Integration

On a system that can access any Virtual Center or ESXi server, install a JRAS. If an ESXi server is being contacted, the ESXi server must be configured to enable its external Web service. To enable the Web service, see the “Modifying the Server Configuration to Support HTTP” section in the *VMware Developer’s Setup Guide* available on the VMware Web site.

## VMware Virtual Infrastructure and vSphere — OO Integration Architecture

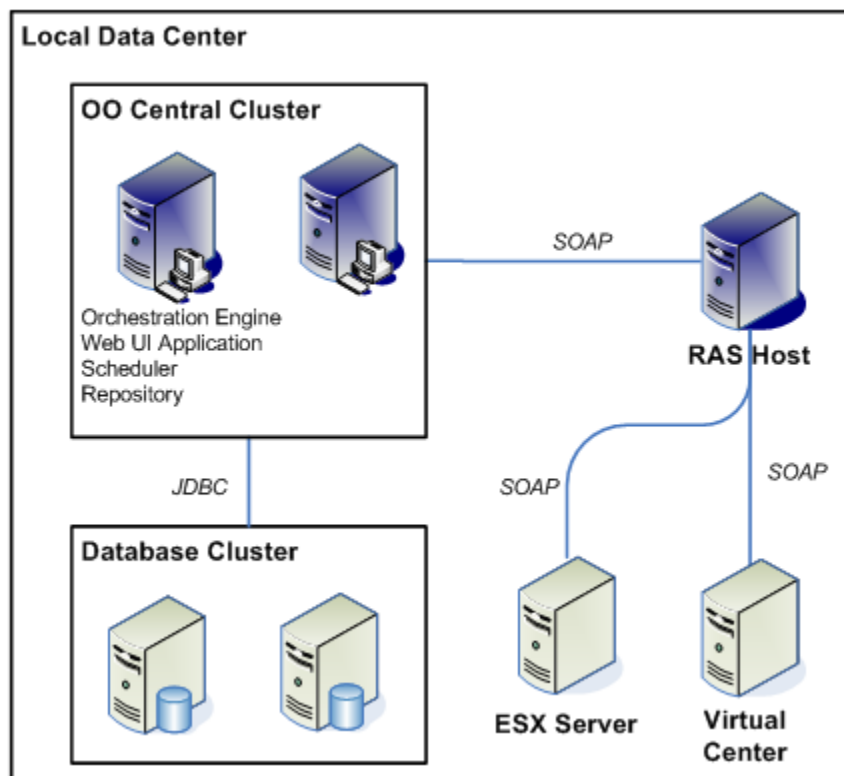


Figure 1 - VMware Virtual Infrastructure and vSphere - OO Architecture

## VMware Virtual Infrastructure and vSphere Use Cases

Following are the major use cases for the VMware Virtual Infrastructure and vSphere integration, and the operations and flows that you can use to implement them.

- 1 Manage cluster operations and flows:
  - Get Hosts in Cluster
  - Get VM Port Groups from Cluster
  - Get VMs in Cluster
  - Move Host Into Cluster

- Remove Host from Cluster
- 2 Manage datastores:
  - Copy DS Path
  - Create DS Directory
  - Delete DS Path
  - Download DS Directory
  - Download DS Directory Multi Thread
  - Download DS File
  - FS Get Children Enhanced
  - Generate DS Url
  - Get DS Children
  - Is DS Directory
  - Move DS Path
  - Upload DS Directory
  - Upload DS Directory Multi Thread
  - Upload DS File
- 3 Manage guest operating systems and VMware tools:
  - Apply Guest Customization Spec to VM
  - Export Guest Customization Spec
  - Get Guest Info
  - Import Guest Customization Spec
  - Mount Tools
  - Reboot Guest
  - Shutdown Guest
  - Standby Guest
  - Upgrade Tools
  - VM Tools Running Status
  - VM Tools Version Status
- 4 Manage ESX and ESXi hosts:
  - Add Host
  - Disconnect Host
  - Get VM Port Groups from Host
  - Reconnect Host
  - Remove Host
  - Set Maintenance Mode
- 5 Manage the network:

- Add Distributed Port Group
  - Add Host to vNetwork Distributed Switch
  - Add Port Group
  - Add Virtual NIC
  - Add Virtual Switch
  - Add vNetwork Distributed Switch
  - Delete Distributed Port Group
  - Delete Network Entity
  - Delete vNetwork Distributed Switch
  - Edit Physical NIC Link
  - Edit Virtual NIC
  - Remove Host from vNetwork Distributed Switch
- 6 Manage virtual machine power states:
- Get Power State
  - Restart
  - Set Power State
- 7 Sample flows:
- Get Powered On VM List As IDs
  - Get Freespace On Datastores
  - Get List of Powered On VMs
  - Get VMs Powered On Longer Than By Event
  - Get VMs Powered On Longer Than By Task
  - Guided Search
  - Host Maintenance With No Downtime
  - Quick Migrate
- 8 Manage virtual machine snapshots:
- Create Snapshot
  - Delete Snapshot
  - Get Snapshot
  - Revert To Snapshot
- 9 Manage tasks:
- Cancel Task
  - Get All Running Tasks
  - Get Task State
- 10 Search and filter elements in the VMware Infrastructure:
- Advanced Search
  - Destroy Object

- Get Latest Event
- Get Latest Task
- Get OS Descriptors
- List Custom Fields
- Simple Search

#### 11 Manage virtual applications:

- Prompt For Each Key
- Prompt For Each Key With Selection
- Import OVF to Cluster
- Import OVF to Host
- Export vApp as OVF
- Get vApp
- Get VMs in vApp
- Import OVF
- Parse OVF
- Power Off vApp
- Power On vApp
- Shutdown vApp
- Suspend vApp

#### 12 Manage virtual machine configuration and devices:

- Add Existing Disk To VM
- Add Floppy Drive To VM
- Add New Disk To VM
- Add NIC To VM
- Add Optical Drive To VM
- Configure NIC on VM
- Edit Floppy Drive on VM
- Edit Optical Drive on VM
- Edit VM Boot Options
- Edit VM CUID Masks
- Edit VM Power Options
- Edit VM Settings
- Edit VM Tools Options
- Get Details of Disk on VM
- Get Details of NIC on VM
- Get Details of SCSI Controller on VM



- Get VM CPU Count
- Get VM Memory Size
- Remove Disk From VM
- Remove Floppy Drive From VM
- Remove NIC From VM
- Remove Optical Drive From VM
- Set VM CPU Count
- Set VM Memory Size

### 13 Host system maintenance and administrative operations:

- Clone Virtual Machine
- Clone Virtual Machine to Template
- Create Linked VM
- Create Virtual Machine
- Create VM Folder
- Delete VM Folder
- Deploy Virtual Machine from Template
- Export VM as OVF
- Get Virtual Machine
- Mark VM As Template
- Migrate Virtual Machine
- Move Virtual Machine
- Promote VM Disk
- Register Virtual Machine
- Register Virtual Machine Template
- Relocate Virtual Machine
- Rename VM Folder
- Unregister Virtual Machine
- VM is Template

---

## 3 Using the VMware Virtual Infrastructure – OO Integration

This section includes the following topics:

- [Location of VMware Virtual Infrastructure Integration Operations and Flows in OO Studio](#)
- [Common Inputs in the Integration](#)
- [Descriptions of VMware Virtual Infrastructure Integration Operations and Flows](#)
- [Descriptions of Virtual Machines Integration Operations and Flows](#)
- [Hidden Inputs in the Integration](#)

# Location of VMware Virtual Infrastructure Integration Operations and Flows in OO Studio

The VMware Virtual Infrastructure integration includes the following operations and flows in the OO Studio Library/Integrations/VMware/VMware Virtual Infrastructure and vSphere/ folder.



Figure 2 - VMware Virtual Infrastructure and vSphere - OO Architecture

## 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 consistently throughout the integration's operations and flows.

### host

The VMware host name or IP address.

### user

The VMware username.

### password

The VMware user's password.

### port

The port on which to connect.

### protocol

The connection protocol. The valid values are **HTTP** and **HTTPS**.

### async

If you specify **false**, the operation waits for the underlying operation to complete before returning. If you specify **true**, the operation provides a task number for the user to monitor.

### closeSession

Specifies whether to close the cached VI session used by the operation at the completion of the step. All sessions are automatically cleaned at the end of a flow's execution, so it is usually not necessary to set this parameter to **true**. These sessions are unique to a **username**, **password**, **port**, and **host** within a flow. Unused sessions expire after a set amount of time in VI, so operations that take a long time to complete an operation should use the **async** option and poll intermittently to maintain an active session. Otherwise, the session may become invalid, and the OO integration does not automatically reconnect. In these situations it is expected that the user closes the session using any operation with **closeSession** set to **true** and attempting the operation again.

### vmIdentifierType

Specifies the way to identify a specific VM on which to perform the operation. The valid values are:

- **ip** — The IP address of the VM. Under certain conditions, the IP listed for the VM in Virtual Center may not be correct.
- **inventoryPath** — A variation of the inventory path viewable in Virtual Center in the **Virtual Machines and Templates** view. In Virtual Center, this usually means a string of the format **Datacenter/vm/sub/folder/path/virtual machine name**. Under ESXi, this usually means **ha-datacenter/vm/sub/folder/path/virtual machine**

**name**, where **ha-datacenter** is the default name of a datacenter for an ESXi server when directly accessed. Note that special characters such as \ and % are URI-escaped in Virtual Infrastructure and must be formatted with the escape sequence. Note, however, that (unlike in URIs) space is not escaped. If necessary, you can find how VI escaped a path by using VMware's Managed Object Browser (MOB). Please see [General Troubleshooting Procedures and Tools](#) to find out more about using MOB.

- **uuid** — Every VM has a UUID that you can access by searching on the property **config.uuid** with the **Advanced Search** operation. You can also find this UUID in the configuration file of the VM.
- **name** — The name of the VM as seen in the VI client. Because there may be multiple VMs with the same name in a datacenter, the VM that is chosen when multiple VMs exist with the same name is undefined. This value may need to be URI-escaped. To find the correct escaped name for the VM, see the discussion of the **inventoryPath** value above.
- **hostname** — The fully-qualified domain name of the host as seen in the VI client.
- **vmid** — The internal ID of the VM within the VI SDK that differs between ESXi and VC. This is in the form **vm-number** in Virtual Center and **number** in ESXi.

#### virtualMachine

The primary virtual machine identifier. The format of this the value depends on the value of the **vmIdentifierType** input:

- **inventoryPath** — The inventory path in the format **Datacenter/vm/Folder/MyVM**.
- **name** — The name of VM.
- **ip** — IPv4 or IPv6 depending upon the ESXi version.
- **hostname** — The fully-qualified domain name of the host.
- **uuid** — The UUID.
- **vmid** — The VM ID in the format **vm-123,123**.

#### resourcePool

The name of the resource pool assigned to a VM. If you do not specify a resource pool, the default name is **Resources**. This input can be empty.

#### vmResourcePool

Similar to **resourcePool**, this input is used for instantiating a new VM and follows the same conventions as **resourcePool**.

## Descriptions of VMware Virtual Infrastructure Integration Operations and Flows

This section describes the VMware Virtual Infrastructure integration's flows and operations, including any flow- and operation-specific inputs. The operations are grouped by their basic functionality:

- Cluster
- Datastore

- Guest
- Host
- Network
- Power
- Samples
- Snapshots
- Tasks
- Utilities
- Virtual Apps
- Virtual Machines

The sample flows in the OO Library/Integrations/VMware/VMware Virtual Infrastructure and vSphere/Samples/ folder perform some of the most common tasks that need to be automated when using VMware Virtual Infrastructure, such as getting free space on data stores, getting a list of powered on VMs, and performing a quick migrate. Each of these sample flows describes in detail what it does. You can use these flows as they are or as templates for new operations.

## Cluster

### Get Hosts in Cluster

The **Hosts in Cluster** flow returns a list of ESX or ESXi host systems in a cluster.

#### Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#).

##### clusterName

The name of the cluster for which to list the host systems.

##### delimiter

The delimiter to separate values in the **hostid** and **name** result lists.

For example, |.

#### Results

The flow returns the following results:

##### count

The number of host systems found in the cluster.

hostid

The list of host system host IDs found in the cluster delimited by the value specified in the **delimiter** input.

name

The list of host system names found in the cluster delimited by the value specified in the **delimiter** input.

## Get VM Port Groups from Cluster

Returns the names of all the port groups available to virtual machines on a cluster of host systems (ESX or ESXi) as a JSON array.

### Inputs

datacenter

Datacenter of the host system.

clusterName

Name of the VMWare HA or DRS cluster.

### Results

returnResult

The names of all the port groups as a JSON array. Reason for error on failure.

netLegacyJS

The names of the legacy port groups as a JSON array.

netDistributedJS

The names of the distributed virtual port groups as a JSON array.

## Get VMs in Cluster

The **Get VMs in Cluster** flow returns a list of virtual machines (VMs) in a cluster.

### Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#).

clusterName

The name of the cluster for which to list the virtual machines.

delimiter

The delimiter to separate the values in the **vmid** and **name** result lists.

For example, |.

## Results

The flow returns the following results:

### `count`

The number of virtual machines found in the cluster.

### `vmid`

The list of virtual machine IDs found in the cluster delimited by the value specified in the **delimiter** input.

### `name`

List of virtual machine names found in the cluster delimited by the value specified in the **delimiter** input.

## Move Host Into Cluster

Move a standalone host into a cluster. The host must be present in the vCenter inventory, and be in the same datacenter as the cluster to which it will be added.

## Inputs

### `taskTimeOut`

Time to wait before the operation is considered to have failed (seconds).

### `datacenter`

Datacenter of the host system and cluster.

### `hsIdentifierType`

Host system identifier type.

### `hostSystem`

Host system identifier based on the `hsIdentifierType`.

### `clusterName`

Name of the cluster to which the host should be added.

### `resourcePool`

Name of the resource pool that the new host's resources will be placed in. If not supplied the resources from the new host will be placed into the default resource pool of the cluster.

## Results

### `returnResult`

Operation result



## Remove Host from Cluster

Remove a host from the cluster which it belongs to, and returns it to a standalone host in the datacenter which contains the cluster.

### Inputs

#### `taskTimeOut`

Time to wait before the operation is considered to have failed (seconds).

#### `datacenter`

Datacenter of the host system.

#### `hsIdentifierType`

Host system identifier type.

#### `hostSystem`

Host system identifier based on the `hsIdentifierType`.

### Results

#### `returnResult`

Operation result

## Datastore

### Copy DS Path

The **Copy DS Path** operation copies the source file or folder to the destination.

### Inputs

All of the operation's inputs except the following are described in [Common inputs in the integration](#).

#### `srcDatacenter`

The source datacenter of the file or folder from which to copy.

#### `dstDatacenter`

The destination datacenter to which to copy the file or folder. If you do not specify a value for this input, the destination datacenter will be same as the source datacenter.

#### `srcDatastore`

The source datastore of the file or folder to copy from the source datacenter.

#### `dstDatastore`

The destination datastore of the file or folder from the destination datacenter. If you do not specify a value for this input, the destination datastore will be same as the source datastore.

#### srcPath

The source path of the file or folder to copy from the source datastore.

For example, **path/to/file(folder)**.

#### dstPath

The destination path of the file or folder from the destination datastore. The new file or folder name must be included at the end of the destination path.

For example, **path/to/ new file(folder)**.

#### Overwrite

If you specify a value of **false** for this input, the operation fails if a file with the same name already exists in the destination datastore path.

The valid values are **true** and **false**.

The default value is **true**.

#### Results

The operation returns the following:

#### returnResult

This is the primary output, which returns a success or failure message.

### Create DS Directory

The **Create DS Directory** operation creates the specified folder. If the intermediate level folders do not exist, all the non-existent folders are created

#### Inputs

All of the operation's inputs except the following are described in [Common inputs in the integration](#).

#### datacenter

The datacenter of the folder.

#### dataStore

The datastore of the folder.

#### path

The path of the folder to create.

For example, **path/to/folder to be created**.

#### Results

The operation returns the following:

`returnResult`

This is the primary output, which returns a success or failure message.

## Delete DS Path

The **Delete DS Path** operation deletes the specified file or folder from the datastore.

### Inputs

All of the operation's inputs except the following are described in *Common inputs in the integration*.

`datacenter`

The datacenter of the file or folder to delete.

`dataStore`

The datastore of the file or folder to delete.

`path`

The path of the file or folder from the root of the datastore.

For example, **path/to/file(folder)**.

### Results

The operation returns the following:

`returnResult`

This is the primary output, which returns a success or failure message.

## Download DS Directory

The **Download DS Directory** flow downloads files or folders from the datastore path to the specified local path.

### Inputs

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

`timeout`

The timeout (in milliseconds) after which the connection should be terminated. Specify a value of 0 (zero) for no timeout.

`datacenter`

The datacenter of the file or folder to download.

`dataStore`

The datastore of the file or folder to download.

#### path

The path of the file or folder from the root of the datastore.

For example, **path/to/file(folder)**.

#### localPath

The path to the folder where the file or folder from the datastore path should be downloaded.

## Download DS Directory Multi Thread

The **Download DS Directory Multi Thread** flow downloads files or folders from the datastore path to the specified local path. The **Download DS** step is multi-instance and throttled to four threads.

### Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

#### timeout

The timeout (in milliseconds) after which the connection should be terminated. Specify a value of 0 (zero) for no timeout.

#### datacenter

The datacenter of the file or folder to download.

#### dataStore

The datastore of the file or folder to download.

#### path

The path of the file or folder from the root of the datastore.

For example, **path/to/file(folder)**.

#### localPath

The path to the folder where the file or folder from the datastore path should be downloaded.

## Download DS File

The **Download DS File** flow downloads a file from the datastore and saves it to the specified file.

### Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

#### url

The datastore URL from which to download.

The valid format is

**scheme://authority/folder/path?dcPath=dcPath&dsName=dsName.**



In the format **scheme://authority/folder/path?dcPath=dcPath&dsName=dsName:**

- **scheme** is **HTTP** or **HTTPS**.
- **authority** specifies the hostname or IP address of the VirtualCenter or ESX server and, optionally the port.
- **dcPath** is the inventory path to the datacenter containing the datastore.
- **dsName** is the name of the datastore.
- **path** is a slash-delimited path from the root of the datastore.

#### file

The file name with the absolute file path in which to store the data read from the specified URL. The file is overwritten if the file already exists.

#### username

The username to use for basic authentication to the URL.

#### password

The password to use for basic authentication to the URL.

#### timeout

The timeout (in milliseconds) after which the connection should be terminated. Specify a value of **0** (zero) for no timeout.

## FS Get Children Enhanced

The **FS Get Children Enhanced** flow gets a list of files and folders that reside in a directory. It does not get the list of files and folders from the subdirectories within this directory.

### Inputs

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

#### source

The directory from which to get the children.

#### delimiter

A delimiter to put between each child in this directory.

### Results

The flow returns the following:

#### returnResult

The list of paths to each child in this directory.

files

The list of paths to each file in this directory.

folders

The list of paths to each folder in this directory.

## Generate DS Url

The **Generate DS Url** operation generates a datastore URL.

### Inputs

All of the operation's inputs except the following are described in [Common inputs in the integration](#).

datacenter

The datacenter of the path.

dataStore

The datastore of the path.

path

The path from the root of the datastore. For example, **path/to/folder(file)**.

### Results

The operation returns the following:

returnResult

This is the primary output, which returns the URL of the datastore path.

## Get DS Children

The **Get DS Children** operation gets a list of files and folders in a directory. It does not get the list of files and folders from the subdirectories within this directory.

### Inputs

All of the operation's inputs except the following are described in [Common inputs in the integration](#).

datacenter

The datacenter of the folder.

dataStore

The datastore of the folder.

`path`

The path of the folder to check.

For example, **path/to/folder**.

## Results

The operation returns the following:

`returnResult`

This is the primary output, which returns the children of the folder.

`files`

The files in the folder.

`folders`

The folders in the folder.

## Is DS Directory

The **Is DS Directory** operation checks to see if a path points to a directory.

## Inputs

All of the operation's inputs except the following are described in [Common inputs in the integration](#).

`datacenter`

The datacenter of the path.

`dataStore`

The datastore of the path.

`path`

The path to check.

For example, **path/to/destination**.

## Results

The operation returns the following:

`returnResult`

This is the primary output, which returns a success or failure message.

## Move DS Path

The **Moves DS Path** operation moves the source file or folder to the destination.

## Inputs

All of the operation's inputs except the following are described in *Common inputs in the integration*.

### srcDatacenter

The source datacenter of the file or folder from which to move.

### dstDatacenter

The destination datacenter of the file or folder to which to move. If you do not specify a value for this input, the destination datacenter will be same as the source datacenter.

### srcDatastore

The source datastore of the file or folder to move from the source datacenter.

### dstDatastore

The destination datastore of the file or folder from the destination datacenter. If you do not specify a value for this input, the destination datastore will be same as the source datastore.

### srcPath

The source path of the file or folder to move from the source datastore.

For example, **path/to/file(folder)**.

### dstPath

The destination path of the file or folder from the destination datastore. If you do not specify the file or folder name to move to, it takes the same file or folder name from the source datastore.

For example, **path/to/file(folder)**.

### overwrite

If you specify a value of **false** for this input, the operation will fail if a file with the same name already exists in the destination datastore path.

The valid values are **true** and **false**.

The default value is **true**.

## Results

The operation returns the following:

### returnResult

This is the primary output, which returns a success or failure message.

## Upload DS Directory

The **Upload DS Directory** flow uploads files or folders to the datastore path from the specified local path.



## Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

### timeout

A timeout (in milliseconds) after which the connection should be terminated, or **0** (zero) for no timeout.

### datacenter

The datacenter of the file or folder to upload.

### dataStore

The datastore of the file or folder to upload.

### path

The path of the file or folder from the root of the datastore.

For example, **path/to/file(folder)**.

### localPath

The path to the folder from where the file or folder should be uploaded to the datastore path.

## Upload DS Directory Multi Thread

The **Upload DS Directory Multi Thread** flow uploads files or folders to the datastore path from the specified local path. The **Upload DS** step is multi-instance and throttled to four threads.

## Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

### timeout

The timeout (in milliseconds) after which the connection should be terminated. Specify **0** (zero) for no timeout.

### datacenter

The datacenter of the file or folder to upload.

### dataStore

The datastore of the file or folder to upload.

### path

The path of the file or folder from the root of the datastore.

For example, **path/to/file(folder)**.

### localPath

The path to the folder from where the file or folder should be uploaded to the datastore path.

## Upload DS File

The **Upload DS File** flow uploads a file to the datastore from the specified file.

### Inputs

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

#### url

The datastore URL to which to upload.

The valid format is

**scheme**://**authority**/**folder**/**path**?**dcPath**=**dcPath**&**dsName**=**dsName**.

In the format **scheme**://**authority**/**folder**/**path**?**dcPath**=**dcPath**&**dsName**=**dsName**:

- **scheme** is **HTTP** or **HTTPS**.
- **authority** specifies the hostname or IP address of the VirtualCenter or ESX server and, optionally the port.
- **dcPath** is the inventory path to the datacenter containing the datastore.
- **dsName** is the name of the datastore.
- **path** is a slash-delimited path from the root of the datastore.

#### file

A file to upload to the specified URL.

#### username

The username to use for basic authentication to the URL.

#### password

The password to use for basic authentication to the URL.

#### timeout

The timeout (in milliseconds) after which the connection should be terminated. Specify a value of **0** (zero) for no timeout.

## Guest

### Apply Guest Customization Spec to VM

The **Apply Guest Customization Spec to VM** operation applies a guest customization spec to an existing VM.

## Inputs

### host

VMWare vCenter hostname or IP.

### user

VMWare username.

### password

VMWare user's password.

### port

Port on which to connect

### protocol

Connection protocol. The valid values are **HTTP** and **HTTPS**.

### closeSession

Close the internally kept VMWare Infrastructure API session upon completion of operation. The valid values are **true** and **false**.

### async

Asynchronously perform the task. The valid values are **true** and **false**.

### taskTimeout

Time to wait before the operation is considered to have failed (seconds).

### vmIdentifierType

Virtual machine identifier type (inventorypath, name, ip, hostname, uuid, vmid).

### virtualMachine

Primary Virtual Machine identifier. Inventorypath (Datacenter/vm/Folder/MyVM), Name of VM, IP (IPv4 or IPv6 depending upon ESX version), hostname (full), UUID, or the VM id (vm-123,123).

### vmDatacenter

Virtual machine's datacenter.

### customizationTemplateName

Name of the customization specification to apply to the virtual machine. The customization specification should already exist in the vCenter customization specifications manager. This is not a required input. However, if both 'customizationTemplateName' and 'customizationSpecXml' are empty or have input, the operation will give an error message and fail. Therefore, either 'customizationTemplateName' or 'customizationSpecXml' should have an input.

### customizationSpecXml

The xml string of the customization spec to apply. This is not a required input.

## Responses

The operation's responses are:

### success

The operation completed successfully.

### failure

Something went wrong.

## Returns

The operation returns the following:

### returnResult

Operation result message.

Notes:

1. Inventory Path Formatting. If host is an ESX server inventory path will be: ha-datacenter/vm/<name of vm>. If host is a vCenter the inventory path will be: <name of datacenter>/vm/<folders>/<name of vm>. The <folders>/<name of vm> part of the path is based on the "Virtual Machines & Templates" view in the vCenter client. The inventory path is case sensitive.
2. Guest customization spec can be applied only to a vm.
3. If the operation fails due to "A customization operation is pending on the vm:", then restart the guest of the vm before applying a different customization specification.
4. The result of the VMware/VMware Virtual Infrastructure and vSphere/Guest/Export Guest Customization Spec operation can be used to get the customizationSpecXml input value.
5. If Microsoft Sysprep Tools for a particular Windows guest operating system is not already bundled with the vCenter Server, then it must be installed on the vCenter Server system. For instructions on Guest Operating System Customization Requirements see the section "Customizing Guest Operating Systems" at [http://www.vmware.com/pdf/vsphere4/r41/vsp\\_41\\_vm\\_admin\\_guide.pdf](http://www.vmware.com/pdf/vsphere4/r41/vsp_41_vm_admin_guide.pdf).

## Export Guest Customization Spec

Export a guest customization spec to customization spec XML string. The exported customization spec xml string can be modified and imported to the vCenter Customization Specifications Manager using the **Import Guest Customization Spec** operation.

## Inputs

### host

VMWare vCenter hostname or IP.

`user`

VMWare username.

`password`

VMWare user's password.

`port`

Port on which to connect

`protocol`

Connection protocol. The valid values are **HTTP** and **HTTPS**.

`closeSession`

Close the internally kept VMWare Infrastructure API session upon completion of operation. The valid values are **true** and **false**.

`customizationTemplateName`

Name of the customization specification to export. The customization specification should already exist in the vCenter customization specifications manager.

## Responses

The operation's responses are:

`success`

The operation completed successfully.

`failure`

Something went wrong.

## Returns

The operation returns the following:

`returnResult`

Returns the customization spec Xml string or the failure message.

Note:

The `returnResult` of this operation can be used for the 'customizationSpecXml' input in the **Clone Virtual Machine** operation, 'Clone Virtual Machine to Template' flow and 'Deploy Virtual Machine from Template' flow.

## Get Guest Info

The **Get Guest Info** operation retrieves information about a guest operating system running on a virtual machine. VM tools must be running on the guest to in order to retrieve this information.

## Inputs

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

### vmDatacenter

The virtual machine's datacenter.

## Results

The operation returns the following:

### returnResult

This is the primary output, containing a descriptive result of the operation or a reason for the error if the operation fails.

### state

The running state of the guest operating system.

Possible values are **running**, **shuttingdown**, **resetting**, **standby**, **notrunning**, and **unknown**.

### hostname

The hostname of the guest according to the guest operating system.

### ipAddress

The primary IP address of the guest according to the guest operating system.

For example, **10.11.22.33**.

### guestOsId

The guest operating system identifier (short name).

For example, **windows7Server64Guest**.

### guestOsDescription

The guest operating system's full name.

For example, **Microsoft Windows Server 2008 R2 (64-bit)**.

### guestFamily

The guest operating system family.

For example, **WindowsGuest**.

### guestNicInfo

A Json array of Json objects that describes the network interfaces of the guest. Each Json object has three keys:

- **macaddress** – The MAC address of the guest NIC (for example, **00:50:56:87:01:93**).
- **ipAddresses** - A JSON array of IP addresses assigned to the NIC (for example, **["fe80::9802:e33:4e2:2caf%15","16.93.30.83"]**).

- **network** - The name of the network port group to which the NIC is attached.

#### guestDiskInfo

A JSONArray of JSON objects that describes the disks attached to the guest. Each Json object has three keys:

- **diskPath** - The mount point of the disk on the guest (for example, "C:\").
- **capacity** – The size of the disk in bytes.
- **freespace** – The amount of freespace on the disk in bytes.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter client.
- The inventory path is case sensitive.

## Import Guest Customization Spec

Import a guest customization spec from a customization spec XML string. The imported guest customization spec can be viewed in the Home/Management/Customization Specifications Manager in the vCenter.

### Inputs

#### host

VMWare vCenter hostname or IP.

#### user

VMWare username.

#### password

VMWare user's password.

#### port

Port on which to connect

#### protocol

Connection protocol. The valid values are **HTTP** and **HTTPS**.

#### closeSession

Close the internally kept VMWare Infrastructure API session upon completion of operation. The valid values are **true** and **false**.

#### customizationSpecXml

The XML string of the customization spec to import.

#### name

The unique name of the guest customization specification. If not specified, then uses the same name existing in the customization spec XML string.

#### description

The description of the customization specification.

### Responses

The operation's responses are:

#### success

The operation completed successfully.

#### failure

Something went wrong.

### Returns

The operation returns the following:

#### returnResult

Operation result message.

## Mount Tools

The **Mount Tools** operation mounts the VMware Tools installer image in the guest virtual machine.

### Inputs

All of the flow's inputs are described in *Common inputs in the integration*.

### Results

The operation returns the following:

#### returnResult

The operation's result message.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or



vSphere client.

- The inventory path is case sensitive.

## Reboot Guest

The **Reboot Guest** operation attempts to reboot the specified guest virtual machine.

### Inputs

All of the flow's inputs are described in *Common inputs in the integration*.

### Results

The operation returns the following:

#### returnResult

The operation's result message.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Shutdown Guest

The **Shutdown Guest** operation attempts to shutdown the specified guest virtual machine.

### Inputs

All of the flow's inputs are described in *Common inputs in the integration*.

### Results

The operation returns the following:

#### returnResult

The operation's result message.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.

vSphere client.

- The inventory path is case sensitive.

## Standby Guest

The **Standby Guest** operation attempts to put the specified guest virtual machine in standby power mode (suspend to memory).

### Inputs

All of the flow's inputs are described in *Common inputs in the integration*.

### Results

The operation returns the following:

#### returnResult

The operation's result message.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Upgrade Tools

The **Upgrade Tools** operation mounts and runs the VMware Tools installer to upgrade VMware Tools on the guest virtual machine. The requirements for this operation are:

- ESXi Server 3.0.1 or higher.
- The virtual machine must be powered on.
- VMware Tools must be running (**guest.toolsStatus** property must be **toolsOK** or **toolsOld**).
- The VMware Tools version must match with a version that ships with ESXi 3.0. Alternatively, check the **disabledMethod** property of the virtual machine.

### Inputs

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

#### taskTimeOut

The time to wait before the operation is considered to have failed (in seconds).

## installerOptions

Command line options that are passed to the installer to modify the installation procedure for tools.

## Results

The operation returns the following:

### returnResult

The operation's result message.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## VM Tools Running Status

The **VM Tools Running Status** flow retrieves the running status of VM Tools running on the guest operating system of a virtual machine.

## Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

### vmDatacenter

The virtual machine's datacenter.

## Results

The flow returns the following:

### guestToolsRunning

The VM tools are running. You can retrieve Information about the guest by using the **Get Guest Info** operation.

### guestToolsNotRunning

The VM tools are not running.

### guestToolsStarting

The VM Tools are starting and not yet ready to provide data about the guest operating system.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter client.
- The inventory path is case sensitive.

## VM Tools Version Status

The **VM Tools Version Status** flow retrieves the version status of the VM Tools installed on the guest operating system of a virtual machine.

### Inputs

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

#### vmDatacenter

The virtual machine's datacenter.

### Results

The flow returns the following:

#### guestToolsCurrent

The VM tools that are installed and the current version.

#### guestToolsNeedUpgrade

The VM tools are installed but they are not the current version. VM tools can be upgraded by using the **Upgrade Tools** operation.

#### guestToolsNotInstalled

The VM tools are not installed on the guest operating system.

#### guestToolsUnmanaged

The VM tools are installed on the guest operating system but they are not managed by VMware.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter client.
- The inventory path is case sensitive.

# Host

## Add Host

The **Add Host** operation adds a new ESX or ESXi host.

### Inputs

All of the operation's inputs except the following are described in *Common inputs in the integration*.

#### datacenter

The datacenter of the host system.

#### hostSystem

The name or IP address of the host system.

#### force

Specifies whether the connection succeeds even if the host is already being managed by another VirtualCenter server. The original VirtualCenter server loses connection to the host.

The valid values are **true** and **false**.

#### hsUser

The administration account on the host system.

#### hsPass

The password for the administration account.

#### hsPort

The port number for the connection. If you do not specify a value for this input, the **hsPort** value will be the same as the port value.

### Results

The operation returns the following:

#### returnResult

This is the primary output which returns the operation result.

## Disconnect Host

The **Disconnect Host** operation disconnects from a ESX or ESXi host.

### Inputs

All of the operation's inputs except the following are described in *Common inputs in the integration*.

`datacenter`

The datacenter of the vNetwork distributed switch and host system.

`hsIdentifierType`

The host system identifier type.

The valid values are **name**, **hostname**, **ip**, and **hostid**.

`hostSystem`

The host system identifier based on the **hsId**.

## Results

The operation returns the following:

`returnResult`

This is the primary output which is the operation's result.

## Get VM Port Groups from Host

The **Get VM Port Groups from Host** operation returns the names of all the port groups available to virtual machines on a particular host system (ESX or ESXi) as a JSON array.

.

## Inputs

`host`

VMware vCenter, ESX or ESXi host hostname or IP.

`user`

VMware username.

`password`

VMware user's password.

`port`

Port on which to connect. Examples: 443, 80.

`protocol`

Connection protocol. The valid values are **HTTP** and **HTTPS**.

`closeSession`

Close the internally kept VMWare Infrastructure API session at completion of operation. The valid values are **true** and **false**.

#### datacenter

Datacenter of the host system

#### hsIdentifierType

Host system identifier type. The valid values are **name**, **hostname**, **ip**, **hostid**.

#### hostSystem

Host system identifier based on the hsIdentifier type.

### Responses

The operation's responses are:

#### success

The operation completed successfully.

#### failure

Something went wrong. See `returnResult` for more details.

### Returns

The operation returns the following:

#### returnResult

The names of all the port groups as a JSON array. Reason for error on failure.

#### netLegacyJS

The names of the legacy port groups as a JSON array.

#### netDistributedJS

The names of the distributed virtual port groups as a JSON array.

## Reconnect Host

The **Reconnect Host** operation reconnects to a ESX or ESXi host.

### Inputs

All of the operation's inputs except the following are described in [Common inputs in the integration](#).

#### datacenter

The datacenter of the host system.

#### hsIdentifierType

The host system identifier type.

The valid values are **name**, **hostname**, **ip**, and **hostid**.

#### hostSystem

The host system identifier based on the value of the **hsIdentifierType** input.

#### hsUser

The administration account on the host system.

#### hsPass

The password for the administration account.

#### hsPort

The port number for the connection. If you do not specify a value for this input, the **hsPort** value will be the same as the port value.

### Results

The operation returns the following:

#### returnResult

This is the primary output which returns the operation's result.

## Remove Host

The **Remove Host** operation removes an ESX or ESXi host.

### Inputs

All of the operation's inputs except the following are described in [Common inputs in the integration](#).

#### datacenter

The datacenter of the host system.

#### hsIdentifierType

The host system identifier type.

The valid values are **name**, **hostname**, **ip**, and **hostid**.

#### hostSystem

The host system identifier based on the value of the **hsIdentifierType** input.

### Results

The operation returns the following:

#### returnResult

This is the primary output which returns the operation's result.



## Set Maintenance Mode

The **Set Maintenance Mode** operation puts a host system into maintenance mode or exits from the mode. Host systems in maintenance mode cannot have any provisioning operations performed on them or have any virtual machines on it powered on.

### Inputs

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

#### hostSystem

The target virtual machine host system.

#### enterMaintenance

Specifies whether to put the host system into maintenance mode.

The valid values are **true** and **false**.

#### taskTimeOut

The time to wait before the operation is considered to have failed (in seconds).

### Results

The operation returns the following:

#### returnResult

The task or operation result.

## Network

### Add Distributed Port Group

The **Add Distributed Port Group** operation adds a Distributed Port Group to a vNetwork Distributed Virtual switch.

### Inputs

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

#### taskTimeOut

The time to wait in seconds before the operation is considered to have failed.

The default value is **800**.

#### datacenter

The datacenter of the vNetwork distributed switch.

#### netVDSwitchName

The vNetwork distributed switch name.

#### netPortGroup

The name of new distributed port group.

#### netNumPorts

The number of ports for the new distributed port group.

The valid values are **0 – 4096**. **Warning:** The sum of all distributed port group ports on a vNetwork distributed switch cannot exceed **4096**.

#### netBindingType

Defines when ports are assigned to virtual machines that are connected to this new distributed port group.

The valid values are **earlyBinding** (static binding), **ephemeral** (no binding), and **lateBinding** (dynamic binding).

#### netVlanType

The type of VLAN for the new distributed port group.

The valid values are **none**, **vlan**, and **trunk**.

#### netVlanId

The VLAN number or VLAN trunk range for the distributed port group. This input is ignored if you specify a value of **none** for the **netVlanType** input.

For example, for a **netVlanType** VLAN specify **1**, for a **netVlanType** trunk specify **1-4**, **6**, **10-21**.

## Results

The operation returns the following:

#### returnResult

The task or operation result.

## Add Host to vNetwork Distributed Switch

The **Add Host to vNetwork Distributed Switch** operation adds a host system (ESX, ESXi) to a vNetwork distributed switch.

## Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#).

#### taskTimeout

The time to wait in seconds before the operation is considered to have failed.

The default value is **800**.

#### datacenter

The datacenter of the vNetwork distributed switch and host system.

#### hsIdentifierType

The host system identifier type.

The valid values are **name**, **hostname**, **ip**, and **hostid**.

#### hostSystem

The host system identifier based on the value of the **hsIdentifierType** input.

For example, **nameOfHostSystem** (name, as seen in vSphere client), **nameofHostSystem.domain.com** (hostname), **11.22.33.44** (ip), and **host-1234** (hostid).

#### netVDSwitchName

The vNetwork distributed switch name.

#### netNicName

A comma-delimited list of the physical NIC names of the host system that will be attached to the uplink port group of the vNetwork distributed switch.

For example, **vmnic0,vmnic1,vmnic2**.

## Results

The operation returns the following:

#### returnResult

The operation result.

## Add Port Group

The **Add Port Group** operation adds a port group to a virtual switch.

## Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

#### hostSystem

The target virtual machine host system.

#### netVSwitchName

The virtual switch name.

#### netPortGroup

The port group name.

`netVlanID`

The VLAN ID number (enter **0** for none).

## Results

The operation returns the following:

`returnResult`

The operation result.

## Add Virtual NIC

The **Add Virtual NIC** operation adds a virtual NIC to a host system.

### Inputs

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

`hostSystem`

The target virtual machine host system.

`netVNIC`

The virtual NIC name.

`netPortGroup`

The port group name.

`netDHCP`

Specifies whether to use DHCP.

The valid values are **true** and **false**.

`netIP`

The virtual NIC's IP address.

`netSubnetMask`

The subnet mask.

`netMacAddr`

The MAC address.

## Results

The operation returns the following:

`returnResult`

The operation result.

## Add Virtual Switch

The **Add Virtual Switch** operation adds a virtual switch to a host system.

### Inputs

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

#### hostSystem

The target virtual machine host system.

#### netVSwitchName

The virtual switch name.

#### netNumPorts

The number of ports on the virtual switch.

#### netBridgeNICs

A comma-separated list of the virtual NIC names to bridge.

### Results

The operation returns the following:

#### returnResult

The operation's result.

## Add vNetwork Distributed Switch

The **Add vNetwork Distributed Switch** operation adds a vNetwork Distributed Switch to a datacenter. A typical workflow using this operation would be followed by both the **Add Host to vNetwork Distributed Switch** and **Add Distributed Port Group** operations. Once those steps are done, virtual machines and ESX(i) VM kernel interfaces can use the new vNetwork distributed switch for networking.

### Inputs

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

#### taskTimeOut

The time to wait in seconds before the operation is considered to have failed.

The default value is **800**.

#### datacenter

The datacenter for the new vNetwork distributed switch.

#### netVDSwitchName

The name of the new vNetwork distributed switch.

#### netNumPorts

The number of uplink ports for the new vNetwork distributed switch.

The valid values are **1 – 32**.

#### version

The vNetwork distributed switch version.

The valid values are **4.0** and **4.1.0**.

### Results

The operation returns the following:

#### returnResult

The task ID or operation's result.

## Delete Distributed Port Group

The **Delete Distributed Port Group** operation deletes a Distributed Port Group from a vNetwork Distributed Virtual switch.

### Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#).

#### taskTimeout

The time to wait in seconds before the operation is considered to have failed.

The default value is **800**.

#### datacenter

The datacenter from which to delete the distributed port group.

#### netPortGroup

The name of new distributed port group to delete.

### Results

The operation returns the following:

#### returnResult

The operation's result.

## Delete Network Entity

The **Delete Network Entry** operation removes a virtual network entity from a host system or another network entity.

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

`hostSystem`

The target virtual machine host system.

`entityName`

The entity's name.

`netEntityType`

The network entity type.

For example, **VirtualSwitch**, **VirtualNIC**, or **VirtualPortGroup**.

## Results

The operation returns the following:

`returnResult`

The operation result.

## Delete vNetwork Distributed Switch

The **Delete vNetwork Distributed Switch** operation deletes a vNetwork Distributed Switch from a Datacenter.

## Inputs

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

`taskTimeOut`

The time to wait in seconds before the operation is considered to have failed.

The default value is **800**.

`datacenter`

The datacenter of the vNetwork distributed switch.

`netVDSwitchName`

The name of vNetwork distributed switch to delete.

## Results

The operation returns the following:

`returnResult`

The task ID or operation's result.

## Edit Physical NIC Link

The **Edit Physical NIC Link** operation edits a physical NIC's speed and duplex settings on a host system.

### Inputs

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

#### hsIdentifierType

The host system identifier type.

The valid values are **name**, **hostname**, **ip**, and **hosted**.

#### hostSystem

The host system identifier based on the **hsIdentifierType**.

For example, **nameOfHostSystem** (name, as seen in vsphere client), **nameOfHostSystem.domain.com** (hostname), **11.22.33.44** (ip), or **host-1234** (hostid).

#### netVNIC

The virtual NIC name.

For example, **vmnic0** or **vmnic1**.

#### netSpeed

The link speed for the NIC. The only speeds that are supported by the physical NIC are supported. If the value is **0**, the NIC is configured to auto negotiate.

For example, **0** (auto), **10**, **100**, **1000**, or **10000**.

#### netDuplex

The link duplex setting when the speed is 10 or 100.

The valid values are **full** and **half**. This input is ignored if **linkSpeed** is 0 or 1000 and greater since those links are full duplex only.

### Results

The operation returns the following:

#### returnResult

The operation's result.

## Edit Virtual NIC

The **Edit Virtual NIC** operation edits a virtual NIC on a host system (ESX, ESXi). Optional inputs with a "net" prefix that are left empty will not be modified on the virtual NIC.



## Inputs

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

### taskTimeOut

The time to wait in seconds before the operation is considered to have failed.

The default value is 800.

### datacenter

The datacenter of the vNetwork distributed switch and host system.

### hsIdentifierType

The host system identifier type.

The valid values are **name**, **hostname**, **ip**, and **hosted**.

### hostSystem

The host system identifier based on the value of the **hsIdentifierType** input.

For example, **nameOfHostSystem** (name, as seen in vsphere client), **nameofHostSystem.domain.com** (hostname), **11.22.33.44** (ip), or **host-1234** (hostid).

### netVNIC

The virtual NIC name.

For example, **vmk0**, **vmk1**.

### netPortGroup

The port group name. This port group can be a legacy port group or a distributed virtual port group.

### netDHCP

Specifies whether to use DHCP to acquire an IPv4 address.

The valid values are **true** and **false**.

### netIP

The static IPv4 address to use. You must specify a value for this input if the value of the **netDHCP** input is **false**. This input is ignored if the value of **netDHCP** is **true**. For example, **11.22.33.44**.

### netSubnetMask

The subnet mask. You must specify a value for this input if the value of the **netDHCP** input is **false**. This input is ignored if the value of **netDHCP** is **true**.

For example, **255.255.255.0**.

### netMacAddr

The custom MAC address to assign to the NIC. The valid format is six groups of two hexadecimal digits (0-F), separated by colons (:).

For example, **00:50:56:94:AB:EF**.

#### netVMotion

Specifies whether to enable vMotion traffic on this virtual NIC.

The valid values are **true** and **false**.

#### netFTLogging

Specifies whether to enable fault tolerance logging on this virtual NIC.

The valid values are **true** and **false**.

### Results

The operation returns the following:

#### returnResult

The operation's result.

## Remove Host from vNetwork Distributed Switch

The **Remove Host from vNetwork Distributed Switch** operation removes a host system (ESX, ESXi) from a vNetwork distributed switch.

### Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#).

#### taskTimeout

The time to wait in seconds before the operation is considered to have failed.

The default value is **800**.

#### datacenter

The datacenter of the vNetwork distributed switch and host system.

#### hostIdentifierType

The host system identifier type.

The valid values are **name**, **hostname**, **ip**, and **hosted**.

#### hostSystem

The host system identifier based on the value of the **hostIdentifierType** input.

For example, **nameOfHostSystem** (name, as seen in vsphere client), **nameofHostSystem.domain.com** (hostname), **11.22.33.44** (ip), or **host-1234** (hostid).

#### netVDSwitchName

The name of vNetwork distributed switch from which to remove the host.

## Results

The operation returns the following:

`returnResult`

The operation's result.

## Power

### Get Power State

The **Get Power State** operation retrieves the power state of the virtual machine (**poweredOn**, **poweredOff**, or **suspended**).

## Inputs

All of the flow's inputs are described in *Common inputs in the integration*.

## Results

The operation returns the following:

`returnResult`

The task completion return message.

`powerState`

The power state.

The possible values are **poweredOn**, **poweredOff**, or **suspended**.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Restart

The **Restart** operation restarts (interrupts power) the virtual machine.

## Inputs

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

`taskTimeout`

The time to wait before the operation is considered to have failed (in seconds).

## Results

The operation returns the following:

`returnResult`

The task ID or completion return message.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Set Power State

The **Set Power State** operation sets the power state of the virtual machine (**poweredOn**, **poweredOff**, or **suspended**). It does nothing if the virtual machine is already in the power state.

## Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

`taskTimeout`

The time to wait before the operation is considered to have failed (in seconds).

`vmPowerState`

The virtual machine power state to set.

The valid values are **poweredOn**, **poweredOff**, and **suspended**.

## Results

The operation returns the following:

`returnResult`

The task ID or completion return message.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.

- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Samples

### Get Freespace On Datastores

The **Get Freespace On Datastores** flow gets a list of datastores on a Virtual Infrastructure host with the amount of free space left on each datastore.

#### Inputs

All of the flow's inputs are described in *Common inputs in the integration*.

#### Results

The flow returns the following:

`returnResult`

A pipe-delimited list of datastore IDs.

`datastoreNames`

A pipe-delimited list of datastore names corresponding to the datastore IDs output.

`freeSpaces`

A pipe-delimited list of free space in the corresponding datastores.

### Get List of Powered On VMs

The **Get List of Powered On VMS** flow gets list of powered on VMs in a Virtual Center or ESXi host.

#### Inputs

All of the flow's inputs are described in *Common inputs in the integration*.

#### Results

The flow returns the following:

`returnResult`

A pipe-delimited list of virtual machines. Each entry is in the form **virtualmachineId:virtualmachineName**.

## Get VMs Powered On Longer Than By Event

The **Get VMs Powered On Longer Than By Event** flow generates a report of the non-clustered VMs that are powered on for more than the specified number of hours. The results include the user who powered the VM on, the name of the VM, and the time that the VM was powered on.

### Inputs

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

#### vmDatacenter

The datacenter from which to report the virtual machines.

#### localeLang

The locale language in which to format the **createdTime** result. If you do not specify a value for this input, the default value of **en** is used.

For example, **en** or **ja**.

#### localeCountry

The locale country in which to format the **createdTime** result. If you do not specify a value for this input, the default value of **US** is used.

For example, **US**.

#### hours

The number of hours the VMs are powered on for more than.

For example, if you specify **168**, the operation will generate a report containing information about the VMs that are powered on for more than a week.

#### notifyMethod

The method by which you are notified of the results.

The valid values are **Write to File**, **Email**, **Display**, and **None**. These values are case-sensitive.

#### destination

The destination of the notification (the e-mail address or file name). This input is required when the **notifyMethod** input value is **Email** or **Write to File**.

#### notificationServer

The e-mail server through which to send the notification. This input is required when the **notifyMethod** input value is **Email**.

#### from

The originator of the message (the e-mail "from" address). This field is required when **notifyMethod** is **Email**.

### serverPort

The port on the server on which the service to notify is running. You can use this optional input when the **notifyMethod** input value is **Email**.

The default value is **25**.

### htmlEmail

Specifies whether the e-mail should be in rich text HTML format.

The valid values are **true** (if the e-mail is in rich text/html format) and **false** if the e-mail is in plain text format.

The default value is **true**.

## Results

The flow returns the following:

A notification of the information about the VMs that are powered on for more than the specified number of hours. The notification is made using the method you specify in the **notifyMethod** input value - **Email**, **Write to File**, or **Display**. The format of the notification is:

The user who powered on the VM | the VM name | The time the VM was powered on

## Get VMs Powered On Longer Than By Task

The **Get VMs Powered On Longer Than By Task** flow generates a report of the clustered/ non-clustered VMs that are powered on for more than the specified number of hours. The results include the users who powered the VMs on, the names of the VMs, and the time that the VMs were powered on.

## Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

### vmDatacenter

The Datacenter from which to get the virtual machines.

### localeLang

The locale language in which to format the **createdTime** result. If you do not specify a value for this input, the default value of **en** is used.

For example, **en** or **ja**.

### localeCountry

The locale country in which to format the **createdTime** result. If you do not specify a value for this input, the default value of **US** is used.

For example, **US**.

### hours

The number of hours the VMs are powered on for more than.

For example, if you specify **168**, the operation will generate a report containing information about the VMs that are powered on for more than a week.

#### notifyMethod

The method by which you are notified of the results.

The valid values are **Write to File**, **Email**, **Display**, and **None**. These values are case-sensitive.

#### destination

The destination of the notification (the e-mail address or file name). This input is required when the **notifyMethod** input value is **Email** or **Write to File**.

#### notificationServer

The email server through which to send the notification through. This input is required when **notifyMethod** input value is **Email**.

#### from

The originator of the message (the e-mail "from" address). This field is required when **notifyMethod** is **Email**.

#### serverPort

The port on the server on which the service to notify is running. You can use this optional input when the **notifyMethod** input value is **Email**.

The default value is **25**.

#### htmlEmail

Specifies whether the e-mail should be in rich text HTML format.

The valid values are **true** (if the e-mail is in rich text/html format) and **false** if the e-mail is in plain text format.

The default value is **true**.

### Results

The flow returns the following:

A notification of the information about the VMs that are powered on for more than the specified number of hours. The notification is made using the method you specify in the **notifyMethod** input value: **Email**, **Write to File**, or **Display**. The format of the notification is:

The user who powered on the VM | the VM name | The time the VM was powered on

### Guided Search

The **Guided Search** flow guides the user through the inputs necessary to use the **Advanced Search** operation.



## Inputs

All of the flow's inputs are described in [Common inputs in the integration](#).

## Results

The flow returns the following:

`returnResult`

The search result information.

## Host Maintenance With No Downtime

The **Host Maintenance With No Downtime** flow places a live ESXi server into maintenance mode, and migrates all running VMs to a specified standby ESXi server. The execution of the flow pauses when this part is complete to allow the user to perform maintenance on the server. Then the migrations are reversed and the original running states are restored.

## Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

`primaryHypervisor`

The ESXi hypervisor to put into maintenance mode.

`standByHypervisor`

The ESXi hypervisor to use as a standby/failover host. The value of this input must match the name in the Virtual Center.

## Results

The flow returns the following:

`returnResult`

The task or operation result of the last operation.

## Quick Migrate

The **Quick Migrate** flow suspends a virtual machine, relocates it to another host system and datastore, and powers the virtual machine on.

## Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

`vmDatacenter`

The virtual machine's datacenter.

## virtualMachine

The primary virtual machine identifier. The format of this the value depends on the value of **vmIdentifierType**:

- **inventoryPath** — The inventory path in the format **Datacenter/vm/Folder/MyVM**.
- **name** — The name of VM.
- **ip** — IPv4 or IPv6 depending upon the ESXi version.
- **hostname** — The fully-qualified domain name of the host.
- **uuid** — The UUID.
- **vmid** — The VM ID in the format **vm-123,123**.

## vmIdentifierType

The virtual machine identifier type specified by using the inventory path, the name, the IP address, the host name, the UUID, or the VMID.

## hostSystem

The target virtual machine host system.

## dataStore

The datastore.

For example, **host:dsname, mydatastore**.

## Results

The operation returns the following:

The return result from the last executed step of the multi-stage procedure.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Samples/Subflows

### Get Powered On VM List As IDs

The **Get Powered On VM List As IDs** flow returns the VM IDs of powered on VMs on the specified host system under VMware Virtual Center/vCenter as a pipe-delimited list.

## Inputs

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

### hostSystem

The ESXi hypervisor to check for powered on VMs.

## Results

The flow returns the following:

A pipe-delimited list of VMs that are powered on, with each entry formatted as **vm-digits:vmName** or **digits:vmName**.

# Snapshots

## Create Snapshot

The **Create Snapshot** operation creates a snapshot of an existing virtual machine's disks and, optionally, the virtual machine memory if it is running.

## Inputs

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

### taskTimeOut

The time to wait before the operation is considered to have failed (in seconds).

### snapshotName

The name of the snapshot.

### description

The description or annotation of the snapshot.

### snapshotMem

Specifies whether the snapshot virtual machine memory as well as the virtual machine is running.

The valid values are **true** and **false**.

The default value is **true**.

### quiesce

Specifies whether the **Quiesce** guest file system of VMware Tools is installed on the guest operating system and the virtual machine is running.

The valid values are **true** and **false**.

The default value is **true**.

## Results

The operation returns the following:

`returnResult`

The task ID or the task completion return message.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Delete Snapshot

The **Delete Snapshot** operation deletes an existing snapshot of a virtual machine or all snapshots if none are specified.

### Inputs

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

`taskTimeout`

The time to wait before the operation is considered to have failed (in seconds).

`snapshotName`

The name of the snapshot.

`removeChildren`

Specifies whether the children of the snapshot should be removed.

The valid values are **true** and **false**.

`removeAllSnapshots`

Specifies whether to remove all snapshots.

The valid values are **true** and **false**.

## Results

The operation returns the following:

`returnResult`

The task ID or the task completion return message.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Get Snapshot

The **Get Snapshot** operation retrieves information about a virtual machine's snapshot. If the name of snapshot not specified, the operation returns the snapshot tree.

### Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

**snapshotName**

The name of the snapshot.

### Results

The operation returns the following:

**returnResult**

The task completion return message.

**snapshotInfo**

The snapshot information.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Revert to Snapshot

The **Revert to Snapshot** operation reverts a virtual machine to an existing named snapshot or the most current one if none is specified.

## Inputs

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

### taskTimeOut

The time to wait before the operation is considered to have failed (in seconds).

### snapshotName

The name of the snapshot.

## Results

The operation returns the following:

### returnResult

The task ID or task completion return message.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Tasks

### Cancel Task

The **Cancel Task** operation cancels a task.

## Inputs

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

### taskId

The task ID.

For example, **task-1234**.

## Results

The operation returns the following:

`returnResult`

The task completion return message.

## Get All Running Tasks

The **Get All Running Tasks** flow retrieves the list of the currently running tasks.

### Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

`tasks`

The types of running tasks you want to retrieve. This value is case-insensitive. It accepts any character-delimited list of task types. If you do not specify a value for this input, it retrieves all of the running tasks.

For example, **VirtualMachine.relocate**, **VirtualMachine.clone**, **VirtualMachine.reconfigure**, **VirtualMachine.powerOff**, **VirtualMachine.powerOn**.

### Results

The flow returns the following:

`runningTasks`

The list of currently running tasks.

The valid format is **taskId : descriptionId**.



To determine the valid task types, run the flow without specifying a value for the **tasks** input. This returns the list of all the currently running tasks. The **descriptionId** from the **runningTasks** could be used to determine the valid task types.

## Get Task State

The **Get Task State** operation retrieves the current state of the task.

### Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

`taskId`

The task ID.

For example, **task-1234**.

### Results

The operation returns the following:

`returnResult`

The task information.

`taskInfo`

The task information.

`taskState`

The task state.

The valid values are **running**, **success**, **queued**, and **error**.

`taskProgress`

The task progress.

The valid values are **0-100**, inclusive.

`taskSuccess`

Specifies whether the task was successful.

The valid values are **true** and **false**.

`taskError`

The task error, if any.

`taskResult`

The task result.

## Utilities

### Advanced Search

The **Advanced Search** operation searches for VMware objects of type **propsType** and their properties (**propsPathset**) from a root object (if any) specified by the values you specify for the **propsRootObjType** and **propsRootObj**.inputs. It retrieves each object's properties one at a time from an initial search query that acts like a SQL Query operation.

This operation directly exposes ID types, objects and their relations, classes, and properties of the virtual infrastructure (VI) SDK used to integrate with VI, so users must understand how different VI objects are related.

### Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

`propsType`

The managed object type to search for.

For example, to find datastore objects, enter **Datastore** as the type.



**Note:** Some objects may not be reachable with the default traversal path and require multiple uses of the **Advanced Search** operation.

#### propsPathSet

A comma-separated list of property to retrieve. Each property is dot-delimited for different levels of nesting and supports arrays using square brackets around the key or index (for instance, **propertyName[indexOrKey]**).

For example, you can find the CPU reservation on a virtual machine by using the property path **resourceConfig.cpuAllocation** or the **deviceInfo** of an LSI Logic SCSI Controller via **config.hardware.device[1000].deviceInfo**.

#### propsRootObjType

The type of the Managed Object Reference to begin the search on. Leaving this input blank defaults the root object to the root folder of the ESXi or VC host.

For example, **VirtualMachine**, **HostSystem**, and **ResourcePool**.

#### propsRootObj

The VMware object ID used as a root object. Leaving this input blank defaults the root object to be the root folder of the ESXi or VC host.

For example, **vm-123**, **datastore-123**, or **network-123**.

## Results

The operation returns the following:

#### numberOfResults

The number of results from the query remaining to be retrieved.

#### returnResult

The query result item or operation result.



To retrieve objects that can't be directly found by the default recursive settings, you can use two calls to the operation. First, use the **propsPathset** input to specify the properties of a managed object of interest that you can traverse to. With the managed object IDs returned, you can set these objects as the root objects and retrieve properties with additional calls to the operation.

For an example of the operation, please see the **Get Freespace On Datastores** sample flow in the Library/Integrations/VMware/VMware Virtual Infrastructure and vSphere/Samples/ folder.

## Destroy Object

The **Destroy Object** operation destroys or deletes an object in VMware.

### Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

`taskTimeout`

The time to wait before the operation is considered to have failed (in seconds).

`propsRootObjType`

The VMware object type.

For example, **VirtualMachine**, **HostSystem**, **ComputeResource**, **ResourcePool**, and **Alarm**.

`propsRootObj`

The VMware object ID.

## Results

The operation returns the following:

`returnResult`

The task ID or task completion return message.

## Get Latest Event

The **Get Latest Event** operation retrieves the most recent event from the root object you specify using the **objType** and **objId** inputs, based on the event types you specify in the events input.

## Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

`events`

The list of event types, from which the event log for the specified events is created.

For example, **VmPoweredOnEvent**, **VmRelocatedEvent**, **VmReconfiguredEvent**, **VmCreatedEvent**, **VmMigratedEvent**, or **VmRenamedEvent**.

`objType`

The type of Managed Object Reference on which to begin the search.

For example, **VirtualMachine**, **HostSystem**, **ComputeResource**, **ResourcePool**, **Datacenter**, or **Datastore**.

`objId`

The value of the reference to use as a root object.

For example, **vm-123**, **datastore-123**, or **network-123**.

`localeLang`

The locale language for which to format the **createdTime** result. If you do not specify a value for this input, the default value of **en** is used.

For example, **en** or **ja**.

#### localeCountry

The locale country for which to format the **createdTime** result. If you do not specify a value for this input, the default value of **US** is used.

For example, **US**.

### Results

The operation returns the following:

#### UserName

The name of the user who caused the event.

#### createdTime

The time the event was created.

#### vm

The **virtualMachine** object of the event.

#### returnResult

The **UserName** or the reason for failure.

### Get Latest Task

The **Get Latest Task** operation retrieves the most recent task from the root object you specify using the **objType** and **objId** inputs, based on the task types you specify in the tasks input.

### Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

#### tasks

The list of task types from which the event log for the specified tasks is created.

For example, **Drm.ExecuteVmPowerOnLRO**, **VirtualMachine.powerOn**, **VirtualMachine.powerOff**, or **VirtualMachine.reconfigure**.

#### objType

The type of Managed Object Reference on which to begin the search.

For example, **VirtualMachine**, **HostSystem**, **ComputeResource**, **ResourcePool**, **Datacenter**, or **Datastore**.

#### objId

The value of the reference to use as a root object.

For example, **vm-123**, **datastore-123**, or **network-123**.

### localeLang

The locale language for which to format the **createdTimer** result. If you do not specify a value for this input, the default value of **en** is used.

For example, **en** or **ja**.

### localeCountry

The locale country for which to format the **createdTimer** result. If you do not specify a value for this input, the default value of **US** is used.

For example, **US**.

## Results

The operation returns the following:

### taskId

The ID of the task.

### descriptionId

The type of the task.

### entityName

The VM name.

### state

The current state of the task.

### parentId

The (parentId) of the task, if it exists.

### UserName

The name of the user who caused the task.

### createdTime

The completed time of the task.

### returnResult

The **taskId** or the reason for failure.

## Get OS Descriptors

The **Get OS Descriptors** operation identifies supported guest operating systems on a host system.

## Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

#### vmDatacenter

The virtual machine's datacenter. Use **ha-datacenter** if the host is an ESXi host.

#### hostSystem

The target virtual machine host system. If the host is an ESXi host, use the full hostname of the ESXi host as seen in the vSphere client for this host.

### Results

The operation returns the following:

#### guestOsId

A pipe-delimited list of the guest operating system IDs.

#### guestOsDescription

A pipe-delimited list of the guest operating system descriptions.

#### returnResult

A pipe-delimited list of the guest operating system IDs.

### List Custom Fields

The **List Custom Fields** operation lists the custom field definitions in Virtual Center. To pair custom field names with their values, match custom field values retrieved from other operations with the corresponding custom field key entries in this listing.

### Inputs

All of the flow's inputs are described in *Common inputs in the integration*.

### Results

The operation returns the following:

#### returnResult

The custom fields found.

### Simple Search

The **Simple Search** operation searches for VMware objects containing a given name (ignoring case). It retrieves each object one at a time from an initial search query using a SQL Query operation.

### Inputs

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

#### propsType

The VMware properties type on which to filter. The type of object results desired.

The valid values are **VirtualMachine**, **Network**, **ComputeResource**, **ResourcePool**, **HostSystem**, **Datastore**, and **Datacenter**.

#### `nameToMatch`

The string to perform a case-insensitive contains match to. Only objects that contain this string in their name will be matched. If you do not specify this input, all objects of the specified type are matched.

### Results

The operation returns the following:

#### `numberOfResults`

The number of results from the query remaining to be retrieved.

#### `returnResult`

The query result item or operation result.

## Virtual Apps/Samples

### Import OVF to Cluster

Imports an OVF template to a DRS cluster as a virtual machine or virtual app.

### Inputs

All of the flow's inputs are described in *[Common inputs in the integration](#)*.

#### `path`

Path to the .ovf file on the RAS filesystem to import.

#### `name`

Name of the new virtual machine or virtual app.

#### `Datacenter`

Datacenter of the host system or cluster.

#### `vmFolder`

Virtual machine's inventory folder. Folder names are delimited by a forward slash "/". This input is case sensitive. Only supported when host is a vCenter. For root folder, use "/".

#### `dataStore`

Name of datastore to store new virtual machine (eg. host:dsname, mydatastore). If not specified the same datastore of the source virtual machine will be used.

#### `thinProvision`

Perform thin provisioning of the virtual disks.

#### clusterName

Name of the VMWare HA or DRS cluster.

#### localeLang

The locale language in which to process the OVF. If you do not specify a value for this input, the default locale language of the RAS will be used.

#### localeCountry

The locale country in which to process the OVF. If you do not specify a value for this input, the default locale country of the RAS will be used.

#### Parallel

If the ovf template has multiple .vmdk files, should they be uploaded in parallel? If true, all .vmdk files will be uploaded using separate threads.

### Results

The operation returns the following:

#### FailureMessage

Reason for failure if flow fails.

## Import OVF to Host

Imports an OVF template to an ESX or ESXi host system as a virtual machine or virtual app.

### Inputs

#### path

Path to the .ovf file on the RAS filesystem to import.

#### name

Name of the new virtual machine or virtual app.

#### datacenter

Datacenter of the host system or cluster.

#### vmFolder

Virtual machine's inventory folder. Folder names are delimited by a forward slash "/". This input is case sensitive. Only supported when host is a vCenter. For root folder, use "/".

#### dataStore

Name of datastore to store new virtual machine (eg. host:dsname, mydatastore). If not specified the same datastore of the source virtual machine will be used

#### thinProvision

Perform thin provisioning of the virtual disks.

#### hsIdentifierType

Host system identifier type.

#### hostSystem

Host system identifier based on the hsIdentifier Type.

#### localeLang

The locale language in which to process the OVF. If you do not specify a value for this input, the default locale language of the RAS will be used.

#### localeCountry

The locale country in which to process the OVF. If you do not specify a value for this input, the default locale country of the RAS will be used.

#### Parallel

If the ovf template has multiple .vmdk files, should they be uploaded in parallel? If true, all .vmdk files will be uploaded using separate threads.

### Results

The operation returns the following:

#### FailureMessage

Reason for failure if flow fails.

## Virtual Apps/Samples/Subflows

### Prompt For Each Key

Prompts for each key

#### Inputs

##### keys

The keys for which to prompt.

### Prompt For Each Key With Selection List

Prompts for each key from a given selection list.

#### Inputs

##### keys

The keys for which to prompt.



## possibleValues

The possible values for the keys

# Virtual Apps

## Export vApp as OVF

Exports a virtual application (vApp) as an Open Virtualization Format (ovf) template in a folder of files format to the local file system of the RAS. This operation will create an .ovf file, and .vmdk files for each disk used by the VMs of the vApp and an .mf file with the SHA1 hashes for the .ovf and each .vmdk file.

## Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#).

### datacenter

Name of datacenter to isolate search of vApp.

### vAppIdentifierType

The virtual application identifier type.

The valid values are: **inventorypath**, **name**, **vappid**.

### virtualApp

Primary virtual application identifier. The format of this the value depends on the value of **vAppIdentifierType**:

- **inventorypath** — The inventory path in the format **Datacenter/vm/Folder/MyVApp**.
- **name** — The name of the vApp.
- **vappid** — The vApp ID in the format **resgroup-v123**.

### localPath

Local directory path on the RAS to write the .ovf template files. The directory must already exist.

### parallel

If the virtual application has multiple .vmdk files, should they be downloaded in parallel? Valid values: true, false. If true, all .vmdk files will be downloaded using separate threads.

## Get vApp

Retrieves the name and managed object id of a virtual application (vApp).

## Inputs

### datacenter

Name of datacenter to isolate search of vApp.

### `vApplIdentifierType`

Virtual application identifier type

### `virtualApp`

Primary virtual application identifier. Inventory path (Datacenter/vm/Folder/MyVApp), Name of vApp, or the vApp id (resgroup-v123).

## Results

The operation returns the following:

### `returnResult`

Operation result or reason for failure.

### `Name`

Name of the vApp.

### `vApplId`

Managed object id of the vApp.

## Get VMs in vApp

Returns a list of virtual machines (VMs) in a virtual application (vApp).

## Inputs

### `datacenter`

Name of datacenter to isolate search of vApp.

### `vApplIdentifierType`

Virtual application identifier type

### `virtualApp`

Primary virtual application identifier. Inventory path (Datacenter/vm/Folder/MyVApp), Name of vApp, or the vApp id (resgroup-v123).

### `Delimiter`

A delimiter for separating values in the vmid and name result lists.

## Results

The operation returns the following:

### `count`

Number of virtual machines found in the vApp.

## Vmid

List of virtual machine ids found in the vApp delimited by delimiter.

## Name

List of virtual machine names found in the cluster delimited by delimiter.

## Import OVF

Imports an OVF image to a host system (ESX or ESXi) or DRS Cluster as a virtual machine (VM) or virtual app (vApp).

## Inputs

### path

Path to the .ovf file on the RAS filesystem to import.

### name

Name of the new virtual machine or virtual app.

### datacenter

Datacenter of the host system or cluster.

### vmFolder

Virtual machine's inventory folder. Folder names are delimited by a forward slash "/". This input is case sensitive. Only supported when host is a vCenter. For root folder, use "/".

### dataStore

Name of datastore to store new virtual machine (eg. host:dsname, mydatastore). If not specified the same datastore of the source virtual machine will be used

### thinProvision

Perform thin provisioning of the virtual disks.

### hsIdentifierType

Host system identifier type.

### hostSystem

Host system identifier based on the hsIdentifier Type.

### clusterName

Name of the VMWare HA or DRS cluster. Can be specified instead of hostSystem if the desired destination ESX(i) host is in a DRS or HA cluster.

### ovfNetworkJS

A JSON array of network in the ovf template to be mapped to vm port groups. The netPortGroupJS input will be a complimentary array that defined the target port groups for these networks.

### netPortGroupJS

A JSON array of port groups that the ovf networks in the template will attach to. The ovfNetworkJS input defined the source networks in the ovf template for these portgroups.

### ovfPropKeyJS

A JSON array of property names to be configured during import of the ovf template.

### ovfPropValueJS

A JSON array of property values respective to the property names defined in ovfPropKeyJS to be applied during import of the ovf template. Including the example from ovfPropKeyJS input, property "vami.ip0.vmName" will have the value "10.10.10.10" and "vami.ip1.vmName" will have the value "10.20.30.40".

### ipProtocol

If supported, the IP protocol of the vApp.

### ipAllocScheme

IP Allocation policy for the deployment of a vApp.

### dhcpPolicy

Specifies that DHCP must be used to allocate IP addresses to the vApp.

### fixedPolicy

The IP addresses are allocated when the vApp is deployed and will be kept with the server as long as it is deployed.

### transientPolicy

The IP addresses are allocated when needed, typically at power-on, and deallocated during power-off.

### localeLang

The locale language in which to process the OVF. If you do not specify a value for this input, the default locale language of the RAS will be used.

### localeCountry

The locale country in which to process the OVF. If you do not specify a value for this input, the default locale country of the RAS will be used.

### parallel

If the ovf template has multiple .vmdk files, should they be uploaded in parallel? If true, all .vmdk files will be uploaded using separate threads.

## Results

The operation returns the following:

`returnResult`

Task result or operation result.

## Parse OVF

Returns details about an OVF (open virtualization format) template. Some of returned details may be needed in a flow using "Import OVF".

### Inputs

`path`

Local file path on the RAS to read the ovf template path to the .ovf file on the RAS filesystem to import.

### Results

The operation returns the following:

`returnResult`

Operation result or reason for failure.

`deployOptions`

Deployment options if specified in the .ovf file.

`isVirtualApp`

Returns true if the .ovf represents a virtual app. Returns false if it represents a virtual machine.

`ovfNetworkJS`

A JSON array of the network names in the .ovf file that will need to be mapped to respective port groups during import.

`name`

Name of the vApp or virtual machine as defined in the .ovf file.

`ovfPropKeyJS`

A JSON array of user configurable properties if defined in the .ovf file.

`ipAllocSchemes`

IP address allocation schemes supported if specified in the .ovf file.

`ipProtocols`

A comma separated list of the supported IP protocols if specified in the .ovf file.

## Power Off vApp

Powers off all Virtual Machines in a Virtual Application (vApp). This operation ignores any defined shutdown order of the vApp.

## Inputs

### taskTimeout

Time to wait in seconds before the operation is considered to have failed.

### datacenter

Name of datacenter to isolate search of vApp.

### vAppIdentifierType

Virtual application identifier type

### virtualApp

Primary virtual application identifier. Inventorypath (Datacenter/vm/Folder/MyVApp), Name of vApp, or the vApp id (resgroup-v123).

## Results

The operation returns the following:

### returnResult

Task id or Operation result.

## Power On vApp

Powers on a Virtual Application (vApp). This operation follows the startup order of the vApp if defined.

## Inputs

### taskTimeout

Time to wait in seconds before the operation is considered to have failed.

### datacenter

Name of datacenter to isolate search of vApp.

### vAppIdentifierType

Virtual application identifier type

### virtualApp

Primary virtual application identifier. Inventorypath (Datacenter/vm/Folder/MyVApp), Name of vApp, or the vApp id (resgroup-v123).

## Results

The operation returns the following:

`returnResult`

Task id or Operation result.

## Shutdown vApp

Powers off all Virtual Machines in a Virtual Application (vApp). This operation follows the defined shutdown order of the vApp.

### Inputs

`taskTimeout`

Time to wait in seconds before the operation is considered to have failed.

`datacenter`

Name of datacenter to isolate search of vApp.

`vAppIdentifierType`

Virtual application identifier type

`virtualApp`

Primary virtual application identifier. Inventory path (Datacenter/vm/Folder/MyVApp), Name of vApp, or the vApp id (resgroup-v123).

### Results

The operation returns the following:

`returnResult`

Task id or Operation result.

## Suspend vApp

Suspends all Virtual Machines in a Virtual Application (vApp).

### Inputs

`taskTimeout`

Time to wait in seconds before the operation is considered to have failed.

`datacenter`

Name of datacenter to isolate search of vApp.

`vAppIdentifierType`

Virtual application identifier type

`virtualApp`

Primary virtual application identifier. Inventory path (Datacenter/vm/Folder/MyVApp), Name of vApp, or the vApp id (resgroup-v123).

## Results

The operation returns the following:

`returnResult`

Task id or Operation result.

# Virtual Machines/Configuration

## Add Existing Disk to VM

The **Add Existing Disk to VM** operation adds an existing virtual (SCSI) disk to an existing virtual machine.

## Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#).

`taskTimeOut`

The time to wait in seconds before the operation is considered to have failed.

`vmDatacenter`

The virtual machine's datacenter.

`dataStore`

The datastore.

For example, **host:dsname, mydatastore**.

`diskPath`

The path to the .vmdk file on the datastore to use for backing the new virtual disk. Include .vmdk suffix and do not include the leading "/". This path is case sensitive and must already exist on the datastore.

`diskMode`

The disk persistence mode.

The valid values for ESXi 3 and later VMs are **persistent**, **independent\_persistent**, and **independent\_nonpersistent**.

The valid values for pre-ESXi 3 VMs are **persistent**, **nonpersistent**, **undoable**, and **append**.



## Results

The operation returns the following results:

### returnResult

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter client.
- The inventory path is case sensitive.

## Add Floppy Drive To VM

The **Add Floppy Drive To VM** operation adds a floppy drive to a virtual machine.

## Inputs

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

### taskTimeout

The time to wait in seconds before the operation is considered to have failed.

The default value is **800**.

### vmDatacenter

The virtual machine's datacenter.

## Results

The operation returns the following results:

### returnResult

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter client.
- The inventory path is case sensitive.

## Add New Disk To VM

The **Add New Disk To VM** operation adds a new virtual (SCSI) disk to an existing virtual machine.

### Inputs

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

#### taskTimeout

The time to wait before the operation is considered to have failed (in seconds).

#### vmDatacenter

The virtual machine's datacenter.

#### dataStore

The datastore.

For example, **host:dsname, mydatastore**.

#### vmDiskSize

The size of the virtual disk to create (in MB).

#### diskPath

The path on the datastore to use for backing the virtual disk. Include the .vmdk suffix and do not include a leading slash (/).

#### diskMode

The disk persistence mode.

The valid values for ESXi 3 and later VMs are **persistent**, **independent\_persistent**, and **independent\_nonpersistent**.

The valid values for pre-ESXi 3 VMs are **persistent**, **nonpersistent**, **undoable**, and **append**.

#### thinProvision

Specifies whether thin provisioning of the virtual disk should be performed.

The valid values are **true** and **false**.

#### diskWritethrough

Specifies whether to enable direct write to the file system.

The valid values are **true** and **false**.

### Results

The operation returns the following results:

`returnResult`

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Add NIC to VM

The **Add NIC to VM** operation adds a network interface card to a virtual machine assigned to a port group or network name.

### Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#).

`taskTimeOut`

The time to wait before the operation is considered to have failed (in seconds).

`vmDatacenter`

The virtual machine's datacenter.

`netPortGroup`

The name of the port to which to connect this new NIC. This port group can be a legacy port group or a distributed virtual port group.

`netNicType`

The virtual network interface card type.

The valid values are **PCNet32**, **E1000**, **VmxNet**, **VmxNet2**, and **VmxNet3**. NIC type **VmxNet2** is only supported for virtual machine hardware version 4 and later. NIC type **VmxNet3** is only supported for virtual machine hardware version 7 and later.

`netMacAddr`

The MAC address to assign if you specify a value of **Manual** for the **netMacGeneration** input.

`netMacGeneration`

The MAC address assignment.

The valid values are **Manual**, **Generated**, and **Assigned**. (The **Assigned** value is only valid when the host is a vCenter server. The MAC is assigned from vCenter).

### netWakeOnLan

Specifies whether to enable **Wake On LAN** for the NIC.

The valid values are **true** and **false**.

### Results

The operation returns the following:

#### returnResult

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Add Optical Drive To VM

The **Add Optical Drive To VM** operation adds an optical drive (CD/DVD) to a virtual machine. The device type of the new optical drive will be **Client Device** in **Passthrough IDE** mode. To change the device type of the optical drive, use the **Edit Optical Drive on VM** operation.

### Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#).

#### taskTimeout

The time to wait in seconds before the operation is considered to have failed.

The default value is **800**.

#### vmDatacenter

The virtual machine's datacenter.

### Results

The operation returns the following:

#### returnResult

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Configure NIC on VM

The **Configure NIC on VM** operation modifies an existing Network Interface Card (NIC) on a virtual machine.

### Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#).

#### taskTimeout

The time to wait before the operation is considered to have failed (in seconds).

#### vmDatacenter

The virtual machine's datacenter.

#### deviceNumber

The NIC number to modify on the VM.

For example, **Network Adapter 1** would be **deviceNumber 1**.

#### netPortGroup

The name of the port group to which to connect this NIC. This port group can be a legacy port group or a distributed virtual port group.

For example, **VM Network** or **dvPortGroup**.

#### netNicType

The virtual network interface card type.

The valid values are **PCNet32**, **E1000**, **VmxNet**, **VmxNet2**, and **VmxNet3**. NIC type **VmxNet2** is only supported for virtual machine hardware version 4 and later. NIC type **VmxNet3** is only supported for virtual machine hardware version 7 and later.

#### netMacAddr

The MAC address to assign if **netMacGeneration** is set to **Manual**.

The valid format is six groups of two hexadecimal digits (0-F), separated by colons (:).

For example, **00:50:56:94:AB:EF**.

#### netMacGeneration

The MAC address assignment.

The valid values are **Manual**, **Generated**, and **Assigned**. (The **Assigned** value is only valid when the host is a vCenter server. The MAC is assigned from vCenter).

#### netWakeOnLan

Specifies whether to enable **Wake On LAN** for the NIC.

The valid values are **true** and **false**.

## Results

The operation returns the following:

#### returnResult

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Edit Floppy Drive on VM

The **Edit Floppy Drive on VM** operation edits an existing floppy drive on a virtual machine.

## Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#).

#### taskTimeout

The time to wait in seconds before the operation is considered to have failed.

The default value is **800**.

#### vmDatacenter

The virtual machine's datacenter.

#### deviceNumber

The floppy drive number on the VM.

For example, **Floppy drive 1** would be **deviceNumber 1**.

#### deviceType

The type of floppy device. The valid values are:

- **datastore**

Uses an FLP image from a datastore. Requires the **dataStore** and **diskPath** inputs to be populated.

- **hostDevice**

Uses the floppy drive of the ESX/ESXi host system. Optionally, you can use the **hostDevice** input to specify the desired host device. Otherwise the first available device is used.

- **clientDevice**

Sets the device type to **Client Device**.

#### dataStore

The datastore in which to find the ISO image if you specify a value of **datastore** for the **deviceType** input.

For example, **host:dsname, mydatastore**.

#### diskPath

The path on the datastore to use for the FLP image. Include an .flp suffix and do not include the leading /. This FLP image must already exist on the datastore if you specify a value of **false** for the **createImage** input.

#### hostDevice

The device path of the floppy drive on the host system (for example, **/dev/fd0**). Use this input if you specify a value of **deviceType** for the **hostDevice** input. If you do not specify a value for this input, the first available floppy drive is used.

#### connectAtPowerOn

Specifies whether the floppy drive should be connected at the time the VM is powered on. This input is not applicable for client device types. If you do not specify a value for this input, the existing setting is retained.

The valid values are **true** and **false**.

#### createImage

Specifies whether to create a blank floppy image. Use this input if you specify a value of **datastore** for the **deviceType** input.

The valid values are **true** and **false**. If you specify a value of **true**, a new floppy image specified in the **diskPath** input is created on the datastore. If you specify **false**, the floppy image specified in **diskPath** must already exist on the datastore. The default value is **false**.

## Results

The operation returns the following:

#### returnResult

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.

- If the host is a vCenter, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter client.
- The inventory path is case sensitive.

## Edit Optical Drive on VM

The **Edit Optical Drive on VM** operation adds a new virtual (SCSI) disk to an existing virtual machine.

### Inputs

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

#### taskTimeOut

The time to wait before the operation is considered to have failed (in seconds).

#### virtualMachine

The Primary Virtual Machine identifier.

The valid values are the inventory path (in the format **Datacenter/vm/Folder/MyVM**), the name of the VM, IP (IPv4 or IPv6 depending upon ESXi version), the full hostname, the UUID, or the VM ID (in the format **vm-123,123**).

#### vmIdentifierType

The virtual machine identifier type.

The valid values are the inventory path, name, IP address, hostname, UUID, or VMID.

#### vmDatacenter

The virtual machine's datacenter.

#### deviceNumber

The optical drive number on the VM.

For example, **CD/DVD Drive 1** would be **deviceNumber 1**.

#### deviceType

The type of optical device. The valid values are:

- **datastore** - Uses an ISO image from a datastore. Requires the **dataStore** and **diskPath** inputs to be populated.
- **hostDevice** - Uses the optical device of the ESX/ESXi host system. Optionally, the **hostDevice** input can specify the desired host device. Otherwise, the first available device is used.
- **clientEmulate** - Sets the device type to **Client Device** in **Emulate IDE** mode.
- **clientPassthrough** - Sets the device type to **Client Device** in **Passthrough IDE** mode.



### dataStore

The datastore to use to find the ISO image if you specify a value of **datastore** for the **deviceType** input.

For example, **host:dsname, mydatastore**.

### diskPath

The path on the datastore to use for the ISO image. Include the .iso suffix; do not include a leading “/”. This ISO image must already exist on the datastore.

### hostDevice

The device path of the optical drive on host system. Use this input if you specify a value of **hostDevice** for the **deviceType** input. If you do not specify a value for this input, the first available optical drive is used.

For example, **/dev/scd00**.

### connectAtPowerOn

Specifies whether to connect the optical drive at the power on time of the VM. This input is not applicable for client device types.

The valid values are **true** and **false**.

## Results

The operation returns the following:

### returnResult

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Edit VM Boot Options

The **Edit VM Boot Options** operation modifies the boot options on a virtual machine. Optional inputs that are left empty will not be modified on the VM.

### Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#).

### taskTimeout

The time to wait in seconds before the operation is considered to have failed.

The default value is **800**.

#### `vmDatacenter`

The virtual machine's datacenter.

#### `bootRetry`

Specifies whether the VM will try again if it fails to boot after the value of the **bootRetryDelaySeconds** input has passed.

The valid values are **true** (to have the VM try again) and **false**.

#### `bootRetryDelaySeconds`

The number of seconds to wait between boot retry attempts. This input is ignored if you specify a value of **false** or no value for the **bootRetry** input.

The default value is **10**.

The valid value is any integer greater than or equal to 10.

#### `bootDelaySeconds`

The delay in seconds to wait before starting the boot sequence after the VM is powered on or reset.

The valid values are **0** through **10** seconds.

#### `enterBiosSetup`

Specifies whether the VM will automatically enter the BIOS setup menu the next time it boots.

The valid values are **true** and **false**.

## Results

The operation returns the following:

#### `returnResult`

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter client.
- The inventory path is case sensitive.

## Edit VM CPUID Masks

The **Edit VM CPUID Masks** operation modifies the CPUID masks of a VM per register level.

## Inputs

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

### taskTimeout

The time to wait in seconds before the operation is considered to have failed.

The default value is **800**.

### vmDatacenter

The virtual machine's datacenter.

### level

The CPU register level for which to edit masks. If the same masks need to be edited across multiple levels, run this operation against each level.

The valid values are **0x80000001**, **0x0**, and **0x1**.

### amdOverride

Specifies whether the AMD override masks are to be edited.

The valid values are **true** and **false**.

### eax

The eax register CPUID mask. See the note below on CPUID mask formatting.

### ebx

The ebx register CPUID mask. See the note below on CPUID mask formatting.

### ecx

The ecx register CPUID mask. See the note below on CPUID mask formatting.

### edx

The edx register CPUID mask. See the note below on CPUID mask formatting.

## Results

The operation returns the following:

### returnResult

The task or operation result.



- For inventory path formatting:
  - If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
  - If the host is a vCenter, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter client.
  - The inventory path is case sensitive.
- CPUID mask format: A CPUID mask consists of eight four-character groups delimited by colons (:). The valid characters and their meaning are as follows:

- (hyphen) - Use the default value.
- x - Unused by guest software.
- T – A feature that the guest software required to be enabled.
- F – A feature that the guest software required to be disabled.
- 1 – A feature will be reported as enabled if queried by the guest software.
- 0 – A feature will be reported as disabled if queried by the guest software.
- R – A feature will be reported as disabled if queried by the guest software except for vMotion where the actual value of this feature is required to be the same on both hosts.
- H - Used by guest software. For vMotion the value of the feature is required to be the same on both hosts. Example value: ----:----:----:----:----

## Edit VM Power Options

The **Edit VM Power Options** operation modifies the power options on a virtual machine. Optional inputs that are left empty are not modified on the VM.

### Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#).

#### taskTimeout

The time to wait in seconds before the operation is considered to have failed.

The default value is **800**.

#### vmDatacenter

The virtual machine's datacenter.

#### powerOff

The default power off behavior for the VM.

The valid values are **hard** and **soft**. If you specify a value of **hard**, power off will use the virtual hardware to power off. If you specify a value of **soft**, power off shuts down the guest operating system.

#### reset

The default reset behavior for the VM.

The valid values are **hard** and **soft**. If you specify a value of **hard**, the virtual hardware is reset. If you specify a value of **soft**, the guest operating system is rebooted.

#### guestStandby

The behavior of the VM when it receives a S1 ACPI call (standby) from the guest operating system.

The valid values are **checkpoint** and **powerOnSuspend**. If you specify a value of **checkpoint**, the guest operating system is put into standby mode and the VM remains powered on. If you specify a value of **powerOnSuspend**, the VM is suspended.

## Results

The operation returns the following:

### `returnResult`

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Edit VM Settings

The **Edit VM Settings** operation modifies advanced settings for a virtual machine. Optional inputs that are left empty will not be modified on the VM.

### Inputs

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

#### `taskTimeout`

The time to wait in seconds before the operation is considered to have failed.

The default value is **800**.

#### `vmDatacenter`

The virtual machine's datacenter.

#### `guestOSID`

The guest operating system ID. You can retrieve a list of valid entries using the **GetOSDescriptors** operation.

For example, **win95Guest** or **winNetEnterprise64Guest** or operating system description (**Microsoft Windows 3.1**, **Microsoft Windows Vista (64-bit)**, **Red Hat Enterprise Linux 3 (64-bit)**).

#### `enableLogging`

The logging setting for the VM.

The valid values are **true** and **false**. If you specify a value of **true**, logging is enabled for the VM. If you specify a value of **false**, logging is disabled.

#### `disableVideoAccel`

The console video acceleration setting for the VM.

The valid values are **true** and **false**. If you specify a value of **true**, video acceleration is disabled. If you specify a value of **false**, video acceleration is enabled.

#### `memHotAdd`

Specifies whether to allow memory hot add for the VM. This is only supported with virtual hardware 7 or greater.

The valid values are **true** and **false**.

#### `cpuHotAdd`

Specifies whether to allow cpu hot add for the VM. This is only supported with virtual hardware 7 or greater.

The valid values are **true** and **false**.

#### `cpuHotRemove`

Allow cpu hot removal for the VM. This is only supported with virtual hardware 7 or greater.

The valid values are **true** and **false**. If you specify a value of **true**, **cpuHotAdd** input is forced to **true**.

#### `swapPlacement`

The preferred location of the VM swap file.

The valid value is **inherit** which honors the virtual machine swapfile placement policy of the compute resource that contains this virtual machine.

#### `vmDirectory`

Specifies whether to store the swapfile in the same directory as the virtual machine.

#### `hostLocal`

Specifies whether to store the swapfile in the datastore specified by the **localSwapDatastore** property of the virtual machine's host.

#### `virtualCpuMmu`

The CPU instruction set and MMU virtualization preference.

The valid value is **auto** which automatically enables the hardware instruction set or MMU virtualization if available.

#### `software`

Specifies the always used software for the instruction set and MMU virtualization.

#### `instructionSet`

Specifies to use the instruction set hardware virtualization and software MMU virtualization.

#### `instructionSetMMU`

Specifies to use both the instruction set and MMU hardware virtualization.

## Results

The operation returns the following:

### `returnResult`

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter client.
- The inventory path is case sensitive.

## Edit VM Tools Options

The **Edit VM Tools Options** operation modifies when VM tools can execute scripts on a guest operating system. It can also modify if the guest time should be synchronized with the host system. Optional inputs that are left empty will not be modified on the VM.

## Inputs

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

### `taskTimeout`

The time to wait in seconds before the operation is considered to have failed.

The default value is **800**.

### `vmDatacenter`

The virtual machine's datacenter.

### `afterPowerOn`

Specifies whether to run VM tools scripts after powering on the VM.

The valid values are **true** and **false**.

### `afterResume`

Specifies whether to run VM tools scripts after resuming the VM.

The valid values are **true** and **false**.

### `beforeGuestReboot`

Specifies whether to run VM tools scripts before rebooting the guest OS.

The valid values are **true** and **false**.

### `beforeGuestShutdown`

Specifies whether to run VM tools scripts before shutting down the guest operating system.

The valid values are **true** and **false**.

#### beforeGuestStandby

Specifies whether to run VM tools scripts before suspending the guest operating system.

The valid values are **true** and **false**.

#### syncTime

Specifies whether to synchronize guest time with the host system.

The valid values are **true** and **false**.

### Results

The operation returns the following:

#### returnResult

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter client.
- The inventory path is case sensitive.

### Get Details of Disk on VM

The **Get Details of Disk on VM** operation retrieves details about an existing hard disk on a virtual machine (VM).

#### Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#).

#### vmDatacenter

The virtual machine's datacenter.

#### deviceNumber

The number of the hard disk on a virtual machine (VM) about which to get details.

For example, if a VM has a hard disk named **Hard disk 1**, the **deviceNumber** would be **1**.

### Results

The operation returns the following:



#### returnResult

The task or operation result.

#### allocatedSize

The size of the hard disk backing up files in bytes. If the disk is thin provisioned, this value is lower than the provisioned size (**vmDiskSize**).

#### diskPath

The path to the disk on the datastore.

#### dataStore

The name of the datastore that contains this disk.

#### diskMode

The disk persistence mode.

The valid values are **persistent**, **independent\_persistent**, and **independent\_nonpersistent**.

#### thinProvision

Specifies whether the hard disk is thinly provisioned.

The valid values are **true** and **false**.

#### uuid

The universally unique identifier (UUID) of the disk.

#### vmDiskSize

The size of the hard disk in bytes. If the hard disk is thinly provisioned, this is the provisioned size.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Get Details of NIC on VM

The **Get Details of NIC on VM** operation retrieves details about an existing network interface card (NIC) on a virtual machine.

### Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#).

#### vmDatacenter

The virtual machine's datacenter.

#### deviceNumber

The NIC number to get the details of on the VM.

For example, if a VM has a network adapter named **Network Adapter 1** the **deviceNumber** would be **1**.

### Results

The operation returns the following:

#### returnResult

The task result or operation result.

#### connectAtPowerOn

Specifies whether the network adapter is automatically connected to the network when the VM is powered on.

The possible values are **true** and **false**.

#### connect

Specifies whether the network adapter is currently connected to the network.

The possible values are **true** and **false**.

#### netPortGroup

The name of the port group to which the network adapter is attached. This can be a legacy port group or a distributed virtual port group.

#### netMacAddr

The MAC address of the network adapter.

An example format is **00:50:56:94:AB:EF**.

#### netMacGeneration

The MAC address assignment method configured for the network adapter.

The possible values are **assigned**, **generated**, and **manual**.

#### netWakeOnLan

Specifies whether **Wake on LAN** has been enabled for this network adapter.

The possible values are **true** and **false**.

#### netNicType

The type of network adapter.

The possible values are **PCNet32**, **E1000**, **VmxNet**, **VmxNet2**, and **VmxNet3**.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Get Details of SCSI Controller on VM

The **Get Details of SCSI Controller on VM** operation retrieves details about an existing SCSI controller on a virtual machine (VM).

### Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#).

#### deviceNumber

The number of the SCSI controller on the VM for which to get details.

For example, if a VM has a hard disk named **SCSI Controller 0**, the **deviceNumber** would be **0**.

### Results

The operation returns the following:

#### returnResult

The task or operation result.

#### diskControllerType

The SCSI controller type.

The possible values are **bus** (BusLogic Parallel), **lsi** (LSI Logic Parallel), **sas** (LSI Logic SAS), and **paravirtual** (VMware Paravirtual).

#### diskControllerHotAddRemove

Specifies whether hard disks can be added and removed while the VM is powered on.

The possible values are **true** and **false**.

#### diskControllerUnitNumber

The unit number of the SCSI controller (since it is part of the SCSI bus).

For example, **7**.

#### diskSharing

The disk sharing policy for this SCSI controller.

The possible values are **noSharing**, **physicalSharing**, and **virtualSharing**.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Get VM CPU Count

The **Get VM CPU Count** flow retrieves the configured number of CPUs for a virtual machine.

### Inputs

All of the operation's inputs are described in *Common Inputs in the Integration*.

### Results

The operation returns the following:

#### vmCpuCount

The number of CPUs configured for the virtual machine.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Get VM Memory Size

The **Get VM Memory Size** flow retrieves the configured memory size for a virtual machine in megabytes.

### Inputs

All of the operation's inputs are described in *Common Inputs in the Integration*.

### Results

The operation returns the following:

### vmMemorySize

The number of megabytes of memory configured for the virtual machine.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Remove Disk From VM

The **Remove Disk From VM** operation removes a virtual disk from an existing virtual machine.

### Inputs

All of the operation's inputs except the following are described in [Common Inputs in the Integration](#).

#### taskTimeOut

The time to wait before the operation is considered to have failed (in seconds).

#### vmDatacenter

The virtual machine's datacenter.

#### deviceNumber

The disk number as seen from the VMware client (such as **Hard Disk 1** or **Hard Disk 2**).

For example, **Hard disk 1** would be **deviceNumber 1**.

#### diskUnlink

Specifies whether to delete the virtual disk file from the datastore.

The valid values are **true** and **false**.

The default value is **false**.

### Results

The operation returns the following:

#### returnResult

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of**

**datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.

- The inventory path is case sensitive.

## Remove Floppy Drive From VM

The **Remove Floppy Drive From VM** operation removes a floppy drive from a virtual machine.

### Inputs

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

#### taskTimeOut

The time to wait in seconds before the operation is considered to have failed.

The default value is **800**.

#### vmDatacenter

The Virtual machine's datacenter.

#### deviceNumber

The number of the floppy drive to remove from the VM.

For example, if a VM has a floppy drive named **Floppy drive 1**, the **deviceNumber** would be **1**.

### Results

The operation returns the following:

#### returnResult

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter client.
- The inventory path is case sensitive.

## Remove NIC From VM

The **Remove NIC From VM** operation removes a Network Interface Card from a virtual machine.

## Inputs

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

### taskTimeout

The time to wait before the operation is considered to have failed (in seconds).

### vmDatacenter

The virtual machine's datacenter.

### deviceNumber

The number of the NICs (such as **Network adapter 1** or **Virtual Network Adapter**).

For example, **Network adapter 1** would be **deviceNumber 1**.

## Results

The operation returns the following:

### returnResult

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Remove Optical Drive From VM

The **Remove Optical Drive From VM** operation removes an optical drive (CD/DVD) from a virtual machine.

## Inputs

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

### taskTimeout

The time to wait in seconds before the operation is considered to have failed.

The default value is **800**.

### vmDatacenter

The virtual machine's datacenter.

### deviceNumber

The number of the optical drive to remove from the VM.

For example, if a VM has an optical drive named **CD/DVD drive 1** the **deviceNumber** would be **1**.

## Results

The operation returns the following:

### returnResult

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter client.
- The inventory path is case sensitive.

## Set VM CPU Count

The **Set VM CPU Count** operation sets the number of CPUs for a virtual machine. The virtual machine cannot be in a suspended power state. CPU hot plug is supported if it is enabled for the virtual machine.

## Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

### taskTimeout

The time to wait before the operation is considered to have failed (in seconds).

### vmDatacenter

The virtual machine's datacenter.

### vmCpuCount

The new number of CPUs for the virtual machine.

The maximum value for **vmCpuCount** is based on the virtual hardware version of the virtual machine. The limits are as follows:

- Virtual hardware 7 (ESXi 4.0): 8
- Virtual hardware 4 (ESXi 3.x): 4
- Virtual hardware 3 (ESXi 2.5): 2



## Results

The operation returns the following:

### returnResult

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Set VM Memory Size

The **Set VM Memory Size** operation sets the memory size for a virtual machine. The virtual machine cannot be in a suspended power state. Memory hot add is supported if it is enabled for the virtual machine.

## Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

### taskTimeout

The time to wait before the operation is considered to have failed (in seconds).

### vmDatacenter

The virtual machine's datacenter.

### vmMemorySize

The new memory size for the virtual machine in megabytes.

The maximum value for **vmMemorySize** is based on the virtual hardware version of the virtual machine. The limits are as follows:

- Virtual hardware 7 (ESXi 4.0): 261120
- Virtual hardware 4 (ESXi 3.x): 65532
- Virtual hardware 3 (ESXi 2.5): 3600

## Results

The operation returns the following:

### returnResult

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Virtual Machines

### Clone Virtual Machine

The **Clone Virtual Machine** operation clones a virtual machine. The virtual machine may be a template VM. The source virtual machine is specified using the **vmIdentifierType** and **virtualMachine** (optionally **vmDatacenter**) inputs while the target configuration is defined by **vmName**, **vmFolder**, **vmResourcePool**, **dataStore**, **hostSystem**, and **clusterName**. **Note:** This operation does not work across virtual datacenters.

#### Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

##### taskTimeOut

The time to wait before the operation is considered to have failed (in seconds).

The default value is **800**.

##### vmName

The name of the new virtual machine to create.

##### vmFolder

The virtual machine's inventory folder. Folder names are delimited by a forward slash "/". This input is case sensitive. It is only supported when the host is a vCenter. For the root folder, use "/".

For example, **/Hewlett Packard/Operations Orchestration/Templates and VMs/**.

##### vmResourcePool

The resource pool for the new virtual machine. Resource pool names are delimited by a forward slash "/". For the root resource pool, specify "Resources" or a single "/".

##### dataStore

The name of the datastore to store the new virtual machine. If you do not specify a value for this input, the datastore of the source virtual machine is used.

For example, **host:dsname** or **mydatastore**.

### hostSystem

The name of destination the host system (ESX or ESXi) for new the virtual machine as seen in the vCenter UI. This input is only supported when the host is a vCenter. If you do not specify a value for this input, the host system of the source virtual machine is used.

### clusterName

The name of the VMware HA or DRS cluster. You can specify this input instead of **hostSystem** if the desired destination ESX(i) host is in a DRS or HA cluster.

### markAsTemplate

Specifies whether to mark the virtual machine as a template.

The valid values are **true** and **false**. If you specify **true**, the virtual machine is marked as a template. Otherwise, it is marked as a regular virtual machine.

### customizationTemplateName

The name of the customization specification to apply while creating this clone. The customization specification should already exist in the vCenter customization specifications manager.

## Results

The operation returns the following:

### returnResult

The task result or operation result.

## Clone Virtual Machine to Template

Clone a virtual machine to template. The source virtual machine is specified via `vmIdentifierType` and `virtualMachine`, while the clone is defined by `templateName`, `templateFolder`, `dataStore`, `hostSystem`, and `clusterName`.

## Inputs

### taskTimeOut

Time to wait before the operation is considered to have failed (seconds).

### vmIdentifierType

Virtual machine identifier type.

### virtualMachine

Primary Virtual Machine identifier of the virtual machine to clone from. Inventory path (Datacenter/vm/Folder/MyVM), Name of VM, IP (IPv4 or IPv6 depending upon ESX version), hostname (full), UUID, or the VM id (vm-123,123).

### vmDatacenter

Virtual machine's datacenter. If host is ESX(i) use "ha-datacenter".

#### templateName

Name of new virtual machine template being created.

#### templateFolder

Virtual machine template's inventory folder. Folder names are delimited by a forward slash "/". This input is case sensitive. Only supported when host is a vCenter. For root folder, use "/".

#### dataStore

Name of datastore to store new template (eg. host:dsname, mydatastore). If not specified the same datastore of the source virtual machine will be used.

#### hostSystem

Name of destination host system (ESX or ESXi) for new template as seen in the vCenter UI. Only supported when host is a vCenter. If not specified the same host system of the source virtual machine will be used.

#### clusterName

Name of the VMWare HA or DRS cluster. Can be specified instead of hostSystem if the desired destination ESX(i) host is in a DRS or HA cluster.

#### customizationTemplateName

Name of the customization specification to apply while creating this clone. The customization specification should already exist in the vCenter customization specifications manager. If both 'customizationTemplateName' and 'customizationSpecXml' have input, the flow will fail.

#### customizationSpecXml

The Xml string of the customization spec.

### Create Linked VM

Creates linked clone vm from the specified snapshotName or from the current running state of the vm. The source virtual machine is specified via vmIdentifierType and virtualMachine (optionally vmDatacenter) while the linked clone is defined by vmName and vmFolder.

#### Inputs

##### taskTimeout

Time to wait before the operation is considered to have failed (seconds).

##### vmIdentifierType

Virtual machine identifier type

##### virtualMachine

Primary Virtual Machine identifier. Inventorypath (Datacenter/vm/Folder/MyVM), Name of VM, IP (IPv4 or IPv6 depending upon ESX version), hostname (full), UUID, or the VM id (vm-123,123).

#### vmDatacenter

Virtual machine's datacenter.

#### snapshotName

The name of the snapshot to create a linked clone vm from. This is an optional input. If not specified then, creates a linked clone vm from the current running state of the vm.

#### vmName

Name of the new virtual machine being created.

#### vmFolder

Virtual machine's inventory folder. Folder names are delimited by a forward slash "/". This input is case sensitive. Only supported when host is a vCenter. For root folder, use "/".

### Results

returnResult - Task result or operation result.

## Create Virtual Machine

Create a virtual machine.

### Inputs

#### taskTimeout

Time to wait before the operation is considered to have failed (seconds).

#### vmName

Virtual machine name being created (new).

#### vmCpuCount

Number of virtual machine CPUs to setup.

#### hostSystem

Target virtual machine host system.

#### clusterName

Name of the VMWare HA or DRS cluster.

#### description

Description / annotation.

#### dataStore

Datastore (eg. host:dsname, mydatastore).

#### guestOSID

Guest OS ID. A list of valid entries can be retrieved via the GetOSDescriptors operation.

#### vmDatacenter

Virtual machine's datacenter.

#### vmMemorySize

Virtual machine memory size (megabytes).

#### vmResourcePool

Virtual machine's resource pool.

#### vmFolder

Virtual machine's folder by inventory path, / delimited not including datacenter (eg. ManagedVMs/DRS/Location1). For root folder, use /.

#### vmDiskSize

Size of the virtual disk to create (MB).

### Results

returnResult - Task result or operation result.

## Create VM Folder

The **Create VM Folder** operation creates a new inventory folder in a datacenter as seen in the vCenter **Inventory > VMs and Templates** view.

### Inputs

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

#### vmDatacenter

The datacenter in which to create the new VM inventory folder.

#### vmFolder

The path of the VM inventory folder to create. Folder names are delimited by a forward slash "/". This input creates parent folders if they do not already exist, and is case sensitive.

For example, **/Hewlett Packard/Operations Orchestration/Templates**.

### Results

The operation returns the following:

#### returnResult

This is the primary output. It contains a descriptive result of the operation or the reason for the error if the operation fails.

## Delete VM Folder

The **Delete VM Folder** operation deletes an existing inventory folder in a datacenter as seen in the **vCenter Inventory > VMs and Templates** view. Optionally, this operation can also delete any VMs contained in the specified folder if it is not empty.

### Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

#### vmFolder

The path of the VM inventory folder to delete (for example, **/Hewlett Packard/Operations Orchestration/Templates**). Folder names are delimited by a forward slash "/". This input is case sensitive.

#### removeChildren

Specifies whether to recursively delete any child folders and VMs.

The valid values are **true** and **false**. If you specify a value of **false**, the folder must be empty before it can be deleted. If you specify a value of **true**, the folder and all of its contents are deleted.

### Results

The operation returns the following:

#### returnResult

This is the primary output. It contains a descriptive result of the operation or the reason for the error if the operation fails.

## Deploy Virtual Machine from Template

Deploy a virtual machine from template. The source is the templateName while the deployment is defined by vmName, vmFolder, vmResourcePool, dataStore, hostSystem, and clusterName.

### Inputs

#### host

VMware vCenter, ESXi host hostname or IP.

#### user

VMware username.

#### password

VMware user's password.

#### port

Port on which to connect.

For example, **443**, **80**

#### Protocol

Connection protocol. The valid values are **HTTP** and **HTTPS**.

#### closeSession

Close the internally kept VMware Infrastructure API session upon completion of operation. The valid values are **true** and **false**.

#### Asyn

Asynchronously perform the task. The valid values are **true** and **false**

#### taskTimeOut

The time to wait before the operation is considered to have failed (in seconds). The default value is **800**.

#### templateName

The name of the template from which to deploy.

#### vmDatacenter

The virtual machine's datacenter. If host is ESX(i), use "ha-datacenter".

#### vmFolder

Virtual machine's inventory folder. Folder names are delimited by a forward slash "/". This input is case sensitive. Only supported when a host is a vCenter. For root folder, use "/".

For example: /Hewlett Packard/Operations Orchestration/VMs

#### vmResourcePool

Resource pool for new virtual machine. Resource pools are delimited by a forward "/". For the root resource pool, specify "Resources" or a single "/".

#### dataStore

Name of datastore to store new virtual machine (for example, host:dsname, mydatastore). If not specified, the same datastore of the source virtual machine will be used.

#### hostSystem

Name of the destination host system (ESX or ESXi) for the new virtual machine as seen in the Vcenter UI. Only supported when host is a vCenter. If not, specified, the same host system of the source virtual machine will be used.

#### clusterName

Name of the VMware HA or DRS cluster. Can be specified instead of hostSystem if the desired destination (ESXi) host is in a DRS or HA cluster.

#### customizationTemplateName

Name of the customization specification to apply while creating this clone. The customization specification should already exist in the vCenter customization specifications manager. If both "customizationTemplateName" and "customizationSpecXml" have input, the flow will fail.

#### customizationSpecXml

The XML string of the customization spec.



## Responses

The operation's responses are:

### success

The operation completed successfully.

### failure

Something went wrong.

**Note:** The result of the VMware Virtual Infrastructure and vSphere/Guest/Export Guest Customization spec operation can be used to get the customizationSpecXml input value.

## Export VM as OVF

Exports a Virtual Machine as an Open Virtualization Format (OVF) package in a folder of files format to the local filesystem of the RAS.

## Inputs

### vmDatacenter

Datacenter from where to get virtual machines.

### hostSystem

Host system to get virtual machines from.

### vmIdentifierType

Virtual machine identifier type. Required if virtualMachine input is used.

### virtualMachine

Primary Virtual Machine identifier. Inventorypath (Datacenter/vm/Folder/MyVM), Name of VM, IP (IPv4 or IPv6 depending upon ESX version), hostname (full), UUID, or the VM id (vm-123,123).

### path

Local directory path on the RAS to write the ovf template files. The directory must already exist.

### parallel

If the virtual machine has multiple .vmdk files, should they be downloaded in parallel? If true, all .vmdk files will be downloaded using separate threads.

## Results

### returnResult

Operation result or reason for failure.

### ovfPath

The path to the .ovf file that was generated on the local filesystem of the RAS.

## Get Virtual Machine

The **Get Virtual Machine** operation retrieves information about a virtual machine or a group of virtual machines in a datacenter or host. The operation retrieves a pipe (|) delimited list of objects one at a time from an initial search query using a SQL Query operation.

### Inputs

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

#### vmDatacenter

The virtual machine's datacenter.

#### getDetailed

Specifies whether to retrieve detailed information about the virtual machine.

The valid values are **true** and **false**.

### Results

The operation returns the following:

#### vmInfo

The virtual machine information.

#### numberOfResults

The number of results from the query remaining to be retrieved.

#### returnResult

The virtual machine ID or operation results.

#### name

The name of the virtual machine.

#### uuid

The UUID of the virtual machine.

#### inventoryPath

The inventory path of the virtual machine.

#### dataStore

The name of the datastore where the VM configuration file (.vmx) is stored.

#### vmxPath

The path to the vm configuration file (.vmx) on datastore.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Mark VM As Template

The **Mark VM As Template** operation marks a virtual machine as a template or regular virtual machine. This operation does NOT deploy a virtual machine from a template. If you mark a VM as a regular VM, you must enter a value for the **resourcePool** and **hostSystem** inputs.

### Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

#### vmDatacenter

The virtual machine's datacenter.

#### hostSystem

The target virtual machine host system or cluster.

#### markAsTemplate

Specifies whether the virtual machine should be marked as a template. If you do not enter a value for the **markAsTemplate** input, it is marked as a regular VM.

The valid values are **true** and **false**.

### Results

The operation returns the following:

#### returnResult

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Migrate Virtual Machine

The **Migrate Virtual Machine** operation migrates a virtual machine between one host system and another. You can migrate a VM cold (powered off) or hot (powered on). Powered on implies the use of VMotion. This does NOT move a virtual machine's virtual disk files.

The source virtual machine is defined using the **vmIdentifierType**, **virtualMachine**, and, optionally, the **vmDatacenter** inputs. The target location of the VM is defined by the **resourcePool** and **hostSystem** inputs. **Note:** This operation does not work across virtual datacenters.

### Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

#### taskTimeout

The time to wait before the operation is considered to have failed (in seconds).

#### vmDatacenter

The virtual machine's datacenter.

#### vmPowerState

The virtual machine power state to set.

#### hostSystem

The target virtual machine host system.

#### priority

The migration or relocation priority.

The valid values are **defaultPriority**, **highPriority**, and **lowPriority**.

### Results

The operation returns the following:

#### returnResult

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Move Virtual Machine

The **Move Virtual Machine** operation moves a virtual machine to a different inventory folder, resource pool, host system, cluster, or datastore. The destination of the VM depends on the optional inputs you specify. This operation does not work across virtual datacenters.

### Inputs

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

#### vmDatacenter

The virtual machine's datacenter. If the host is ESX(i), specify a value of **ha-datacenter**.

#### vmFolder

The path of the new inventory folder for VM. Folder names are delimited by a forward slash "/". This input is case sensitive. This input is only supported when the host is a vCenter.

For example, **/Hewlett Packard/Operations Orchestration/Templates**.

#### resourcePool

The destination resource pool for the VM relative to the host system or cluster. Resource pool names are delimited by a forward slash "/". For the root resource pool, specify "Resources" or a single "/".

#### hostSystem

The name of the destination host system for the VM. This input is only supported when the host is a vCenter.

#### dataStore

The name of the destination datastore.

For example, **host:dsname, mydatastore**.

### Results

The operation returns the following:

#### returnResult

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter client.
- The inventory path is case sensitive.

## Promote VM Disk

Promotes disks on this virtual machine that have delta disk backings.

## Inputs

### taskTimeout

Time to wait before the operation is considered to have failed (seconds).

### vmIdentifierType

Virtual machine identifier type

### virtualMachine

Primary Virtual Machine identifier. Inventory path (Datacenter/vm/Folder/MyVM), Name of VM, IP (IPv4 or IPv6 depending upon ESX version), hostname (full), UUID, or the VM id (vm-123,123).

### vmDatacenter

Virtual machine's datacenter.

## Results

returnResult - Task result or operation result.

## Register Virtual Machine Template

The **Register Virtual Machine Template** operation registers an existing .vmtx file from a datastore as a virtual machine template.

## Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

### taskTimeout

The time to wait before the operation is considered to have failed (seconds).

The default value is **800**.

### vmDatacenter

The virtual machine's datacenter.

### dataStore

The datastore.

For example, **host:dsname** or **mydatastore**.

### vmxPath

The Path to the .vmtx file on the datastore.

For example, **Name of VM Template/Name of VM Template.vmtx**.

#### vmName

The display name of the virtual machine template being registered. If you do not specify a value for this input, the display name from the .vmx file is used.

#### vmFolder

The path of inventory folder for the VM template. Folder names are delimited by a forward slash "/". This input is case sensitive. For the root folder, use "/".

For example, **/Hewlett Packard/Operations Orchestration/VM Templates**.

#### hostSystem

The name of host system on which to register VM template.

### Results

The operation returns the following:

#### returnResult

The task or operation result.

## Register Virtual Machine

The **Register Virtual Machine** operation registers a existing .vmx file from a datastore as a virtual machine.

### Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

#### taskTimeout

The time to wait before the operation is considered to have failed (seconds).

The default value is **800**.

#### vmDatacenter

The virtual machine's datacenter. If the host is ESX(i), use the value **ha-datacenter**.

#### dataStore

The datastore.

For example, **host:dsname** or **mydatastore**.

#### vmxPath

The path to the .vmx file on the datastore.

For example, **Name of VM/Name of VM.vmx**.

#### vmFolder

The path of the inventory folder for the VM. Folder names are delimited by a forward slash "/". This input is case sensitive. For the root folder, use "/". Only the root folder is supported with ESX(i) hosts.

For example, **/Hewlett Packard/Operations Orchestration/VMs**.

#### vmName

The display name of the virtual machine being registered. If you do not specify a value for this input, the display name from the .vmx file is used.

#### hostSystem

The name of host system on which to register the VM. You can only specify a value for the **hostSystem** or **clusterName** input. You cannot specify both.

#### clusterName

The name of the cluster on which to register the VM. This input is only valid when host is a vCenter. You can only specify a value for the **hostSystem** or **clusterName** input. You cannot specify both.

#### vmResourcePool

The virtual machine's resource pool relative to the host system or cluster.

### Results

The operation returns the following:

#### returnResult

The task or operation result.

## Relocate Virtual Machine

The **Relocate Virtual Machine** operation relocates a virtual machine's associated virtual disks and information to a different host system, resource pool, and datastore. **Note:** This operation does not work across virtual datacenters.

### Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

#### taskTimeout

The time to wait before the operation is considered to have failed (in seconds).

#### vmDatacenter

The virtual machine's datacenter.

#### hostSystem

The target virtual machine host system.

#### dataStore

The datastore.

For example, **host:dsname** or **mydatastore**.



## Results

The operation returns the following:

### returnResult

The task or operation result.



For inventory path formatting:

- If the host is an ESXi server, the inventory path will be **ha-datacenter/vm/<name of vm>**.
- If the host is a vCenter or vSphere server, the inventory path will be **<name of datacenter>/vm/<folders>/<name of vm>**. The **<folders>/<name of vm>** part of the path is based on the **Virtual Machines & Templates** view in the vCenter or vSphere client.
- The inventory path is case sensitive.

## Rename VM Folder

The **Rename VM Folder** operation renames an inventory folder in a datacenter as seen in vCenter **Inventory > VMs and Templates** view.

## Inputs

All of the flow's inputs except the following are described in [Common inputs in the integration](#).

### taskTimeout

The time to wait in seconds before the operation is considered to have failed.

The default value is **800**.

### vmDatacenter

The datacenter in which to rename the VM inventory folder.

### vmFolder

The path of the VM inventory folder to rename. Folder names are delimited by a forward slash "/". This input is case sensitive.

For example, **Hewlett Packard/Operations Orchestration/Templates**.

### name

The new name of the folder.

For example, if you specify a value of **Windows Templates**, the operation would rename **Hewlett Packard/Operations Orchestration/Templates** to **Hewlett Packard/Operations Orchestration/Windows Templates**.

## Results

The operation returns the following:

`returnResult`

This is the primary output. It contains a descriptive result of the operation or the reason for the error if the operation fails.

## Unregister Virtual Machine

The **Unregister Virtual Machine** operation unregisters a virtual machine or virtual machine template from a vCenter or ESX(i) host. The .vmx or .vmtx file path and datastore are returned as results for future registration of the virtual machine or virtual machine template.

### Inputs

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

`vmDatacenter`

The datacenter from which to get virtual machines.

### Results

The operation returns the following:

`returnResult`

The operation result or the reason for failure.

`dataStore`

The name of the datastore where the VM configuration file (.vmx or .vmtx) is stored.

`vmxPath`

The path to the VM configuration file (.vmx or .vmtx) on the dataStore.

## VM is template

Checks to see if the virtual machine is template.

### Inputs

`vmDatacenter`

Datacenter from where to get virtual machine.

`vmIdentifierType`

Virtual machine identifier type (inventorypath, name, ip, hostname, uuid, vmid).

`virtualMachine`

Primary Virtual Machine identifier. Inventorypath (Datacenter/vm/Folder/MyVM), Name of VM, IP (IPv4 or IPv6 depending upon ESX version), hostname (full), UUID, or the VM id (vm-123,123).

## Results

isTemplate - Returns true if the virtual machine is template, else false.

# Hidden Inputs in the Integration

Hidden inputs are inputs that are available for an operation, but are not documented. You can add these inputs on the **Inputs** tab of the operation. This section lists any of the integration's operations that use hidden inputs and describes those hidden inputs.

## Get Task State

### `_overrideProperties`

Used to override the default properties retrieved by the operation from a Task object. This input is a comma-delimited list of managed object reference properties as seen in the *VMware Virtual Infrastructure Programming Guide SDK*. For an example of the format, see the “Search For Virtual Machines” step in the **Get List of Powered On VMs** sample flow.

## Create Virtual Machine

### `netNicType`

The virtual network interface card (NIC) type. By default, the virtual NIC device is **PCNet32**. The valid values are **E1000**, **PCNet32**, and **Vmxnet**, all case insensitive. By default, virtual machines are created with **PCNet32** NICs because it is the only NIC type that is supported when a virtual machine does not have VMware Tools installed.

---

## 4 Creating Custom VMware Virtual Infrastructure Integration Operations

This section includes the following topics:

- Customizing the Integration
  - Using the Web Services Wizard
  - Using OO IActions
  - Using Third-Party Tools

## Customizing the Integration

The VMware Infrastructure integration should work if the following requirements are met:

- The ESXi host version is 2.0 or later.
- The Virtual Center host version is 1.0 or later.
- The Web service is enabled on the host and accessible from the respective OO JRAS.  
There are no customizations supported by VMware (except for access control lists) that could invalidate the provided operations or prevent them from accomplishing their tasks.

If tasks unsupported by the included operations are needed, the user must use OO's Web Service Wizard, OO's IActions, or third-party tools such as Microsoft's PowerShell. All of these methods utilize VMware's Virtual Infrastructure API, so familiarity with VMware's APIs is required.

### Using the Web Services Wizard to Create OO Operations for a Web Service

Using the OO Web Service Wizard (wswizard.exe) for custom operations requires knowledge of the VI API and VMware object hierarchy at the XML level and is generally not advised unless the integrator is comfortable reading API documentation and directly modifying XML. The OO Web Service Wizard is located in the HP OO home folder in the \Studio\tools\ folder. For instructions on using the Web Services Wizard, see the *Guide to Authoring Operations Orchestration Flows* (Studio\_UsersGuide.pdf) in the documentation set for the current OO release.

### Using OO IActions to Create OO Operations

Custom operations written using IActions require knowledge of the VI API and VMware object hierarchy as well as familiarity with either Java or a .NET language. For information on creating custom IActions, see the IAction documentation in the *OO SDK Guide* (SDKGuide.pdf) in the documentation set for the current OO release.

### Using Third-Party Tools to Create OO Operations

The third customization option allows the integrator to use any of the options that VMware provides for integrations. One popular integration method within the VMware community is to use PowerShell wrappers to script VMware's APIs. Currently, users can leverage PowerShell from OO by remotely executing commands through RAS operations.

---

## 5 Troubleshooting

This section includes the following topics:

- [Troubleshooting Overview](#)
- [General Troubleshooting Procedures and Tools](#)
- [Error Messages](#)

# 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

The best troubleshooting tool available is VMware's MOB, a Web-based application that allows you to browse a Virtual Center or ESXi host system's inventory as seen by the API. MOB also allows you to execute some Virtual Infrastructure commands from a Web browser. To use MOB, open a Web browser to the URL **https://hostsystem/mob** and enter your credentials.

To reach the parts of MOB that you will need, click the content link and go to the root folder property, which contains all the components in the virtual datacenter. Components such as alarms, events, and performance counters are available via other properties and services within the VMware SDK.

## Error Messages

### Couldn't find VM type

The **vmIdentifierType** input was left blank. VMs are identified by the identifier type (a way of specifying a single virtual machine) and a value for which to search.

### Unsupported VM reference style

The **VM identifier type** input is invalid. Pick a valid one from the selection list provided.

### Must specify VM inventory path, IP, hostname, or UUID

The **virtualMachine** input was left blank. VMs are identified by the identifier type and a value for which to search. Check the format of your input as well:

- Inventory paths are / delimited.
- IP addresses are either IPv4 or IPv6 addresses (depending upon your ESXi version).
- Host names must be fully qualified domain names.
- UUIDs must consist of hexadecimal characters and dashes.

Other options include searching by the VM's name or its internal ID number. You can see the internal ID number in MOB as either **vm-number** (in Virtual Center) or just a number (ESXi) listed under its type of **virtualMachine**.

### Datacenter "... " not found

The datacenter specified was not found. Make sure that you are accessing the correct ESXi or Virtual Center server if you are accessing an ESXi server, and that the datacenter is named **ha-datacenter** (the default name of the datacenter on a standalone ESXi server).

#### No ConfigTarget found in ComputeResource "..."

#### No VirtualHardwareInfo found in ComputeResource

The compute resource associated with a host system or cluster does not include the following, which are required for specifying the virtual hardware configuration of a VM:

- Information on virtual hardware
- Available configuration options

Make sure that the host system or specified cluster has been initialized correctly. The errors leading to this error message may occur if Virtual Center's information is not yet ready or out of date, and may require waiting for Virtual Center or possibly restarting Virtual Center.

#### Invalid nic card type "..."

The NIC type entered is not supported. Use **PCNet32**, **E1000**, or **VMXNet** (all case-insensitive) for the input.

#### Could not find volume "..."

The specified datastore could not be found. Check the format used to specify the datastore. It should not include brackets or any non-word characters.

#### No Datastore found in ComputeResource

There was no available datastore found for the host system specified. This can mean that the local datastore on the host system is corrupt or that connectivity to the VI host has been lost.

#### Couldn't find a network configuration for host system "..."

The host system does not have a virtual network configuration. This can be caused by an inconsistent state in Virtual Center or by a configuration issue with the host system that is being targeted.

#### Must bridge to at least one NIC device!

You must specify a NIC device that can be found in the host network configuration. To find the name of virtual NICs that are in the host network configuration, open the **Configuration** tab under the Virtual Infrastructure client information on a host system or via MOB.

#### No power state information found for vm

An error occurred during retrieval of information about the VM. Make sure the virtual machine has not been deleted and that the Virtual Center host is able to connect to the virtual machine.

#### Unactionable power state request

The power state change requested is not possible (for instance, **powered off** to **suspended**, **suspended** to **powered off**). Transition the VM to a suitable power state before attempting this operation.

#### No action performed. VM is currently in requested power state or target power state invalid.

You cannot put a VM in the same power state that it is already in, and you cannot put a VM in a power state that doesn't exist. Make sure the requested power state is **poweredOn**, **poweredOff**, or **suspended**.



#### Snapshot tree for VM "... " not found

##### No snapshot tree found

No snapshots were found associated with the VM. Make sure that the specified VM has a snapshot history and that none of the snapshots are corrupted.

##### Unexpected number of snapshot trees found

This can occur if a snapshot tree changes while the operation was in execution or if there has been corruption of data within Virtual Center or ESXi. Make sure that no other operations are running against the VM and try the operation again.

##### Unable to retrieve snapshot named "... "

A snapshot tree was found, but no snapshots matching the name were found. Make sure that any escaping has been completed, and if necessary, browse to the appropriate snapshot in MOB to see how it has been escaped within VI.

##### Host system "... " not found

In the datacenter specified, the specified host system was not found as connected or registered. Check the name of the host system and make sure it matches the name of an ESXi host. For example, if there is a host system named **hahost123.mydomain.com**, you should specify **hahost123.mydomain.com** rather than just **hahost123**.

##### Could not find computeResource for host: "... "

Every ESXi host is assigned a compute resource which allows the creation of resource pools within a host. This error occurs when a specific ESXi host is specified but no compute resources can be found for it. This is usually the result of a configuration error or outdated data in Virtual Center.

##### Cluster "... " not found

The specified cluster was not found in the datacenter. Check the cluster name that you specified. Note that cluster names are case-sensitive.

##### Couldn't find any compute resources!

This error occurs if there are no host systems or clusters found. This is sometimes due to Virtual Center having communication problems with its managed hosts. Make sure the VC or ESXi host is fully connected and configured correctly.

##### Couldn't find a resource pool on host system

##### Couldn't find a resource pool on cluster

A suitable cluster or host system was found, but it appears that there are no resource pools. This condition should not happen because there should be at least one resource pool, the default **Resources** pool, available to all virtual machines. Make sure the VC or ESXi host is fully connected and configured correctly.

##### You cannot specify both a host and cluster name

##### Must specify a valid hostSystem or cluster first

To execute the operation, you must specify either the host system or cluster name exactly. Specify a host system or cluster name in the appropriate input and make sure the other is blank.

#### Could not find resource pool named "..."

Although a resource pool was found, the resource pool specified was not found in the datacenter or host. Remember that:

- Resource pool identifiers are case-sensitive.
- Virtual machines cannot be assigned to resource pools outside their host system or cluster, depending upon whether the VM is on a standalone system or VMware cluster.

#### Resource pool required when deploying a VM from a template! Try "Resources" for the default resource pool

A resource pool was not specified, and use of the default resource pool **Resources** was not attempted. This error can occur because of configuration problems or if you specify a blank or whitespace for the resource pool input. Check your inputs, or refresh or restart Virtual Center.

#### No custom fields found via this ESXi / VC server. If this is not what is expected, please check that the host is a Virtual Center machine

No custom fields have been defined on the specified ESXi or VC server. Custom fields are not supported on standalone ESXi host systems and are only accessible via a VC host.

#### No virtual machines could be found meeting given criteria

The host has no virtual machines, or there are no valid virtual machines connected to the host. This error is expected when running against a new ESXi or VC host with no VMs specified.

#### AxisFault – InvalidPropertyFault "..."

This is an internal error that occurs when a VI object is retrieved and an unexpected object comes back. This is usually due to user input error (such as use of the **Advanced Search** operation with invalid or nonexistent properties), but it can also mean that there is an error in the operation that requires engineering effort.

#### AxisFault - Read time out

This occurs when an operation is waiting for an update from the VI host for too long. This can occur when there's significant load on a VI host or an operation in synchronous mode (async = false) takes a long time to finish. Synchronous tasks taking over 25 minutes are considered to be failures. Set the **async** input on any potentially long-running operation (such as **clone**, **relocate**, or **migrate**) to **true** and periodically use the **Get Task State** operation to avoid this error.

---

## 6 Security

This section includes the following topics:

- [About VMware Virtual Infrastructure Security](#)

## About VMware Virtual Infrastructure Security

VMware ESXi and Virtual Center are accessed via SOAP over HTTPS (or HTTP, if enabled on the host) in the integration and are subject to the same security principles and rules as any other Web service. Client sessions are established as session tokens separate from the HTTP protocol and can be viewed through the VMware APIs or, more conveniently, via MOB.

---

## 7 OO Tools

This section includes the following topic:

- [OO Tools You Can Use with the VMware Virtual Infrastructure-OO Integration](#)

# OO Tools You Can Use with the VMware Virtual Infrastructure Integration

Following are OO tools that you can use with the [Technology Name] integration:

- **RSFlowInvoke.exe and JRSFlowInvoke.jar**

RSFlowInvoke (RSFlowInvoke.exe or the Java version, JRSFlowInvoke.jar) is a command-line utility that allows you to start a flow without using Central (although the Central service must be running). RSFlowInvoke is useful when you want to start a flow from an external system, such as a monitoring application that can use a command line to start a flow.

- **Web Services Wizard (wswizard.exe)**

When you run the Web Services Wizard, you provide it with the WSDL for a given Web service. The WSDL string you provide as a pointer can be a file's location and name or a URL. The Web Services Wizard displays a list of the methods in the API of the Web service that you specify. When you run the wizard, pick the methods you want to use, and with one click for each method you have selected, the wizard creates an HP OO operation that can execute the method. This allows you to use the Web Services Wizard to create operations from your monitoring tool's API.

These tools are available in the Operations Orchestration home folder in /Studio/tools/.