

Custom Properties Reference

vRealize Automation 6.2

vmware[®]

You can find the most up-to-date technical documentation on the VMware Web site at:

<https://docs.vmware.com/>

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:

docfeedback@vmware.com

Copyright © 2008–2016 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

Custom Properties Reference	5
Updated Information	7
1 Using Custom Properties	9
Custom Property Uses	9
Custom Properties in Machine Provisioning, Blueprints, and Build Profiles	10
Specifying Custom Properties	10
Order of Precedence for Custom Properties	11
Specifying Custom Properties for Multi-Machine Services	12
Custom Property Types	13
Creating Custom Properties for Application Services	13
Build Profiles and Property Sets	14
Create a Build Profile	15
Create a Property Set	16
2 Custom Properties Grouped by Function	19
Custom Properties for Openstack Endpoints	20
Custom Properties for Basic Workflow Blueprints	20
Custom Properties for Clone Blueprints	22
Custom Properties for FlexClone Blueprints	25
Custom Properties for Linked Clone Blueprints	27
Custom Properties for Linux Kickstart Blueprints	29
Custom Properties for SCCM Blueprints	31
Custom Properties for WIM Blueprints	32
Custom Properties for vApp Blueprints	35
Custom Properties for vRealize Automation Guest Agent	37
Custom Properties for Networking	40
Custom Properties for PXE Provisioning	42
Custom Properties for BMC BladeLogic Configuration Manager Integration	45
Custom Properties for HP Server Automation Integration	46
3 Custom Properties Grouped by Name	49
Custom Properties A Table	49
Custom Properties B Table	50
Custom Properties C Table	51
Custom Properties E Table	52
Custom Properties H Table	52
Custom Properties I Table	53
Custom Properties L Table	54
Custom Properties M Table	55
Custom Properties O Table	55

Custom Properties P Table	56
Custom Properties R Table	57
Custom Properties S Table	57
Custom Properties V Table	59
Custom Properties X Table	76
4 Using the Property Dictionary	77
Adding Property Definitions	78
Add a Property Definition	78
Adding Property Attributes	79
Add a Help Text Attribute	80
Add an Order Index Attribute	80
Add a Relationship Attribute	81
Add a Value Expression Attribute	82
Add a Value List Attribute	82
Add a Minimum Value Attribute	83
Add a Maximum Value Attribute	83
Add an Interval Attribute	84
Example of Creating a Relationship Between Two Properties and Adding Them to a Blueprint	84
Creating Property Control Layouts	87
Add a Property Control Layout	88
Index	89

Custom Properties Reference

Custom Properties Reference provides information about the custom properties, and their use, that are available when you use VMware vRealize™ Automation.

This documentation is intended to be used with the vRealize Automation product documentation available from the VMware vRealize™ Automation Documentation page at <https://www.vmware.com/support/pubs/vcac-pubs.html>.

NOTE Not all features and capabilities of vRealize Automation are available in all editions. For a comparison of feature sets in each edition, see <https://www.vmware.com/products/vrealize-automation/>.

Intended Audience

This information is intended for IaaS administrators, fabric administrators, and business group managers of vRealize Automation. This content is written for experienced Windows or Linux system administrators who are familiar with virtualization technology and the basic concepts described in *Foundations and Concepts*.

VMware Technical Publications Glossary

VMware Technical Publications provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation, go to <http://www.vmware.com/support/pubs>.

Updated Information

This *Custom Properties Reference* is updated with each release of the product or when necessary.

This table provides the update history of *Custom Properties Reference*.

Revision	Description
001638-05	<ul style="list-style-type: none">■ Added custom property Scvmm.Generation2 to “Custom Properties S Table,” on page 57. 4■ Updated custom property Hyperv.Network.Type in “Custom Properties H Table,” on page 52.■ Added custom property VMware.Endpoint.Openstack.IdentityProvider.Version to “Custom Properties V Table,” on page 59 and “Custom Properties for Openstack Endpoints,” on page 20 to support authentication with Keystone version 3.■ Updated custom property VMware.Endpoint.Openstack.Release in “Custom Properties V Table,” on page 59 and “Custom Properties for Openstack Endpoints,” on page 20 for deprecation as of 6.2.4.■ Added VirtualMachine.Admin.ConnectAddress.Regex and VirtualMachine.NetworkN.AddressM to “Custom Properties V Table,” on page 59 and “Custom Properties for Openstack Endpoints,” on page 20.■ Added VirtualMachine.ScriptPath.Decrypt to “Custom Properties V Table,” on page 59 and “Custom Properties for vRealize Automation Guest Agent,” on page 37.■ Added the following new custom properties to “Custom Properties A Table,” on page 49:<ul style="list-style-type: none">■ amazon.IAMInstanceProfile.ARN■ AppService.SyncMachines.MachineProvisioned
001638-04	<p>Updated “Custom Properties S Table,” on page 57 to document the Snapshot.Policy.Limit default.</p> <p>Updated “Custom Properties V Table,” on page 59 to add the following custom properties:</p> <ul style="list-style-type: none">■ VirtualMachine.Admin.ExportProperties■ VirtualMachine.SoftwareN.PostInstallDelay■ VirtualMachine.SoftwareN.FailsBuild■ VirtualMachine.SoftwareN.NoQuotes
001638-03	<p>Updated “Custom Properties I Table,” on page 53 to note that values for Image.ISO.Location are case sensitive.</p>
001638-02	<ul style="list-style-type: none">■ Updated various topics to include new and updated vRealize Automation custom properties and their descriptions based on information from product development and field staff.■ Added new functional grouping “Custom Properties for vRealize Automation Guest Agent,” on page 37.■ Removed Opware.WOL.Delay and Opware.WOL.Enabled from “Custom Properties O Table,” on page 55 and “Custom Properties for HP Server Automation Integration,” on page 46.

Revision	Description
001638-01	<ul style="list-style-type: none"> ■ Updated the <code>VirtualMachine.NetworkN.Name</code> property in “Custom Properties for Networking,” on page 40 and “Custom Properties for vApp Blueprints,” on page 35. ■ Added the <code>VirtualMachine.DiskN.IsFixed</code> property to “Custom Properties V Table,” on page 59 and “Custom Properties for vApp Blueprints,” on page 35. ■ Corrected the property names <code>VirtualMachine.SoftwareN.ScriptPath</code>, <code>VirtualMachine.SoftwareN.ISOName</code> , <code>VirtualMachine.Admin.ConnectAddress</code>, and <code>VirtualMachine.SoftwareN.ISOLocation</code> in “Custom Properties V Table,” on page 59. ■ Added information to the description for property <code>VirtualMachine.Storage.ReserveMemory</code> in “Custom Properties V Table,” on page 59. ■ Corrected the property name <code>Cisco.Organization.Dn</code> in “Custom Properties C Table,” on page 51. ■ Added the following previously undocumented custom properties: <ul style="list-style-type: none"> ■ <code>ApplicationDirector.Provisioning.SkipTeardown</code> ■ <code>VirtualMachine.Admin.EncryptPasswords</code> ■ Added the following new properties to “Custom Properties V Table,” on page 59: <ul style="list-style-type: none"> ■ <code>VirtualMachine.Storage.ReserveMemory</code> ■ <code>VirtualMachine.Admin.ForceHost</code> ■ <code>VirtualMachine.Admin.HostSelectionPolicy</code> ■ <code>VMware.Endpoint.Openstack.Release</code> ■ <code>VirtualMachine.Admin.HostSelectionPolicy</code> ■ <code>VirtualMachine.Admin.ForceHost</code> ■ <code>VirtualMachine.Admin.EncryptPasswords</code>
001638-00	Initial 6.2 release.

Using Custom Properties

You can use custom properties to add new attributes or override default attributes.

A tenant administrator or business group manager can include custom properties when they create or edit blueprints. They can also include a build profile, which contains one or more custom properties. The properties are retrieved when a machine is provisioned by using the blueprint. A fabric administrator can also specify custom properties when they create or edit a reservation.

This chapter includes the following topics:

- [“Custom Property Uses,”](#) on page 9
- [“Specifying Custom Properties,”](#) on page 10
- [“Custom Property Types,”](#) on page 13
- [“Creating Custom Properties for Application Services,”](#) on page 13
- [“Build Profiles and Property Sets,”](#) on page 14

Custom Property Uses

You use custom properties to control aspects of the machines that users can provision.

Some properties are determined by standard settings that you must specify for all machines. For example, memory and disk size values are required for all blueprints. You can specify additional properties individually or in build profiles in blueprints and in reservations. Use custom properties to add values or override existing or default values for the following information.

- Machine operating system
- Virtualization platform
- Build settings such as disk size
- Integration with an external system

When you add a property to a blueprint or a build profile, you can mark it as a required property. When a property is specified as required, the user must provide a value for that property when they request a machine, such as in the following examples.

- Require information about multiple disks sharing the machine’s allocated storage.
- Require information about users or groups to be added to a local group on the machine.
- Require the host name of the machine.

The Windows guest agent records property values on the provisioned machine in the %SystemDrive%\VRMGuestAgent\site\workitem.xml file.

The Linux guest agent records property values on the provisioned machine in the /usr/share/gugent/site/workitem.xml file.

Custom Properties in Machine Provisioning, Blueprints, and Build Profiles

Custom properties are name-value pairs used to specify attributes of a machine or to override default specifications.

Different custom properties are used for different provisioning methods, types of machines, and machine options. Custom properties can be used as described in the following examples:

- Specify a particular type of guest OS
- Enable WIM-based provisioning, in which a Windows Imaging File Format (WIM) image of a reference machine is used to provision new machines
- Customize the behavior of Remote Desktop Protocol when connecting to a machine
- Register a virtual machine with a XenDesktop Desktop Delivery Controller (DDC) server
- Customize a virtual machine's system specifications, such as adding multiple hard disk drives
- Customize the guest OS for a machine, for instance, by including specified users in selected local groups
- Enable cleanup of a the Active Directory account of a machine after it is destroyed

Specifying the characteristics of the machines to be provisioned is generally done by adding properties to blueprints and build profiles. You can make custom properties available to multiple blueprints and all business groups by placing them in build profiles.

Any property specified in a blueprint overrides the same property specified in the incorporated build profile. This enables a blueprint to use most of the properties in a profile while differing from the profile in some limited way. For more information, see Order of Precedence for Custom Properties..

For example, a blueprint that incorporates a standard developer workstation profile might override the US English settings in the profile with UK English settings. On the other hand, if no appropriate profile is available all the needed properties can be specified in the blueprint itself. This arrangement ensures that the number and complexity of blueprints remain manageable.

At new machine request time, vRealize Automation has not yet allocated a reservation and the compute resource and endpoint are unknown as well. Therefore, only custom properties from a build profile, blueprint and business group are reconciled and presented when the machine is requested.

Specifying Custom Properties

You can use custom properties to control machine provisioning. A fabric administrator can use custom properties when they create or edit reservations. A tenant administrator or business group manager can specify custom properties when they create or edit blueprints.

Any property specified in a blueprint overrides the same property specified in the incorporated build profile. For example, a blueprint that contains a particular build profile might override the US English settings in the profile with UK English settings. This arrangement ensures that the number and complexity of blueprints remain manageable.

A custom property can optionally require that the user specify a property value when they create a machine request.

Custom property names are typically case-insensitive. Property values are typically case-sensitive. Custom properties cannot contain leading or trailing spaces.

For related information about using custom properties on blueprints and in build profiles, see *IaaS Configuration* in vRealize Automation documentation.

Order of Precedence for Custom Properties

When the same property exists in more than one source, a specific order is followed when applying properties to the machine.

You can add custom properties that apply to provisioned machines to the following elements:

- A reservation, to apply the custom properties to all machines provisioned from that reservation
- A business group, to apply the custom properties to all machines provisioned by business group members
- A global or local blueprint, to apply the custom properties to all machines provisioned from the blueprint
- Build profiles, which can be incorporated into any global or local blueprint, to apply the custom properties to all machines provisioned from the blueprint
- A machine request, if you are a business group manager, to apply the custom properties to the machine being provisioned
- The applicable approval policy, if any exist and if advanced approval support is enabled, to require approvers to provide the values to be applied to the machine being approved

The full order of precedence for custom properties is that any property value specified in a source later in the list overrides values for the same property specified in sources earlier in the list. The order is shown in the following list:

- 1 Build profile
- 2 Blueprint
- 3 Business group
- 4 Compute resource
- 5 Reservations
- 6 Endpoint
- 7 Runtime

For vApps, the order is similar, as shown in the following list:

- 1 Build profile, specified on a vApp component blueprint
- 2 vApp component blueprint
- 3 Build profile, specified on a vApp blueprint
- 4 vApp blueprint
- 5 Business group
- 6 Compute resources
- 7 Reservations
- 8 Endpoint
- 9 Runtime specified on a vApp
- 10 Runtime specified on a component machine

Any runtime property takes higher precedence and overrides a property from any source. A custom property is marked as runtime if the following conditions exist:

- The property is marked as Prompt User, which specifies that the user must supply a value for it when requesting a machine. This requires that the machine requestor customize individual characteristics of each machine, or gives them the option of doing so when a default value is provided for the required property.
- A business group manager is requesting a machine and the property appears in the custom properties list on the Properties tab of the Confirm Machine Request page.

Custom properties in reservations and business groups may be applied to many machines so they should be used carefully. Their use is typically limited to purposes related to their sources, such as resource management, line of business accounting, and so on. Specifying the characteristics of the machine to be provisioned is generally done by adding properties to blueprints and build profiles.

Each blueprint of any type can optionally incorporate one or more build profiles and thereby inherit the custom properties in those profiles. Build profiles are especially useful for applying common sets of properties for specific purposes to a wide range of blueprints. For example, your site might want to add a second disk to, customize Microsoft Remote Desktop Protocol behavior for, and enable Active Directory cleanup for a wide variety of machines. If a build profile with the necessary properties is created, it can be incorporated into all of your blueprints, local or global.

When creating and managing build profiles, a fabric administrator can load a number of predefined property sets to add several related properties all at once, instead of one by one.

Specifying Custom Properties for Multi-Machine Services

Tenant administrators and business group managers can specify custom properties that apply to all component machines in a multi-machine service in the multi-machine blueprint. When the same property exists in more than one source, vRealize Automation follows a specific order of precedence when applying properties to the machine.

Custom properties in a multi-machine blueprint override properties specified in component blueprints. Runtime properties on the component machine, which are specified at request time or by editing the machine after it is provisioned, override runtime properties specified at the multi-machine service level. This allows the multi-machine blueprint to apply consistent behavior across all of its component machines, while allowing a user to override the multi-machine service properties for each component type.

Custom properties on multi-machine service and component machines are processed in the following order.

- 1 Build profile specified on component blueprint
- 2 Component blueprint
- 3 Build profile specified on multi-machine blueprint
- 4 Multi-machine blueprint
- 5 Business group
- 6 Compute resource
- 7 Reservations
- 8 Endpoint
- 9 Request time specified on a multi-machine service
- 10 Request time specified on a component machine

A property value specified in a source that appears later in the list overrides values for the same property specified in sources earlier in the list. Custom properties specified in the multi-machine service are applied to all component machines in the service. If a property is designated as Prompt User on a component blueprint, the value specified at request time is applied to all machines of the same component type that are provisioned as part of that request.

Some properties, such as Hostname, must be unique to each machine. Do not specify the property at the component level. If the Hostname property is specified at the component level, it is ignored.

Custom Property Types

You can use vRealize Automation external and updated property types for cloned machines. You cannot use Internal and read-only property types for cloned machines.

The following vRealize Automation custom property types are available.

- Internal

The specified value is maintained in the database only. For example, the email address of the manager who approved a machine request is recorded in the `VirtualMachine.Admin.Approver` property but the property has no effect on the machine.

- Read-only

The specified value is implemented on the machine and cannot be changed. For example, `VirtualMachine.Admin.UUID` specifies the UUID of the machine, which cannot be changed.

- External

A machine's external properties are determined when the virtualization platform creates the machine or during the WinPE phase of the build process. To set these properties, their values must be provided to the proxy agent, which passes them on to the virtualization platform, or to the guest agent, which implements them in the WinPE phase.

The specified value is implemented on the machine but is never updated. For example, if the property `VirtualMachine.Admin.AddOwnerToAdmins` is set to true, the owner of the machine is added to its local administrators group. If the owner is later removed from this group, the property is not updated to false.

- Updated

The specified value is implemented on the machine and is updated through data collection. For example, if the compute resource of a machine is changed, a proxy agent updates the value of the machine's `VirtualMachine.Admin.Hostname` property.

Internal and read-only property types set attributes that the template determines.

You can use the vRealize Automation machine menu to change all reserved custom properties except the read-only properties `VirtualMachine.Admin.AgentID`, `VirtualMachine.Admin.UUID`, and `VirtualMachine.Admin.Name`.

Creating Custom Properties for Application Services

You can use custom properties that you define on a vRealize Automation blueprint when you define an application blueprint in Application Services.

Using vRealize Automation you can add custom properties to blueprints either individually or in a build profile. A build profile is a named collection of a group of custom properties. If you enable the Prompt User option for a custom property, the user is required to enter a value for the property when they use that blueprint to request machine provisioning.

Custom properties that have been made available to a specific business group are also available to members of that group as they create and edit blueprints.

You can use the custom properties on vRealize Automation blueprints to define applications in Application Services and publish them to the vRealize Automation catalog.

You can also use the vRealize Automation Property Dictionary when defining custom properties that use the Prompt User option. The following dictionary control types are supported for use with catalog items that are published from Application Services:

- DropDownList
- CheckBox

For related information about using custom properties in conjunction with Application Services, see *Using Application Services* in vRealize Automation documentation.

Build Profiles and Property Sets

vRealize Automation contains property sets that fabric administrators can use when they create build profiles.

Property sets are groups of related properties that are commonly used together in build profiles and machine blueprints. Instead of adding custom properties to a build profile or a machine blueprint individually, you can add an entire set to a build profile and provide the values.

Many commonly used property sets are included in vRealize Automation.

For example, the `WimImagingProperties` property set contains custom properties commonly used for WIM-based provisioning:

- `Image.ISO.Location`
- `Image.ISO.Name`
- `Image.Network.Password`
- `Image.Network.User`
- `Image.WIM.Index`
- `Image.WIM.Name`
- `Image.WIM.Path`

As another example, the `vApp` property set contains the following custom properties that can be used for vApp provisioning:

- `VirtualMachine.NetworkN.Name`
- `VCloud.Template.MakeIdenticalCopy`
- `VMware.SCSI.Type`
- `Sysprep.Identification.DomainAdmin`
- `Sysprep.Identification.DomainAdminPassword`
- `Sysprep.Identification.JoinDomain`

Fabric administrators can create their own property sets and add them to vRealize Automation.

Create a Build Profile


Fabric administrators can organize commonly used custom properties into build profiles so that tenant administrators and business group managers can include these custom property sets in blueprints.

Build profiles are available to tenant administrators and business group managers of all tenants. You can create your build profile by adding custom properties individually, by loading property sets that contain multiple custom properties, or by using a combination of the two methods.



Prerequisites

Log in to the vRealize Automation console as a **fabric administrator**.


Procedure

- 1 Select **Infrastructure > Blueprints > Build Profiles**.
- 2 Click the **Add** icon ().
- 3 Enter a name and, optionally, a description.
- 4 (Optional) Add custom properties individually to your build profile.
 - a Click **New Property**.
 - b Enter the custom property in the **Name** text box.
 - c Enter the value of the custom property in the **Value** text box.
 - d (Optional) Select the **Encrypted** check box to encrypt the custom property in the database.
 - e (Optional) Select the **Prompt user** check box to require the user to provide a value when they request a machine.

If you choose to prompt users for a value, any value you provide for the custom property is presented to them as the default. If you do not provide a default, users cannot continue with the machine request until they provide a value for the custom property.

- f Click the **Save** icon ().
- 5 (Optional) Select a property set to load into your build profile.
 - a Select a property set from the **Add from property set** drop-down menu.
 - b Click **Load**.
- 6 (Optional) Click the **Edit** icon () to configure a custom property loaded from a property set.
 - a Enter the value of the custom property in the **Value** text box.
 - b Select the **Encrypted** check box to encrypt the custom property in the database.
 - c Select the **Prompt user** check box to require the user to provide a value when they request a machine.

If you choose to prompt users for a value, any value you provide for the custom property is presented to them as the default. If you do not provide a default, users cannot continue with the machine request until they provide a value for the custom property.

 - d Click the **Save** icon ().
- 7 Click **OK**.

Tenant administrators and business group managers can now select your build profile and include it in their blueprints.

Create a Property Set

Fabric administrators can create their own groupings of related custom properties and add them to vRealize Automation for use in build profiles.

Property sets are available to fabric administrators of all tenants.

Create a Property Set XML File

A fabric administrator defines property sets in an XML file and uploads them to vRealize Automation.

If you edit a property set that is already in use in a build profile, vRealize Automation does not automatically update the build profile. A fabric administrator must reload the property set to the build profile.

Procedure

- 1 Create an XML file.
- 2 Insert the following version and encoding values into the schema declaration.
version="1.0" encoding="UTF-16".

- 3 Insert a <Doc> element.

```
<Doc>
</Doc>
```

- 4 Insert a <CustomProperties> element in the <Doc> element.

```
<Doc>
  <CustomProperties>
  </CustomProperties>
</Doc>
```

- 5 Define the attributes of the custom property to include in the property set.

```
<Doc>
  <CustomProperties>
    <Property Name="property_name" DefaultValue="property_value" Encrypted="true_or_false"
    PromptUser="true_or_false"/>
  </CustomProperties>
</Doc>
```

If you do not include the DefaultValue attribute, no default value is stored. If you do not include the Encrypted or PromptUser attributes, they default to false.

- 6 Repeat [Step 5](#) step for each property to include in the property set.
- 7 Save and close the file.

A fabric administrator can now upload your property set XML file to vRealize Automation.

Add a Property Set to vRealize Automation

After you create a property set XML file, a fabric administrator can upload the property set to vRealize Automation.

Prerequisites

- Log in to the vRealize Automation console as a **fabric administrator**.

- Create a property set XML file.

Procedure

- 1 Select **Infrastructure > Blueprints > Build Profiles**.
- 2 Click **Manage Property Sets**.
- 3 Click **Browse** to select the **Property set XML file name**.
- 4 Enter a name and, optionally, a description.
- 5 Click **OK**.

You can now include your property set in build profiles.

Custom Properties Grouped by Function

2

You can use custom properties to provide additional vRealize Automation controls.

These properties are also grouped by name and described in relevant guides in vRealize Automation product documentation. See [Chapter 3, “Custom Properties Grouped by Name,”](#) on page 49.

This chapter includes the following topics:

- [“Custom Properties for Openstack Endpoints,”](#) on page 20
- [“Custom Properties for Basic Workflow Blueprints,”](#) on page 20
- [“Custom Properties for Clone Blueprints,”](#) on page 22
- [“Custom Properties for FlexClone Blueprints,”](#) on page 25
- [“Custom Properties for Linked Clone Blueprints,”](#) on page 27
- [“Custom Properties for Linux Kickstart Blueprints,”](#) on page 29
- [“Custom Properties for SCCM Blueprints,”](#) on page 31
- [“Custom Properties for WIM Blueprints,”](#) on page 32
- [“Custom Properties for vApp Blueprints,”](#) on page 35
- [“Custom Properties for vRealize Automation Guest Agent,”](#) on page 37
- [“Custom Properties for Networking,”](#) on page 40
- [“Custom Properties for PXE Provisioning,”](#) on page 42
- [“Custom Properties for BMC BladeLogic Configuration Manager Integration,”](#) on page 45
- [“Custom Properties for HP Server Automation Integration,”](#) on page 46

Custom Properties for Openstack Endpoints

vRealize Automation includes custom properties you might want to use when you configure your Openstack endpoints in vRealize Automation.

Table 2-1. Custom Properties for Openstack Endpoints

Custom Property	Description
<code>VirtualMachine.Admin.ConnectAddress.Regex</code>	Used by a vRealize Automation administrator to define a regular expression to match an IP address for terminal connections, such as an RDP connection. If matched, the IP address is saved under the <code>VirtualMachine.Admin.ConnectAddress</code> custom property. Otherwise, the first available IP address is designated. For example, setting the property value to <code>10.10.0.</code> allows selection of an IP address from a <code>10.10.0.*</code> subnet that is assigned to the virtual machine. If the subnet has not been assigned, the property is ignored. This property is available for use with Openstack.
<code>VirtualMachine.NetworkN.AddressM</code>	Defines additional <i>M</i> IP address allocated for an Openstack instance for network <i>N</i> , excluding the IP address set specified by the <code>VirtualMachine.NetworkN.Address</code> property. Additional addresses are displayed on the Network tab in the Additional Addresses column. This property is used by Openstack machine state data collection. While this property is only data-collected by the OpenStack endpoint, it is not specific to OpenStack and can be used for lifecycle extensibility by other endpoint types.
<code>VMware.Endpoint.Openstack.IdentityProvider.Version</code>	For 6.2.4 and 6.2.5, specifies the version of Openstack Identity provider (Keystone) to use when authenticating an Openstack endpoint. Configure a value of 3 to authenticate with Keystone version 3 OpenStack Identity Provider. If you use any other value, or do not use this custom property, authentication defaults to Keystone version 2.
<code>VMware.Endpoint.Openstack.Release</code>	Specifies the OpenStack release, for example Havana or Icehouse, when creating an OpenStack endpoint. Required for 6.2.1, 6.2.2, and 6.2.3 OpenStack provisioning. Deprecated as of 6.2.4.

Custom Properties for Basic Workflow Blueprints

vRealize Automation includes custom properties that you can use to provide additional controls for basic workflow blueprints.

Table 2-2. Custom Properties for Basic Workflow Blueprints

Custom Property	Description
<code>VirtualMachine.CDRom.Attach</code>	Set to False to provision the machine without a CD-ROM device. The default is True.
<code>VirtualMachine.Admin.ThinProvision</code>	Determines whether thin provisioning is used on ESX compute resources using local or iSCSI storage. Set to True to use thin provisioning. Set to False to use standard provisioning. This property is for virtual provisioning.

Table 2-2. Custom Properties for Basic Workflow Blueprints (Continued)

Custom Property	Description
VirtualMachine.DiskN.StorageReservationPolicy	Specifies the storage reservation policy to use to find storage for disk <i>N</i> . Also assigns the named storage reservation policy to a volume. To use this property, substitute the volume number for <i>N</i> in the property name and specify a storage reservation policy name as the value. This property is equivalent to the storage reservation policy name specified on the blueprint. Disk numbering must be sequential. This property is valid for all Virtual and vCloud reservations. This property is not valid for Physical or Cloud reservations, other than vCloud reservations.
VirtualMachine.Storage.AllocationType	Stores collected groups to a single datastore. A distributed environment stores disks round-robin style.
VirtualMachine.Storage.Name	Identifies the storage path on which the machine resides. The default is the value specified in the reservation that was used to provision the machine.
VirtualMachine.Storage.ReserveMemory	Set to True to manage vSwap storage allocation to ensure availability and set allocation in the reservation. vSwap allocation is considered when you create or reconfigure a virtual machine. vSwap allocation checking is only available for vCenter Server endpoints. For multi-machine blueprints, add the property to the component blueprints. NOTE If you do not specify the <code>VirtualMachine.Storage.ReserveMemory</code> custom property when you create or provision the machine from vRealize Automation, swap space availability is not ensured. If you add the property for an already provisioned machine, and the allocated reservation is full, the storage allocated in the reservation might exceed the actual allocated storage.
VMware.Hardware.Version	Specifies the VM hardware version to be used for vSphere settings. Supported values are currently vmx-04, vmx-07, vmx-08, vmx-09 and vmx-10. This property is applicable for VM Create and VM Update workflows and is available only for basic workflow blueprints.

Custom Properties for Clone Blueprints

vRealize Automation includes custom properties that you can use to provide additional controls for clone blueprints.

Table 2-3. Custom Properties for Clone Blueprints

Custom Property	Description
<code>VirtualMachine.DiskN.StorageReservationPolicy</code>	Specifies the storage reservation policy to use to find storage for disk <i>N</i> . Also assigns the named storage reservation policy to a volume. To use this property, substitute the volume number for <i>N</i> in the property name and specify a storage reservation policy name as the value. This property is equivalent to the storage reservation policy name specified on the blueprint. Disk numbering must be sequential. This property is valid for all Virtual and vCloud reservations. This property is not valid for Physical or Cloud reservations, other than vCloud reservations.
<code>VirtualMachine.NetworkN.ProfileName</code>	Specifies the name of a network profile from which to assign a static IP address to network device <i>Nor</i> from which to obtain the range of static IP addresses that can be assigned to network device <i>N</i> of a cloned machine, where <i>N=0</i> for the first device, 1 for the second, and so on. If a network profile is specified in the network path in the reservation on which the machine is provisioned, a static IP address is assigned from that network profile. You can ensure that a static IP address is assigned from a specific profile by setting the value of this property to the name of a network profile. With WIM-based provisioning for virtual machines, you can use this property to specify a network profile and network interface or you can use the Network section of the Virtual Reservation page. You can also assign the network interface to a virtual network using the <code>VirtualMachine.NetworkN.Name</code> custom property.
<code>Linux.ExternalScript.Name</code>	Specifies the name of an optional customization script, for example <code>config.sh</code> , that the Linux guest agent runs after the operating system is installed. This property is available for Linux machines cloned from templates on which the Linux agent is installed. If you specify an external script, you must also define its location by using the <code>Linux.ExternalScript.LocationType</code> and <code>Linux.ExternalScript.Path</code> properties.
<code>Linux.ExternalScript.LocationType</code>	Specifies the location type of the customization script named in the <code>Linux.ExternalScript.Name</code> property. This can be either <code>local</code> or <code>nfs</code> . You must also specify the script location using the <code>Linux.ExternalScript.Path</code> property. If the location type is <code>nfs</code> , also use the <code>Linux.ExternalScript.Server</code> property.
<code>Linux.ExternalScript.Server</code>	Specifies the name of the NFS server, for example <code>lab-ad.lab.local</code> , on which the Linux external customization script named in <code>Linux.ExternalScript.Name</code> is located.
<code>Linux.ExternalScript.Path</code>	Specifies the local path to the Linux customization script or the export path to the Linux customization on the NFS server. The value must begin with a forward slash and not include the file name, for example <code>/scripts/linux/config.sh</code> .

If your administrators installed the guest agent to run scripts that accept custom properties and customize provisioned machines, you can use custom properties to further customize cloned machines that use the guest agent.

Table 2-4. Custom Properties for Customizing Cloned Machines with a Guest Agent

Custom Property	Description
<code>VirtualMachine.Admin.AddOwnerToAdmins</code>	Set to True (default) to add the machine's owner, as specified by the <code>VirtualMachine.Admin.Owner</code> property, to the local administrators group on the machine.
<code>VirtualMachine.Admin.AllowLogin</code>	Set to True (default) to add the machine owner to the local remote desktop users group, as specified by the <code>VirtualMachine.Admin.Owner</code> property.
<code>VirtualMachine.Admin.UseGuestAgent</code>	If the guest agent is installed as a service on a template for cloning, set to True on the machine blueprint to enable the guest agent service on machines cloned from that template. When the machine is started, the guest agent service is started. Set to False to disable the guest agent. If set to False, the enhanced clone workflow will not use the guest agent for guest operating system tasks, reducing its functionality to <code>VMwareCloneWorkflow</code> . If not specified or set to anything other than False, the enhanced clone workflow will send work items to the guest agent.
<code>VirtualMachine.DiskN.Active</code>	Set to True (default) to specify that the machine's disk <i>N</i> is active. Set to False to specify that the machine's disk <i>N</i> is not active.
<code>VirtualMachine.DiskN.Size</code>	Defines the size in GB of disk <i>N</i> . For example, to give a size of 150 GB to a disk <i>G</i> , define the custom property <code>VirtualMachine.Disk0.Size</code> and enter a value of 150. Disk numbering must be sequential. By default a machine has one disk referred to by <code>VirtualMachine.Disk0.Size</code> , where size is specified by the storage value on the blueprint from which the machine is provisioned. The storage value on the blueprint user interface overwrites the value in the <code>VirtualMachine.Disk0.Size</code> property. The <code>VirtualMachine.Disk0.Size</code> property is not available as a custom property because of its relationship with the storage option on the blueprint. More disks can be added by specifying <code>VirtualMachine.Disk1.Size</code> , <code>VirtualMachine.Disk2.Size</code> and so on. <code>VirtualMachine.Admin.TotalDiskUsage</code> always represents the total of the <code>.DiskN.Size</code> properties plus the <code>VMware.Memory.Reservation</code> size allocation.
<code>VirtualMachine.DiskN.Label</code>	Specifies the label for a machine's disk. The disk label maximum is 32 characters. Disk numbering must be sequential. When used in conjunction with a guest agent, specifies the label of a machine's disk <i>N</i> inside the guest operating system.
<code>VirtualMachine.DiskN.Letter</code>	Specifies the drive letter or mount point of a machine's disk <i>N</i> . The default is C. For example, to specify the letter D for Disk 1, define the custom property as <code>VirtualMachine.Disk1.Letter</code> and enter the value D. Disk numbering must be sequential. When used in conjunction with a guest agent, this value specifies the drive letter or mount point under which an additional disk <i>N</i> is mounted by the guest agent in the guest operating system.

Table 2-4. Custom Properties for Customizing Cloned Machines with a Guest Agent (Continued)

Custom Property	Description
<code>VirtualMachine.Admin.CustomizeGuestOSDelay</code>	Specifies the time to wait after customization is complete and before starting the guest operating system customization. The value must be in HH:MM:SS format. If the value is not set, the default value is one minute (00:01:00). If you choose not to include this custom property, provisioning can fail if the virtual machine reboots before guest agent work items are completed.
<code>VirtualMachine.Customize.WaitComplete</code>	Set to True to prevent the provisioning workflow from sending work items to the guest agent until all customizations have been completed.
<code>VirtualMachine.SoftwareN.Name</code>	Specifies the descriptive name of a software application or script to install or run during provisioning. This is an optional and information-only property. It serves no real function for the enhanced clone workflow or the guest agent but it is useful for a custom software selection in a user interface or for software usage reporting.
<code>VirtualMachine.SoftwareN.ScriptPath</code>	<p>Specifies the full path to an application's install script. The path must be a valid absolute path as seen by the guest operating system and must include the name of the script file name.</p> <p>You can pass custom property values as parameters to the script by inserting <code>{CustomPropertyName}</code> in the path string. For example, if you have a custom property named <code>ActivationKey</code> whose value is 1234, the script path is <code>D:\InstallApp.bat -key {ActivationKey}</code>. The guest agent runs the command <code>D:\InstallApp.bat -key 1234</code>. Your script file can then be programmed to accept and use this value.</p>
<code>VirtualMachine.SoftwareN.ISOName</code>	Specifies the path and file name of the ISO file relative to the data store root. The format is <code>/folder_name/subfolder_name/file_name.iso</code> . If a value is not specified, the ISO is not mounted.
<code>VirtualMachine.SoftwareN.ISOLocation</code>	Specifies the storage path that contains the ISO image file to be used by the application or script. Format the path as it appears on the host reservation, for example <code>netapp-1:it_nfs_1</code> . If a value is not specified, the ISO is not mounted.

Custom Properties for FlexClone Blueprints

vRealize Automation includes custom properties that you can use to provide additional controls for FlexClone blueprints.

Table 2-5. Custom Properties for FlexClone Blueprints

Custom Property	Description
<code>VirtualMachine.NetworkN.ProfileName</code>	<p>Specifies the name of a network profile from which to assign a static IP address to network device <code>Nor</code> from which to obtain the range of static IP addresses that can be assigned to network device <code>N</code> of a cloned machine, where <code>N=0</code> for the first device, 1 for the second, and so on.</p> <p>If a network profile is specified in the network path in the reservation on which the machine is provisioned, a static IP address is assigned from that network profile. You can ensure that a static IP address is assigned from a specific profile by setting the value of this property to the name of a network profile.</p> <p>With WIM-based provisioning for virtual machines, you can use this property to specify a network profile and network interface or you can use the Network section of the Virtual Reservation page. You can also assign the network interface to a virtual network using the <code>VirtualMachine.NetworkN.Name</code> custom property.</p>
<code>Linux.ExternalScript.Name</code>	<p>Specifies the name of an optional customization script, for example <code>config.sh</code>, that the Linux guest agent runs after the operating system is installed. This property is available for Linux machines cloned from templates on which the Linux agent is installed.</p> <p>If you specify an external script, you must also define its location by using the <code>Linux.ExternalScript.LocationType</code> and <code>Linux.ExternalScript.Path</code> properties.</p>
<code>Linux.ExternalScript.LocationType</code>	<p>Specifies the location type of the customization script named in the <code>Linux.ExternalScript.Name</code> property. This can be either <code>local</code> or <code>nfs</code>.</p> <p>You must also specify the script location using the <code>Linux.ExternalScript.Path</code> property. If the location type is <code>nfs</code>, also use the <code>Linux.ExternalScript.Server</code> property.</p>
<code>Linux.ExternalScript.Server</code>	<p>Specifies the name of the NFS server, for example <code>lab-ad.lab.local</code>, on which the Linux external customization script named in <code>Linux.ExternalScript.Name</code> is located.</p>
<code>Linux.ExternalScript.Path</code>	<p>Specifies the local path to the Linux customization script or the export path to the Linux customization on the NFS server. The value must begin with a forward slash and not include the file name, for example <code>/scripts/linux/config.sh</code>.</p>

If you installed the guest agent to customize cloned machines, the Custom Properties for Customizing FlexClone Machines with a Guest Agent table describes the most commonly used custom properties for your situation.

Table 2-6. Custom Properties for Customizing FlexClone Machines with a Guest Agent

Custom Property	Description
<code>VirtualMachine.Admin.UseGuestAgent</code>	If the guest agent is installed as a service on a template for cloning, set to True on the machine blueprint to enable the guest agent service on machines cloned from that template. When the machine is started, the guest agent service is started. Set to False to disable the guest agent. If set to False, the enhanced clone workflow will not use the guest agent for guest operating system tasks, reducing its functionality to <code>VMwareCloneWorkflow</code> . If not specified or set to anything other than False, the enhanced clone workflow will send work items to the guest agent.
<code>VirtualMachine.DiskN.Size</code>	Defines the size in GB of disk <i>N</i> . For example, to give a size of 150 GB to a disk G, define the custom property <code>VirtualMachine.Disk0.Size</code> and enter a value of 150. Disk numbering must be sequential. By default a machine has one disk referred to by <code>VirtualMachine.Disk0.Size</code> , where size is specified by the storage value on the blueprint from which the machine is provisioned. The storage value on the blueprint user interface overwrites the value in the <code>VirtualMachine.Disk0.Size</code> property. The <code>VirtualMachine.Disk0.Size</code> property is not available as a custom property because of its relationship with the storage option on the blueprint. More disks can be added by specifying <code>VirtualMachine.Disk1.Size</code> , <code>VirtualMachine.Disk2.Size</code> and so on. <code>VirtualMachine.Admin.TotalDiskUsage</code> always represents the total of the <code>.DiskN.Size</code> properties plus the <code>VMware.Memory.Reservation</code> size allocation.
<code>VirtualMachine.DiskN.Label</code>	Specifies the label for a machine's disk. The disk label maximum is 32 characters. Disk numbering must be sequential. When used in conjunction with a guest agent, specifies the label of a machine's disk <i>N</i> inside the guest operating system.
<code>VirtualMachine.DiskN.Letter</code>	Specifies the drive letter or mount point of a machine's disk <i>N</i> . The default is C. For example, to specify the letter D for Disk 1, define the custom property as <code>VirtualMachine.Disk1.Letter</code> and enter the value D. Disk numbering must be sequential. When used in conjunction with a guest agent, this value specifies the drive letter or mount point under which an additional disk <i>N</i> is mounted by the guest agent in the guest operating system.
<code>VirtualMachine.Admin.CustomizeGuestOSDelay</code>	Specifies the time to wait after customization is complete and before starting the guest operating system customization. The value must be in HH:MM:SS format. If the value is not set, the default value is one minute (00:01:00). If you choose not to include this custom property, provisioning can fail if the virtual machine reboots before guest agent work items are completed.

Table 2-6. Custom Properties for Customizing FlexClone Machines with a Guest Agent (Continued)

Custom Property	Description
<code>VirtualMachine.Customize.WaitComplete</code>	Set to True to prevent the provisioning workflow from sending work items to the guest agent until all customizations have been completed.
<code>VirtualMachine.SoftwareN.ScriptPath</code>	Specifies the full path to an application's install script. The path must be a valid absolute path as seen by the guest operating system and must include the name of the script file name. You can pass custom property values as parameters to the script by inserting <code>{CustomPropertyName}</code> in the path string. For example, if you have a custom property named <code>ActivationKey</code> whose value is <code>1234</code> , the script path is <code>D:\InstallApp.bat -key {ActivationKey}</code> . The guest agent runs the command <code>D:\InstallApp.bat -key 1234</code> . Your script file can then be programmed to accept and use this value.

Custom Properties for Linked Clone Blueprints

vRealize Automation includes custom properties that you can use to provide additional controls for linked clone blueprints.

Certain vRealize Automation custom properties are required to use with linked clone blueprints.

Table 2-7. Custom Properties for Linked Clone Blueprints

Custom Property	Description
<code>VirtualMachine.DiskN.Storage</code>	Specifies the datastore on which to place the machine disk <i>N</i> , for example <code>DATASTORE01</code> . This property is also used to add a single datastore to a linked clone blueprint. <i>N</i> is the index (starting at 0) of the volume to assign. Enter the name of the datastore to assign to the volume. This is the datastore name as it appears in the Storage Path on the Edit Compute Resource page. Disk numbering must be sequential.
<code>VirtualMachine.DiskN.StorageReservationPolicy</code>	Specifies the storage reservation policy to use to find storage for disk <i>N</i> . Also assigns the named storage reservation policy to a volume. To use this property, substitute the volume number for <i>N</i> in the property name and specify a storage reservation policy name as the value. This property is equivalent to the storage reservation policy name specified on the blueprint. Disk numbering must be sequential. This property is valid for all Virtual and vCloud reservations. This property is not valid for Physical or Cloud reservations, other than vCloud reservations.

Table 2-7. Custom Properties for Linked Clone Blueprints (Continued)

Custom Property	Description
VirtualMachine.DiskN.Size	<p>Defines the size in GB of disk <i>N</i>. For example, to give a size of 150 GB to a disk <i>G</i>, define the custom property <code>VirtualMachine.Disk0.Size</code> and enter a value of 150. Disk numbering must be sequential. By default a machine has one disk referred to by <code>VirtualMachine.Disk0.Size</code>, where size is specified by the storage value on the blueprint from which the machine is provisioned. The storage value on the blueprint user interface overwrites the value in the <code>VirtualMachine.Disk0.Size</code> property. The <code>VirtualMachine.Disk0.Size</code> property is not available as a custom property because of its relationship with the storage option on the blueprint. More disks can be added by specifying <code>VirtualMachine.Disk1.Size</code>, <code>VirtualMachine.Disk2.Size</code> and so on. <code>VirtualMachine.Admin.TotalDiskUsage</code> always represents the total of the <code>.DiskN.Size</code> properties plus the <code>VMware.Memory.Reservation</code> size allocation.</p>
VirtualMachine.DiskN.Label	<p>Specifies the label for a machine's disk. The disk label maximum is 32 characters. Disk numbering must be sequential. When used in conjunction with a guest agent, specifies the label of a machine's disk <i>N</i> inside the guest operating system.</p>
VirtualMachine.DiskN.Letter	<p>Specifies the drive letter or mount point of a machine's disk <i>N</i>. The default is <i>C</i>. For example, to specify the letter <i>D</i> for Disk 1, define the custom property as <code>VirtualMachine.Disk1.Letter</code> and enter the value <i>D</i>. Disk numbering must be sequential. When used in conjunction with a guest agent, this value specifies the drive letter or mount point under which an additional disk <i>N</i> is mounted by the guest agent in the guest operating system.</p>
MaximumProvisionedMachines	<p>Specifies the maximum number of linked clones for one machine snapshot. The default is unlimited.</p>
Linux.ExternalScript.Name	<p>Specifies the name of an optional customization script, for example <code>config.sh</code>, that the Linux guest agent runs after the operating system is installed. This property is available for Linux machines cloned from templates on which the Linux agent is installed.</p> <p>If you specify an external script, you must also define its location by using the <code>Linux.ExternalScript.LocationType</code> and <code>Linux.ExternalScript.Path</code> properties.</p>
Linux.ExternalScript.LocationType	<p>Specifies the location type of the customization script named in the <code>Linux.ExternalScript.Name</code> property. This can be either <code>local</code> or <code>nfs</code>.</p> <p>You must also specify the script location using the <code>Linux.ExternalScript.Path</code> property. If the location type is <code>nfs</code>, also use the <code>Linux.ExternalScript.Server</code> property.</p>

Table 2-7. Custom Properties for Linked Clone Blueprints (Continued)

Custom Property	Description
<code>Linux.ExternalScript.Server</code>	Specifies the name of the NFS server, for example <code>lab-ad.lab.local</code> , on which the Linux external customization script named in <code>Linux.ExternalScript.Name</code> is located.
<code>Linux.ExternalScript.Path</code>	Specifies the local path to the Linux customization script or the export path to the Linux customization on the NFS server. The value must begin with a forward slash and not include the file name, for example <code>/scripts/linux/config.sh</code> .

If you installed the guest agent to customize cloned machines, you use some custom properties more often than others.

Table 2-8. Custom Properties for Customizing Cloned Machines with a Guest Agent

Custom Property	Description
<code>VirtualMachine.Admin.UseGuestAgent</code>	If the guest agent is installed as a service on a template for cloning, set to <code>True</code> on the machine blueprint to enable the guest agent service on machines cloned from that template. When the machine is started, the guest agent service is started. Set to <code>False</code> to disable the guest agent. If set to <code>False</code> , the enhanced clone workflow will not use the guest agent for guest operating system tasks, reducing its functionality to <code>VMwareCloneWorkflow</code> . If not specified or set to anything other than <code>False</code> , the enhanced clone workflow will send work items to the guest agent.
<code>VirtualMachine.Admin.CustomizeGuestOSDelay</code>	Specifies the time to wait after customization is complete and before starting the guest operating system customization. The value must be in <code>HH:MM:SS</code> format. If the value is not set, the default value is one minute (<code>00:01:00</code>). If you choose not to include this custom property, provisioning can fail if the virtual machine reboots before guest agent work items are completed.
<code>VirtualMachine.Customize.WaitComplete</code>	Set to <code>True</code> to prevent the provisioning workflow from sending work items to the guest agent until all customizations have been completed.
<code>VirtualMachine.SoftwareN.ScriptPath</code>	Specifies the full path to an application's install script. The path must be a valid absolute path as seen by the guest operating system and must include the name of the script file name. You can pass custom property values as parameters to the script by inserting <code>{CustomPropertyName}</code> in the path string. For example, if you have a custom property named <code>ActivationKey</code> whose value is <code>1234</code> , the script path is <code>D:\InstallApp.bat -key {ActivationKey}</code> . The guest agent runs the command <code>D:\InstallApp.bat -key 1234</code> . Your script file can then be programmed to accept and use this value.

Custom Properties for Linux Kickstart Blueprints

vRealize Automation includes custom properties that you can use to provide additional controls for Linux Kickstart blueprints.

Certain vRealize Automation custom properties are required to use with Linux Kickstart blueprints.

Table 2-9. Required Custom Properties for Linux Kickstart Blueprints

Custom Property	Description
<code>VMware.VirtualCenter.OperatingSystem</code>	Specifies the vCenter Server guest operating system version (<code>VirtualMachineGuestOsIdentifier</code>) with which vCenter Server creates the machine. This operating system version must match the operating system version to be installed on the provisioned machine. Administrators can create property groups using one of several property sets, for example, <code>VMware[OS_Version]Properties</code> , that are predefined to include the correct <code>VMware.VirtualCenter.OperatingSystem</code> values. This property is for virtual provisioning. For related information, see the enumeration type <code>VirtualMachineGuestOsIdentifier</code> in vSphere API/SDK Documentation. For a list of currently accepted values, see the VMware vCenter Server™ documentation.
<code>Image.ISO.Location</code>	Values for this property are case sensitive. Specifies the location of the ISO image from which to boot, for example <code>http://192.168.2.100/site2/winpe.iso</code> . The format of this value depends on your platform. For details, see documentation provided for your platform. This property is required for WIM-based provisioning, Linux Kickstart and autoYaST provisioning, and SCCM-based provisioning.
<code>Image.ISO.Name</code>	Values for this property are case sensitive. Specifies the name of the ISO image from which to boot, for example <code>/ISO/Microsoft/WinPE.iso</code> . The format of this value depends on your platform. For details, see documentation provided for your platform. This property is required for WIM-based provisioning, Linux Kickstart and autoYaST provisioning, and SCCM-based provisioning.
<code>Image.ISO.UserName</code>	Specifies the user name to access the CIFS share in the format <code>username@domain</code> . For Dell iDRAC integrations where the image is located on a CIFS share that requires authentication to access.
<code>Image.ISO.Password</code>	Specifies the password associated with the <code>Image.ISO.UserName</code> property. For Dell iDRAC integrations where the image is located on a CIFS share that requires authentication to access.

Optional custom properties are available.

Table 2-10. Optional Custom Properties for Linux Kickstart Blueprints

Custom Property	Description
<code>VirtualMachine.Admin.ThinProvision</code>	Determines whether thin provisioning is used on ESX compute resources using local or iSCSI storage. Set to True to use thin provisioning. Set to False to use standard provisioning. This property is for virtual provisioning.
<code>Machine.SSH</code>	Set to True to enable the Connect Using SSH option, on the vRealize Automation Items page, for Linux machines provisioned from this blueprint. If set to True and the Connect using RDP or SSH machine operation is enabled in the blueprint, all Linux machines that are provisioned from the blueprint display the Connect Using SSH option to entitled users.

Custom Properties for SCCM Blueprints

vRealize Automation includes custom properties that you can use to provide additional controls for SCCM blueprints.

Certain custom properties are required to use with SCCM blueprints.

Table 2-11. Required Custom Properties for SCCM Blueprints

Custom Property	Description
Image.ISO.Location	Values for this property are case sensitive. Specifies the location of the ISO image from which to boot, for example <code>http://192.168.2.100/site2/winpe.iso</code> . The format of this value depends on your platform. For details, see documentation provided for your platform. This property is required for WIM-based provisioning, Linux Kickstart and autoYaST provisioning, and SCCM-based provisioning.
Image.ISO.Name	Values for this property are case sensitive. Specifies the name of the ISO image from which to boot, for example <code>/ISO/Microsoft/WinPE.iso</code> . The format of this value depends on your platform. For details, see documentation provided for your platform. This property is required for WIM-based provisioning, Linux Kickstart and autoYaST provisioning, and SCCM-based provisioning.
Image.ISO.UserName	Specifies the user name to access the CIFS share in the format <code>username@domain</code> . For Dell iDRAC integrations where the image is located on a CIFS share that requires authentication to access.
Image.ISO.Password	Specifies the password associated with the <code>Image.ISO.UserName</code> property. For Dell iDRAC integrations where the image is located on a CIFS share that requires authentication to access.
SCCM.Collection.Name	Specifies the name of the SCCM collection that contains the operating system deployment task sequence.
SCCM.Server.Name	Specifies the fully qualified domain name of the SCCM server on which the collection resides, for example <code>lab-sccm.lab.local</code> .
SCCM.Server.SiteCode	Specifies the site code of the SCCM server.
SCCM.Server.UserName	Specifies a user name with administrator-level access to the SCCM server.
SCCM.Server.Password	Specifies the password associated with the <code>SCCM.Server.UserName</code> property.

Certain custom properties are used most often with SCCM blueprints.

Table 2-12. Common Custom Properties for SCCM Blueprints

Custom Property	Description
SCCM.CustomVariable.Name	Specifies the value of a custom variable, where <i>Name</i> is the name of any custom variable to be made available to the SCCM task sequence after the provisioned machine is registered with the SCCM collection. The value is determined by your choice of custom variable. If your integration requires it, you can use SCCM.RemoveCustomVariablePrefix to remove the SCCM.CustomVariable. prefix from your custom variable.
SCCM.RemoveCustomVariablePrefix	Set to <i>true</i> to remove the prefix SCCM.CustomVariable. from SCCM custom variables you created by using the custom property SCCM.CustomVariable.Name.

Custom Properties for WIM Blueprints

vRealize Automation includes custom properties that provide additional controls for WIM blueprints.

Certain vRealize Automation custom properties are required for WIM blueprints.

Table 2-13. Required Custom Properties for WIM Blueprints

Custom Property	Description
Image.ISO.Location	Values for this property are case sensitive. Specifies the location of the ISO image from which to boot, for example http://192.168.2.100/site2/winpe.iso . The format of this value depends on your platform. For details, see documentation provided for your platform. This property is required for WIM-based provisioning, Linux Kickstart and autoYaST provisioning, and SCCM-based provisioning.
Image.ISO.Name	Values for this property are case sensitive. Specifies the name of the ISO image from which to boot, for example /ISO/Microsoft/WinPE.iso . The format of this value depends on your platform. For details, see documentation provided for your platform. This property is required for WIM-based provisioning, Linux Kickstart and autoYaST provisioning, and SCCM-based provisioning.
Image.ISO.UserName	Specifies the user name to access the CIFS share in the format <i>username@domain</i> . For Dell iDRAC integrations where the image is located on a CIFS share that requires authentication to access.
Image.ISO.Password	Specifies the password associated with the <code>Image.ISO.UserName</code> property. For Dell iDRAC integrations where the image is located on a CIFS share that requires authentication to access.
Image.Network.Letter	Specifies the drive letter to which the WIM image path is mapped on the provisioned machine. The default value is K.
Image.WIM.Path	Specifies the UNC path to the WIM file from which an image is extracted during WIM-based provisioning. The path format is <code>\\server\share\$</code> format, for example <code>\\lab-ads\dfs\$</code> .
Image.WIM.Name	Specifies the name of the WIM file, for example <code>win2k8.wim</code> , as located by the <code>Image.WIM.Path</code> property.
Image.WIM.Index	Specifies the index used to extract the correct image from the WIM file.

Table 2-13. Required Custom Properties for WIM Blueprints (Continued)

Custom Property	Description
Image.Network.User	Specifies the user name with which to map the WIM image path (<code>Image.WIM.Path</code>) to a network drive on the provisioned machine. This is typically a domain account with access to the network share.
Image.Network.Password	Specifies the password associated with the <code>Image.Network.User</code> property.
VMware.VirtualCenter.OperatingSystem	Specifies the vCenter Server guest operating system version (<code>VirtualMachineGuestOsIdentifier</code>) with which vCenter Server creates the machine. This operating system version must match the operating system version to be installed on the provisioned machine. Administrators can create property groups using one of several property sets, for example, <code>VMware[OS_Version]Properties</code> , that are predefined to include the correct <code>VMware.VirtualCenter.OperatingSystem</code> values. This property is for virtual provisioning. For related information, see the enumeration type <code>VirtualMachineGuestOsIdentifier</code> in vSphere API/SDK Documentation. For a list of currently accepted values, see the VMware vCenter Server™ documentation.

Optional custom properties are also available for WIM blueprints.

Table 2-14. Common Custom Properties for WIM Blueprints

Custom Property	Description
SysPrep. <i>Section.Key</i> ■ SysPrep.GuiUnattended.AdminPassword ■ SysPrep.GuiUnattended.EncryptedAdminPassword ■ SysPrep.GuiUnattended.TimeZone	<p>Specifies information to be added to the SysPrep answer file on machines during the WinPE stage of provisioning. Information that already exists in the SysPrep answer file is overwritten by these custom properties. <i>Section</i> represents the name of the section of the SysPrep answer file, for example GuiUnattended or UserData. <i>Key</i> represents a key name in the section. For example, to set the time zone of a provisioned machine to West Pacific Standard Time, define the custom property GuiUnattended.UserData.TimeZone and set the value to 275.</p> <p>For a full list of sections, keys, and accepted values, see the System Preparation Utility for Windows documentation.</p> <p>The following <i>Section.Key</i> combinations can be specified for WIM-based provisioning:</p> <ul style="list-style-type: none"> ■ GuiUnattended <ul style="list-style-type: none"> ■ AdminPassword ■ EncryptedAdminPassword ■ TimeZone ■ UserData <ul style="list-style-type: none"> ■ ProductKey ■ FullName ■ ComputerName ■ OrgName ■ Identification <ul style="list-style-type: none"> ■ DomainAdmin ■ DomainAdminPassword ■ JoinDomain ■ JoinWorkgroup
Sysprep.Identification.DomainAdmin	Specifies a user name with administrator-level access to the target domain in Active Directory. Do not include the user domain in the credentials that you send to vCloud Director.
Sysprep.Identification.DomainAdminPassword	Specifies the password to associate with the Sysprep.Identification.DomainAdmin property.
Sysprep.Identification.JoinDomain	Specifies the name of the domain to join in Active Directory.
Sysprep.Identification.JoinWorkgroup	Specifies the name of the workgroup to join if not using a domain.
SysPrep.UserData.ComputerName	Specifies a machine name, for example lab-client005.
SysPrep.UserData.FullName	Specifies the full name of a user.
SysPrep.UserData.OrgName	Specifies the organization name of the user.
SysPrep.UserData.ProductKey	Specifies the Windows product key.
VirtualMachine.Admin.ThinProvision	Determines whether thin provisioning is used on ESX compute resources using local or iSCSI storage. Set to True to use thin provisioning. Set to False to use standard provisioning. This property is for virtual provisioning.

Custom Properties for vApp Blueprints

You can add certain vRealize Automation custom properties to a vApp or vApp component blueprint or property group to allow a vApp machine to join an Active Directory domain.

The Active Directory domain is supported for Windows machines. If the vApp machine contains Linux and Windows component machines, use a property group to assign custom properties to the vApp component blueprint associated with the Windows machines.

Table 2-15. Custom Properties for vApp Blueprints

Custom Property	Description
<code>Sysprep.Identification.DomainAdmin</code>	Specifies a user name with administrator-level access to the target domain in Active Directory. Do not include the user domain in the credentials that you send to vCloud Director.
<code>Sysprep.Identification.DomainAdminPassword</code>	Specifies the password to associate with the <code>Sysprep.Identification.DomainAdmin</code> property.
<code>Sysprep.Identification.JoinDomain</code>	Specifies the name of the domain to join in Active Directory.
<code>VirtualMachine.DiskN.IsFixed</code>	Disables the editing of a specific disk when reconfiguring a machine. Set to True to disable display of the edit capacity option for a specific volume. The True value is case-sensitive. The <i>N</i> value is the 0-based index of the disk. Alternatively, you can set the <code>VirtualMachine.DiskN.IsFixed</code> custom property to True in the <code>VirtualMachineProperties</code> table in the database or use the Repository API to specify a URI value such as <code>.../Repository/Data/ManagementModelEntities.svc/VirtualMachines(guid'60D93A8A-F541-4CE0-A6C6-78973AC0F1D2')/VirtualMachineProperties</code> .
<code>VirtualMachine.DiskN.StorageReservationPolicy</code>	Specifies the storage reservation policy to use to find storage for disk <i>N</i> . Also assigns the named storage reservation policy to a volume. To use this property, substitute the volume number for <i>N</i> in the property name and specify a storage reservation policy name as the value. This property is equivalent to the storage reservation policy name specified on the blueprint. Disk numbering must be sequential. This property is valid for all Virtual and vCloud reservations. This property is not valid for Physical or Cloud reservations, other than vCloud reservations.

Table 2-15. Custom Properties for vApp Blueprints (Continued)

Custom Property	Description
<code>VirtualMachine.NetworkN.Name</code>	<p>Specifies the name of the network to connect to, for example the network device <i>N</i> to which a machine is attached.</p> <p>By default, a network is assigned from the network paths available on the reservation on which the machine is provisioned.</p> <p>You can ensure that a network device is connected to a specific network by setting the value of this property to the name of a network on an available reservation.</p> <p><code>VirtualMachine.NetworkN</code> custom properties are designed to be specific to individual blueprints and machines. When a machine is requested, network and IP address allocation is performed before the machine is assigned to a reservation. Because blueprints are not guaranteed to be allocated to a specific reservation, do not use this property on a reservation.</p> <p>You can add this property to a vApp (vCloud) blueprint or a vApp (vCloud) Component blueprint.</p> <p>For related information, see “Custom Properties for Networking,” on page 40.</p>
<code>VCloud.Lease.Sync.TimeBufferMins</code>	<p>Specifies a threshold integer value for a compute resource such that lease synchronization between vCloud Director and vRealize Automation will only occur for vCloud Director or vCloud Air provisioned machines that are set to expire in vCloud Director or vCloud Air in that time period. If a conflict is found, the lease value is synchronized to match the lease length defined in vRealize Automation. The default <code>VCloud.Lease.Sync.TimeBufferMins</code> value is 720 minutes, which is 12 hours. If <code>VCloud.Lease.Sync.TimeBufferMins</code> is not present, the default value is used. For example, if the default values are used, vRealize Automation will run the lease synchronization check workflow every 45 minutes, which is the workflow default, and only the leases of machines that are set to expire within 12 hours are changed to match the lease length defined in vRealize Automation.</p>

Table 2-15. Custom Properties for vApp Blueprints (Continued)

Custom Property	Description
<code>VCloud.Template.MakeIdenticalCopy</code>	<p>Set to True to clone an identical copy of the vApp template in vCloud Director and provision the results in vRealize Automation. This ignores all settings specified in the blueprints except the name of the vApp and its virtual machines. The storage path specified in the vApp template during cloning is used, even if a different storage path is specified in a vApp component blueprint or when requesting a vApp machine.</p> <p>Set to False to clone a copy of the vApp template with settings specified by the vApp and vApp component blueprints.</p>
<code>VMware.SCSI.Type</code>	<p>For vApps and vSphere, specifies the SCSI machine type using one of the following case-sensitive values:</p> <ul style="list-style-type: none"> ■ <code>buslogic</code> Use BusLogic emulation for the virtual disk. ■ <code>lsilogic</code> Use LSILogic emulation for the virtual disk (default). ■ <code>lsilogicsas</code> Use LSILogic SAS 1068 emulation for the virtual disk. ■ <code>pvscsi</code> Use para-virtualization emulation for the virtual disk. ■ <code>none</code> Use if a SCSI controller does not exist for this machine.

Custom Properties for vRealize Automation Guest Agent

If you have installed the vRealize Automation guest agent in your templates for cloning or in your WinPE, you can use custom properties to run custom scripts within the guest operating system of a provisioned machine after the machine is fully deployed.

Table 2-16. Custom Properties for Customizing Provisioned Machines with a Guest Agent

Custom Property	Description
<code>VirtualMachine.Admin.AddOwnerToAdmins</code>	Set to True (default) to add the machine's owner, as specified by the <code>VirtualMachine.Admin.Owner</code> property, to the local administrators group on the machine.
<code>VirtualMachine.Admin.AllowLogin</code>	Set to True (default) to add the machine owner to the local remote desktop users group, as specified by the <code>VirtualMachine.Admin.Owner</code> property.
<code>VirtualMachine.Admin.UseGuestAgent</code>	If the guest agent is installed as a service on a template for cloning, set to True on the machine blueprint to enable the guest agent service on machines cloned from that template. When the machine is started, the guest agent service is started. Set to False to disable the guest agent. If set to False, the enhanced clone workflow will not use the guest agent for guest operating system tasks, reducing its functionality to <code>VMwareCloneWorkflow</code> . If not specified or set to anything other than False, the enhanced clone workflow will send work items to the guest agent.
<code>VirtualMachine.DiskN.Active</code>	Set to True (default) to specify that the machine's disk <i>N</i> is active. Set to False to specify that the machine's disk <i>N</i> is not active.

Table 2-16. Custom Properties for Customizing Provisioned Machines with a Guest Agent (Continued)

Custom Property	Description
<code>VirtualMachine.DiskN.Size</code>	<p>Defines the size in GB of disk <i>N</i>. For example, to give a size of 150 GB to a disk <i>G</i>, define the custom property <code>VirtualMachine.Disk0.Size</code> and enter a value of 150. Disk numbering must be sequential. By default a machine has one disk referred to by <code>VirtualMachine.Disk0.Size</code>, where size is specified by the storage value on the blueprint from which the machine is provisioned. The storage value on the blueprint user interface overwrites the value in the <code>VirtualMachine.Disk0.Size</code> property. The <code>VirtualMachine.Disk0.Size</code> property is not available as a custom property because of its relationship with the storage option on the blueprint. More disks can be added by specifying <code>VirtualMachine.Disk1.Size</code>, <code>VirtualMachine.Disk2.Size</code> and so on. <code>VirtualMachine.Admin.TotalDiskUsage</code> always represents the total of the <code>.DiskN.Size</code> properties plus the <code>VMware.Memory.Reservation</code> size allocation.</p>
<code>VirtualMachine.DiskN.Label</code>	<p>Specifies the label for a machine's disk. The disk label maximum is 32 characters. Disk numbering must be sequential. When used in conjunction with a guest agent, specifies the label of a machine's disk <i>N</i> inside the guest operating system.</p>
<code>VirtualMachine.DiskN.Letter</code>	<p>Specifies the drive letter or mount point of a machine's disk <i>N</i>. The default is <i>C</i>. For example, to specify the letter <i>D</i> for Disk 1, define the custom property as <code>VirtualMachine.Disk1.Letter</code> and enter the value <i>D</i>. Disk numbering must be sequential. When used in conjunction with a guest agent, this value specifies the drive letter or mount point under which an additional disk <i>N</i> is mounted by the guest agent in the guest operating system.</p>
<code>VirtualMachine.Admin.CustomizeGuestOSDelay</code>	<p>Specifies the time to wait after customization is complete and before starting the guest operating system customization. The value must be in <i>HH:MM:SS</i> format. If the value is not set, the default value is one minute (00:01:00). If you choose not to include this custom property, provisioning can fail if the virtual machine reboots before guest agent work items are completed.</p>
<code>VirtualMachine.Customize.WaitComplete</code>	<p>Set to <i>True</i> to prevent the provisioning workflow from sending work items to the guest agent until all customizations have been completed.</p>
<code>VirtualMachine.SoftwareN.Name</code>	<p>Specifies the descriptive name of a software application or script to install or run during provisioning. This is an optional and information-only property. It serves no real function for the enhanced clone workflow or the guest agent but it is useful for a custom software selection in a user interface or for software usage reporting.</p>

Table 2-16. Custom Properties for Customizing Provisioned Machines with a Guest Agent (Continued)

Custom Property	Description
VirtualMachine.SoftwareN.ScriptPath	<p>Specifies the full path to an application's install script. The path must be a valid absolute path as seen by the guest operating system and must include the name of the script file name.</p> <p>You can pass custom property values as parameters to the script by inserting <code>{CustomPropertyName}</code> in the path string. For example, if you have a custom property named <code>ActivationKey</code> whose value is <code>1234</code>, the script path is <code>D:\InstallApp.bat -key {ActivationKey}</code>. The guest agent runs the command <code>D:\InstallApp.bat -key 1234</code>. Your script file can then be programmed to accept and use this value.</p>
VirtualMachine.ScriptPath.Decrypt	<p>Allows vRealize Automation to obtain an encrypted string that is passed as a properly formatted <code>VirtualMachine.SoftwareN.ScriptPath</code> custom property statement to the gagent command line.</p> <p>You can provide an encrypted string, such as your password, as a custom property in a command line argument. This allows you to store encrypted information that the guest agent can decrypt and understand as a valid command line argument. For example, the <code>VirtualMachine.Software0.ScriptPath = c:\dosomething.bat password</code> custom property string is not secure as it contains an actual password.</p> <p>To encrypt the password, you can create a vRealize Automation custom property, for example <code>MyPassword = password</code>, and enable encryption by selecting the available checkbox. The guest agent decrypts the <code>[MyPassword]</code> entry to the value in the custom property <code>MyPassword</code> and executes the script as <code>c:\dosomething.bat password</code>.</p> <ul style="list-style-type: none"> ■ Create custom property <code>MyPassword = password</code> where <code>password</code> is the value of your actual password. Enable encryption by selecting the available checkbox. ■ Set custom property <code>VirtualMachine.ScriptPath.Decrypt</code> as <code>VirtualMachine.ScriptPath.Decrypt = true</code>. ■ Set custom property <code>VirtualMachine.Software0.ScriptPath</code> as <code>VirtualMachine.Software0.ScriptPath = c:\dosomething.bat [MyPassword]</code>. <p>If you set <code>VirtualMachine.ScriptPath.Decrypt</code> to <code>false</code>, or don't create the <code>VirtualMachine.ScriptPath.Decrypt</code> custom property, then the string inside the square brackets ([and]) is not decrypted.</p>
VirtualMachine.SoftwareN.ISOName	<p>Specifies the path and file name of the ISO file relative to the data store root. The format is <code>/folder_name/subfolder_name/file_name.iso</code>. If a value is not specified, the ISO is not mounted.</p>
VirtualMachine.SoftwareN.ISOLocation	<p>Specifies the storage path that contains the ISO image file to be used by the application or script. Format the path as it appears on the host reservation, for example <code>netapp-1:it_nfs_1</code>. If a value is not specified, the ISO is not mounted.</p>

Custom Properties for Networking

The vRealize Automation custom properties for networking specify configuration for a specific network device on a machine.

NOTE This information does not apply to Amazon Web Services.

Network assignments are performed during machine allocation. vRealize Automation retrieves network information from the blueprint. If you want to assign more than one network, use the `VirtualMachine.NetworkN.Name` custom property on your machine blueprint. If you do not provide custom properties, allocation only assigns one network, which is picked using a round robin method in conjunction with the selected reservation.

By default, a machine has one network device configured with the `VirtualMachine.Network0.Name` property. You can configure additional network devices by using the `VirtualMachine.NetworkN.Name` custom property.

The numbering of network properties must be sequential, starting with 0. For example, if you specify custom properties for only `VirtualMachine.Network0` and `VirtualMachine.Network2`, the properties for `VirtualMachine.Network2` are ignored, because the preceding network, `VirtualMachine.Network1`, was not specified.

Table 2-17. Custom Properties for Networking Configuration

Custom Property	Description
<code>VirtualMachine.NetworkN.Address</code>	Specifies the IP address of network device <i>N</i> in a machine provisioned with a static IP address.
<code>VirtualMachine.NetworkN.MacAddressType</code>	<p>Indicates whether the MAC address of network device <i>N</i> is generated or user-defined (static). This property is available for cloning.</p> <p>The default value is generated. If the value is static, you must also use <code>VirtualMachine.NetworkN.MacAddress</code> to specify the MAC address.</p> <p><code>VirtualMachine.NetworkN</code> custom properties are designed to be specific to individual blueprints and machines. When a machine is requested, network and IP address allocation is performed before the machine is assigned to a reservation. Because blueprints are not guaranteed to be allocated to a specific reservation, do not use this property on a reservation.</p>
<code>VirtualMachine.NetworkN.MacAddress</code>	<p>Specifies the MAC address of a network device <i>N</i>. This property is available for cloning.</p> <p>If the value of <code>VirtualMachine.NetworkN.MacAddressType</code> is generated, this property contains the generated address.</p> <p>If the value of <code>VirtualMachine.Network.N.MacAddressType</code> is static, this property specifies the MAC address. For virtual machines provisioned on ESX server hosts, the address must be in the range specified by VMware. For details, see vSphere documentation.</p> <p><code>VirtualMachine.NetworkN</code> custom properties are designed to be specific to individual blueprints and machines. When a machine is requested, network and IP address allocation is performed before the machine is assigned to a reservation. Because blueprints are not guaranteed to be allocated to a specific reservation, do not use this property on a reservation.</p>

Table 2-17. Custom Properties for Networking Configuration (Continued)

Custom Property	Description
VirtualMachine.NetworkN.Name	<p>Specifies the name of the network to connect to, for example the network device <i>N</i> to which a machine is attached.</p> <p>By default, a network is assigned from the network paths available on the reservation on which the machine is provisioned.</p> <p>You can ensure that a network device is connected to a specific network by setting the value of this property to the name of a network on an available reservation.</p> <p>VirtualMachine.NetworkN custom properties are designed to be specific to individual blueprints and machines. When a machine is requested, network and IP address allocation is performed before the machine is assigned to a reservation. Because blueprints are not guaranteed to be allocated to a specific reservation, do not use this property on a reservation.</p> <p>You can add this property to a vApp (vCloud) blueprint or a vApp (vCloud) Component blueprint.</p>
VirtualMachine.NetworkN.PortID	<p>Specifies the port ID to use for network device <i>N</i> when using a dvPort group with a vSphere distributed switch.</p> <p>VirtualMachine.NetworkN custom properties are designed to be specific to individual blueprints and machines. When a machine is requested, network and IP address allocation is performed before the machine is assigned to a reservation. Because blueprints are not guaranteed to be allocated to a specific reservation, do not use this property on a reservation.</p>
VirtualMachine.NetworkN.ProfileName	<p>Specifies the name of a network profile from which to assign a static IP address to network device <i>N</i> or from which to obtain the range of static IP addresses that can be assigned to network device <i>N</i> of a cloned machine, where <i>N</i>=0 for the first device, 1 for the second, and so on.</p> <p>If a network profile is specified in the network path in the reservation on which the machine is provisioned, a static IP address is assigned from that network profile. You can ensure that a static IP address is assigned from a specific profile by setting the value of this property to the name of a network profile.</p>
<ul style="list-style-type: none"> ■ VirtualMachine.NetworkN.SubnetMask ■ VirtualMachine.NetworkN.Gateway ■ VirtualMachine.NetworkN.PrimaryDns ■ VirtualMachine.NetworkN.SecondaryDns ■ VirtualMachine.NetworkN.PrimaryWins ■ VirtualMachine.NetworkN.SecondaryWins ■ VirtualMachine.NetworkN.DnsSuffix ■ VirtualMachine.NetworkN.DnsSearchSuffixes 	<p>Appending a name allows you to create multiple versions of a custom property. For example, the following properties might list load balancing pools set up for general use and machines with high, moderate, and low performance requirements:</p> <ul style="list-style-type: none"> ■ VCNS.LoadBalancerEdgePool.Names ■ VCNS.LoadBalancerEdgePool.Names.moderate ■ VCNS.LoadBalancerEdgePool.Names.high ■ VCNS.LoadBalancerEdgePool.Names.low <p>NOTE In a multi-machine service, this custom property is supported only for VM components with Routed Virtual Wire. It is also not supported for pre-defined static IP addresses for VM components in the context of a multi-machine blueprint.</p> <p>Configures attributes of the network profile specified in VirtualMachine.NetworkN.ProfileName.</p>

Table 2-17. Custom Properties for Networking Configuration (Continued)

Custom Property	Description
<code>VCNS.LoadBalancerEdgePool.Names.name</code>	<p>Specifies the vCloud Networking and Security load balancing pools to which the virtual machine is assigned during provisioning. The virtual machine is assigned to all service ports of all specified pools. The value is an <i>edge/pool</i> name or a list of <i>edge/pool</i> names separated by commas. Names are case-sensitive.</p> <p>Appending a name allows you to create multiple versions of a custom property. For example, the following properties might list load balancing pools set up for general use and machines with high, moderate, and low performance requirements:</p> <ul style="list-style-type: none"> ■ <code>VCNS.LoadBalancerEdgePool.Names</code> ■ <code>VCNS.LoadBalancerEdgePool.Names.moderate</code> ■ <code>VCNS.LoadBalancerEdgePool.Names.high</code> ■ <code>VCNS.LoadBalancerEdgePool.Names.low</code> <p>NOTE In a multi-machine service, this custom property is supported only for VM components with Routed Virtual Wire. It is also not supported for pre-defined static IP addresses for VM components in the context of a multi-machine blueprint.</p>
<code>VCNS.SecurityGroup.Names.name</code>	<p>Specifies the vCloud Networking and Security security group or groups to which the virtual machine is assigned during provisioning. The value is a security group name or a list of names separated by commas. Names are case-sensitive.</p> <p>Appending a name allows you to create multiple versions of the property, which can be used separately or in combination. For example, the following properties can list security groups intended for general use, for the sales force, and for support:</p> <ul style="list-style-type: none"> ■ <code>VCNS.SecurityGroup.Names</code> ■ <code>VCNS.SecurityGroup.Names.sales</code> ■ <code>VCNS.SecurityGroup.Names.support</code>
<code>VCNS.SecurityTag.Names.name</code>	<p>Specifies the vCloud Networking and Security security tag or tags to which the virtual machine is associated during provisioning. The value is a security tag name or a list of names separated by commas. Names are case-sensitive.</p> <p>Appending a name allows you to create multiple versions of the property, which can be used separately or in combination. For example, the following properties can list security tags intended for general use, for the sales force, and for support:</p> <ul style="list-style-type: none"> ■ <code>VCNS.SecurityTag.Names</code> ■ <code>VCNS.SecurityTag.Names.sales</code> ■ <code>VCNS.SecurityTag.Names.support</code>

Custom Properties for PXE Provisioning

PXE is the only provisioning method supported for Cisco UCS Manager. You can use the network bootstrap program with vRealize Automation custom properties to initiate WIM, SCCM, or Linux Kickstart provisioning. You can also use custom properties to call your own PowerShell scripts. Linux Kickstart provisioning does not require custom properties.

Custom Properties for Provisioning With PowerShell Scripts

You can use these properties for calling PowerShell scripts.

Table 2-18. Custom Properties for Calling PowerShell Scripts

Custom Property	Description
Pxe.Setup.ScriptName	Specifies a custom EPI PowerShell script to run on the machine before it is started by using the PXE network boot program. The value is the name assigned to the script when it is uploaded to the model manager, for example <code>setup.ps1</code> .
Pxe.Clean.ScriptName	Specifies the name of an EPI PowerShell script installed in the vRealize Automation Model Manager, to run on the machine after it is provisioned. The value is the name assigned to the script when it is uploaded to the Model Manager, for example <code>clean.ps1</code> .

Custom Properties For PXE and SCCM Provisioning

You can use these properties for PXE and SCCM provisioning.

Table 2-19. Custom Properties for PXE and SCCM Provisioning

Custom Property	Description
SCCM.Collection.Name	Specifies the name of the SCCM collection that contains the operating system deployment task sequence.
SCCM.Server.Name	Specifies the fully qualified domain name of the SCCM server on which the collection resides, for example <code>lab-sccm.lab.local</code> .
SCCM.Server.SiteCode	Specifies the site code of the SCCM server.
SCCM.Server.UserName	Specifies a user name with administrator-level access to the SCCM server.
SCCM.Server.Password	Specifies the password associated with the <code>SCCM.Server.UserName</code> property.
SCCM.CustomVariable.	Specifies the value of a custom variable, where <i>Name</i> is the name of any custom variable to be made available to the SCCM task sequence after the provisioned machine is registered with the SCCM collection. The value is determined by your choice of custom variable. If your integration requires it, you can use <code>SCCM.RemoveCustomVariablePrefix</code> to remove the <code>SCCM.CustomVariable.</code> prefix from your custom variable.

Custom Properties For PXE and WIM Provisioning

You can use these properties for PXE and WIM provisioning.

Table 2-20. Custom Properties for PXE and WIM Provisioning

Custom Property	Description
Image.Network.Letter	Specifies the drive letter to which the WIM image path is mapped on the provisioned machine. The default value is <code>K</code> .
Image.WIM.Path	Specifies the UNC path to the WIM file from which an image is extracted during WIM-based provisioning. The path format is <code>\\server\share\$</code> format, for example <code>\\lab-ad\dfs\$</code> .
Image.WIM.Name	Specifies the name of the WIM file, for example <code>win2k8.wim</code> , as located by the <code>Image.WIM.Path</code> property.

Table 2-20. Custom Properties for PXE and WIM Provisioning (Continued)

Custom Property	Description
Image.WIM.Index	Specifies the index used to extract the correct image from the WIM file.
Image.Network.User	Specifies the user name with which to map the WIM image path (<code>Image.WIM.Path</code>) to a network drive on the provisioned machine. This is typically a domain account with access to the network share.
Image.Network.Password	Specifies the password associated with the <code>Image.Network.User</code> property.
SysPrep. <i>Section.Key</i> ■ SysPrep.GuiUnattended.AdminPassword ■ SysPrep.GuiUnattended.EncryptedAdminPassword ■ SysPrep.GuiUnattended.TimeZone	<p>Specifies information to be added to the SysPrep answer file on machines during the WinPE stage of provisioning. Information that already exists in the SysPrep answer file is overwritten by these custom properties. <i>Section</i> represents the name of the section of the SysPrep answer file, for example <code>GuiUnattended</code> or <code>UserData</code>. <i>Key</i> represents a key name in the section. For example, to set the time zone of a provisioned machine to West Pacific Standard Time, define the custom property <code>GuiUnattended.UserData.TimeZone</code> and set the value to 275.</p> <p>For a full list of sections, keys, and accepted values, see the System Preparation Utility for Windows documentation. The following <i>Section.Key</i> combinations can be specified for WIM-based provisioning:</p> <ul style="list-style-type: none"> ■ GuiUnattended <ul style="list-style-type: none"> ■ AdminPassword ■ EncryptedAdminPassword ■ TimeZone ■ UserData <ul style="list-style-type: none"> ■ ProductKey ■ FullName ■ ComputerName ■ OrgName ■ Identification <ul style="list-style-type: none"> ■ DomainAdmin ■ DomainAdminPassword ■ JoinDomain ■ JoinWorkgroup
Sysprep.Identification.DomainAdmin	Specifies a user name with administrator-level access to the target domain in Active Directory. Do not include the user domain in the credentials that you send to vCloud Director.
Sysprep.Identification.DomainAdminPassword	Specifies the password to associate with the <code>Sysprep.Identification.DomainAdmin</code> property.
Sysprep.Identification.JoinDomain	Specifies the name of the domain to join in Active Directory.
Sysprep.Identification.JoinWorkgroup	Specifies the name of the workgroup to join if not using a domain.
SysPrep.UserData.ComputerName	Specifies a machine name, for example <code>lab-client005</code> .
SysPrep.UserData.FullName	Specifies the full name of a user.
SysPrep.UserData.OrgName	Specifies the organization name of the user.
SysPrep.UserData.ProductKey	Specifies the Windows product key.

Custom Properties for BMC BladeLogic Configuration Manager Integration

vRealize Automation includes custom properties that you can use to provide additional controls for BMC BladeLogic Configuration Manager integration.

Table 2-21. Custom Properties Required for BMC BladeLogic Configuration Manager Integrations

Custom Property	Description
<code>VirtualMachine.EPI.Type</code>	Specifies the type of external provisioning infrastructure.
<code>BMC.Software.Install</code>	Set to True to enable BMC BladeLogic Configuration Manager integration.
<code>EPI.Server.Name</code>	Specifies the name of the external provisioning infrastructure server, for example, the name of the server hosting BMC BladeLogic. If at least one general BMC EPI agent was installed without specifying a BMC BladeLogic Configuration Manager host, this value directs the request to the desired server. If only dedicated BMC EPI agents for specific BMC BladeLogic Configuration Manager hosts were installed, this value must exactly match the server name configured for one of these agents.
<code>BMC.Service.Profile</code>	Specifies the name of the default authentication profile on the BMC BladeLogic server.
<code>BMC.Software.BatchLocation</code>	Specifies the location in BMC BladeLogic configuration where software jobs are deployed. This value must match either the appropriate field in <code>Website\Software.txt</code> or the appropriate value of <code>VRM.Software.IdNNNN</code> , depending on the method used to prepare software jobs for integration.
<code>VMware.VirtualCenter.OperatingSystem</code>	Specifies the vCenter Server guest operating system version (<code>VirtualMachineGuestOsIdentifier</code>) with which vCenter Server creates the machine. This operating system version must match the operating system version to be installed on the provisioned machine. Administrators can create property groups using one of several property sets, for example, <code>VMware[OS_Version]Properties</code> , that are predefined to include the correct <code>VMware.VirtualCenter.OperatingSystem</code> values. This property is for virtual provisioning. For related information, see the enumeration type <code>VirtualMachineGuestOsIdentifier</code> in vSphere API/SDK Documentation. For a list of currently accepted values, see the VMware vCenter Server™ documentation.

Custom Properties To Make BMC BladeLogic Configuration Manager Software Jobs Available

Configure BMC BladeLogic Configuration Manager jobs for vRealize Automation integrations. Make all software jobs available to machine requesters to select from, or specify a software job to apply to all machines provisioned from the blueprint.

Table 2-22. Custom Properties to Make Software Jobs Available

Custom Property	Description
LoadSoftware	Set to True to enable software install options and make the software jobs listed in Website\Software.txt available to the user requesting the machine.
VRM.Software.IdNNNN	Specifies a software job or policy to be applied to all machines provisioned from the blueprint. Set the value to job_type=job_path, where job_type is the numeral that represents the BMC BladeLogic job type and job_path is the location of the job in BMC BladeLogic, for example 4=/Utility/putty. NNNN is a number from 1000 to 1999. 1 - AuditJob 2 - BatchJob 3 - ComplianceJob 4 - DeployJob 5 - FileDeployJob 6 - NSHScriptJob 7 - PatchAnalysisJob 8 - SnapshotJob

Optional Custom Properties for BMC BladeLogic Configuration Manager Integrations

You can also use optional custom properties that are commonly used with BMC BladeLogic Configuration Manager blueprints.

Table 2-23. Optional Custom Properties for BMC BladeLogic Configuration Manager Integrations

Property	Definition
BMC.AddServer.Delay	Specifies the number of seconds to wait before adding the machine to BMC BladeLogic Configuration Manager. The default is 30.
BMC.AddServer.Retry	Specifies the number of seconds to wait before retrying if the first attempt to add the machine to BMC BladeLogic Configuration Manager is unsuccessful. The default is 100.

Custom Properties for HP Server Automation Integration

vRealize Automation includes custom properties that you can use to provide additional controls for HP Server Automation integration. Some custom properties are required for HP Server Automation integration. Other custom properties are optional.

Required Custom Properties for HP Server Automation Integration

Certain custom properties are required for a blueprint to work with HP Server Automation.

Table 2-24. Required Custom Properties for HP Server Automation Integration

Property	Definition
<code>VMware.VirtualCenter.OperatingSystem</code>	Specifies the vCenter Server guest operating system version (<code>VirtualMachineGuestOsIdentifier</code>) with which vCenter Server creates the machine. This operating system version must match the operating system version to be installed on the provisioned machine. Administrators can create property groups using one of several property sets, for example, <code>VMware[OS_Version]Properties</code> , that are predefined to include the correct <code>VMware.VirtualCenter.OperatingSystem</code> values. This property is for virtual provisioning.
<code>VirtualMachine.EPI.Type</code>	Specifies the type of external provisioning infrastructure.
<code>EPI.Server.Name</code>	Specifies the name of the external provisioning infrastructure server, for example, the name of the server hosting BMC BladeLogic. If at least one general BMC EPI agent was installed without specifying a BMC BladeLogic Configuration Manager host, this value directs the request to the desired server.
<code>Opware.Software.Install</code>	Set to True to allow HP Server Automation to install software.
<code>Opware.Server.Name</code>	Specifies the fully qualified name of the HP Server Automation server.
<code>Opware.Server.Username</code>	Specifies the user name provided when a password file in the agent directory was created, for example <code>opswareadmin</code> . This user name requires administrative access to the HP Server Automation instance.
<code>Opware.BootImage.Name</code>	Specifies the boot image value as defined in HP Server Automation for the 32-bit WinPE image, for example <code>winpe32</code> . The property is not required when provisioning by cloning.
<code>Opware.Customer.Name</code>	Specifies a customer name value as defined in HP Server Automation, for example <code>MyCompanyName</code> .
<code>Opware.Facility.Name</code>	Specifies a facility name value as defined in HP Server Automation, for example <code>Cambridge</code> .
<code>Opware.Machine.Password</code>	Specifies the default local administrator password for an operating system sequence WIM image such as <code>Opware.OSSequence.Name</code> as defined in HP Server Automation, for example <code>P@ssword1</code> .
<code>Opware.OSSequence.Name</code>	Specifies the operating system sequence name value as defined in HP Server Automation, for example <code>Windows 2008 WIM</code> .
<code>Opware.Realm.Name</code>	Specifies the realm name value as defined in HP Server Automation, for example <code>Production</code> .
<code>Opware.Register.Timeout</code>	Specifies the time, in seconds, to wait for creation of a provisioning job to complete.
<code>VirtualMachine.CDRom.Attach</code>	Set to False to provision the machine without a CD-ROM device. The default is True.
<code>Linux.ExternalScript.Name</code>	Specifies the name of an optional customization script, for example <code>config.sh</code> , that the Linux guest agent runs after the operating system is installed. This property is available for Linux machines cloned from templates on which the Linux agent is installed.

Table 2-24. Required Custom Properties for HP Server Automation Integration (Continued)

Property	Definition
Linux.ExternalScript.LocationType	Specifies the location type of the customization script named in the Linux.ExternalScript.Name property. This can be either local or nfs.
Linux.ExternalScript.Path	Specifies the local path to the Linux customization script or the export path to the Linux customization on the NFS server. The value must begin with a forward slash and not include the file name, for example /scripts/linux/config.sh.

Optional Custom Properties for HP Server Automation Integration

Certain custom properties are optional for a blueprint to work with HP Server Automation.

Table 2-25. Optional Custom Properties for HP Server Automation Integration

Property	Definition
Opware.ProvFail.Notify	(Optional) Specifies the notification email address for HP Server Automation to use in the event of provisioning failure, for example provisionfail@lab.local.
Opware.ProvFail.Notify	(Optional) Specifies the HP Server Automation user to whom ownership is assigned if provisioning fails.
Opware.ProvSuccess.Notify	(Optional) Specifies the notification email address for HP Server Automation to use if provisioning is successful.
Opware.ProvSuccess.Owner	(Optional) Specifies the HP Server Automation user to whom ownership is assigned if provisioning is successful.

Custom Properties That Make HP Server Automation Software Jobs Available

Depending on how your fabric administrator configures HP Server Automation jobs for vRealize Automation integration, you might have a choice between making all software jobs available to machine requesters to select, or you can specify jobs to apply to all machines provisioned from your blueprint.

Table 2-26. Custom Properties to Make Software Jobs Available

Property	Definition
LoadSoftware	Set to True to enable software install options and make the software jobs listed in Website\Software.txt available to the user requesting the machine.
VRM.Software.Id	(Optional) Specifies an HP Server Automation policy to be applied to all machines provisioned from the blueprint. NNNN is a number from 1000 to 1999.

Custom Properties Grouped by Name

You can use custom properties to provide additional vRealize Automation controls.

These properties are also grouped functionally and described in relevant guides in vRealize Automation product documentation. See [Chapter 2, “Custom Properties Grouped by Function,”](#) on page 19.

This chapter includes the following topics:

- [“Custom Properties A Table,”](#) on page 49
- [“Custom Properties B Table,”](#) on page 50
- [“Custom Properties C Table,”](#) on page 51
- [“Custom Properties E Table,”](#) on page 52
- [“Custom Properties H Table,”](#) on page 52
- [“Custom Properties I Table,”](#) on page 53
- [“Custom Properties L Table,”](#) on page 54
- [“Custom Properties M Table,”](#) on page 55
- [“Custom Properties O Table,”](#) on page 55
- [“Custom Properties P Table,”](#) on page 56
- [“Custom Properties R Table,”](#) on page 57
- [“Custom Properties S Table,”](#) on page 57
- [“Custom Properties V Table,”](#) on page 59
- [“Custom Properties X Table,”](#) on page 76

Custom Properties A Table

This section lists vRealize Automation custom properties that begin with the letter A.

Table 3-1. Custom Properties A Table

Property	Description
AD.Lookup.Department	Specifies the cost center value that is included in a notification email sent to approvers. This property value must be specified in the blueprint.
Amazon.Placement.Tenancy	Set to = dedicated to specify that the AWS connection be specific to a dedicated tenant. This property is valid for use with VPC subnets.

Table 3-1. Custom Properties A Table (Continued)

Property	Description
<code>amazon.AmazonEC2Config.ServiceURL</code>	Specifies the Amazon configuration service URL for Amazon GovCloud, for example <code>amazon.AmazonEC2Config.ServiceURL=https://ec2.us-gov-west-1.amazonaws.com</code> .
<code>amazon.ElasticLoadBalancingConfig.ServiceURL</code>	Specifies the Amazon load balancer configuration service URL for Amazon GovCloud, for example <code>amazon.ElasticLoadBalancingConfig.ServiceURL=https://elasticloadbalancing.us-gov-west-1.amazonaws.com</code> .
<code>Amazon.ElasticLoadBalancer.Names</code>	Assigns machines that are provisioned by a blueprint to the elastic load balancers that match the specified values. This property is valid for vSphere, Amazon, and Hyper-V configurations.
<code>amazon.IAMInstanceProfile.ARN</code>	Specifies the AWS Identity and Access Management (IAM) instance profile Amazon Resource Names (ARNs) when requesting an AWS instance. When you add this property, for example <code>amazon.IAMInstanceProfile.ARN = IAM Instance Profile ARN(s) value</code> , to a blueprint and then request provisioning from the catalog, the provisioned Amazon virtual machine or instance contains the specified IAM role.
<code>Amazon.Instance.Id</code>	Specifies the Amazon instance ID of a machine provisioned on an Amazon EC2 endpoint. This property is valid for vSphere and Amazon configurations.
<code>AppService.SyncMachines.MachineProvisioned</code>	Used specifically in combination with the procedure described in vRealize Automation KB 2132084 to address the issue of a multi-machine blueprint provisioning process being only partially successful when in fact all the components are correctly provisioned and deployments are accessible from the Items and Managed Machines tabs. For vRealize Automation 6.2.5, you only need to add the <code>AppService.SyncMachines.MachineProvisioned</code> custom property to your blueprint to avoid this issue. For details, see the VMware Knowledge Base article <i>Multi-Machine Blueprint Reported as Partially Succeeded But All the Components Provisioned Correctly (2132084)</i> at http://kb.vmware.com/kb/2132084 for more information.

Custom Properties B Table

This section lists vRealize Automation custom properties that begin with the letter B.

Table 3-2. Custom Properties B Table

Property	Definition
<code>BMC.AddServer.Delay</code>	Specifies the number of seconds to wait before adding the machine to BMC BladeLogic Configuration Manager. The default is 30.
<code>BMC.AddServer.Retry</code>	Specifies the number of seconds to wait before retrying if the first attempt to add the machine to BMC BladeLogic Configuration Manager is unsuccessful. The default is 100.
<code>BMC.Service.Profile</code>	Specifies the name of the default authentication profile on the BMC BladeLogic server.

Table 3-2. Custom Properties B Table (Continued)

Property	Definition
BMC.Software.BatchLocation	Specifies the location in BMC BladeLogic configuration where software jobs are deployed. This value must match either the appropriate field in Website\Software.txt or the appropriate value of VRM.Software.IdNNNN, depending on the method used to prepare software jobs for integration.
BMC.Software.Install	Set to True to enable BMC BladeLogic Configuration Manager integration.

Custom Properties C Table

This section lists vRealize Automation custom properties that begin with the letter C.

Table 3-3. Custom Properties C Table

Property	Definition
Cisco.Organization.Dn	Specifies the distinguished name of the Cisco UCS Manager organization in which Cisco UCS machines provisioned by the business group are placed, for example org-root/org-Engineering. If the specified organization does not exist in the Cisco UCS Manager instance that is managing the machine, provisioning fails. This property is available for business groups only.
CloneFrom	Specifies the name of an existing machine or virtualization platform object to clone from, for example a template in vCenter Server such as Win2k8tmpl.
CloneSpec	Specifies the name of a customization specification on a cloned machine, for example a predefined SysPrep object in vCenter Server such as Win2k Customization Spec. The default value is specified on the blueprint.
Command.DiskPart.Options	When you use WIM-based virtual provisioning on ESX server hosts, set to Align=64 to use the recommended alignment parameters when you format and partition the machine's disk. This property is not available for physical provisioning.
Command.FormatDisk.Options	When you use WIM-based virtual provisioning on ESX server hosts, set to /A:32K to use the recommended alignment parameters when you format and partition the machine's disk. This property is not available for physical provisioning.

Custom Properties E Table

This section lists vRealize Automation custom properties that begin with the letter E.

Table 3-4. Custom Properties E Table

Property	Definition
<code>EPI.Server.Collection</code>	Specifies the name of the Citrix provisioning collection to which the machine is to be registered.
<code>EPI.Server.Name</code>	<p>Specifies the name of the external provisioning infrastructure server, for example, the name of the server hosting BMC BladeLogic. If at least one general BMC EPI agent was installed without specifying a BMC BladeLogic Configuration Manager host, this value directs the request to the desired server.</p> <p>If only dedicated BMC EPI agents for specific BMC BladeLogic Configuration Manager hosts were installed, this value must exactly match the server name configured for one of these agents.</p> <p>Specifies the name of the server hosting HP Server Automation. If at least one general Opsware EPI agent was installed without specifying a server automation server, this value directs the request to the desired server.</p> <p>If only dedicated EPI agents for specific HP server automation servers were installed, this value must exactly match the server name configured for one of these agents.</p> <p>If at least one general EPI agent of the appropriate type (<code>VirtualMachine.EPI.Type</code>) was installed without specifying a server, this value directs the request to the desired server. If only dedicated EPI agents for specific servers of the appropriate type were installed, this value must exactly match the server name configured for one of these agents.</p>
<code>EPI.Server.Port</code>	Specifies the port on which to contact the provisioning server. If you are using a Citrix provisioning server, omit to specify the default port value of 54321.
<code>EPI.Server.Site</code>	Specifies the name of the Citrix provisioning site that contains the collection and store identified by the <code>EPI.Server.Collection</code> and <code>EPI.Server.Store</code> properties, for example <code>site1</code> .
<code>EPI.Server.Store</code>	Specifies the name of the Citrix provisioning store that contains the vDisk identified by the <code>EPI.Server.VDiskName</code> property, for example <code>store1</code> .
<code>EPI.Server.VDiskName</code>	Specifies the name of the Citrix provisioning vDisk from which to provision, for example <code>disk1</code> .

Custom Properties H Table

This section lists vRealize Automation custom properties that begin with the letter H.

Table 3-5. Custom Properties H Table

Property	Definition
<code>Hostname</code>	Specifies the host machine name, overriding the generated machine name contained in the <code>VirtualMachine.Admin.Name</code> property. If <code>Hostname</code> is not used, the <code>VirtualMachine.Admin.Name</code> value is used as the machine name. This property is not valid for multi-machine blueprints.
<code>Hyperv.Network.Type</code>	Specifies the network adapter type of the virtual machine. This property is valid for use with Hyper-V (SCVMM) only. When the value is set to <code>synthetic</code> , specifies that the blueprint be allowed to provision a Generation-2 machine on a Hyper-V (SCVMM) 2012 R2 resource. Generation-2 provisioning also requires that the blueprint includes the <code>Scvmm.Generation2 = true</code> property setting. The legacy value is not compatible with WinXP or Server 2003 x64 guest operating systems. The default value is <code>synthetic</code> .

Custom Properties I Table

This section lists vRealize Automation custom properties that begin with the letter I.

Table 3-6. Custom Properties I Table

Property	Definition
<code>Image.ISO.Location</code>	<p>Values for this property are case sensitive. Specifies the location of the ISO image from which to boot, for example <code>http://192.168.2.100/site2/winpe.iso</code>. The format of this value depends on your platform. For details, see documentation provided for your platform. This property is required for WIM-based provisioning, Linux Kickstart and autoYaST provisioning, and SCCM-based provisioning.</p> <p>For virtual provisioning with vCenter Server, this specifies the name of a datastore in the instance that will be accessible to the provisioning compute resource. For virtual provisioning with XenServer, this specifies the name of a storage repository.</p> <p>For physical provisioning, this specifies the HTTP URL of the web-accessible location of the image.</p>
<code>Image.ISO.Name</code>	<p>Values for this property are case sensitive. Specifies the name of the ISO image from which to boot, for example <code>/ISO/Microsoft/WinPE.iso</code>. The format of this value depends on your platform. For details, see documentation provided for your platform. This property is required for WIM-based provisioning, Linux Kickstart and autoYaST provisioning, and SCCM-based provisioning.</p> <p>For virtual provisioning with vCenter Server, this value specifies the path to the image in the datastore specified by <code>Image.ISO.Location</code>, for example <code>/MyISOs/Microsoft/MSDN/win2003.iso</code>. The value must use forward slashes and begin with a forward slash. For virtual provisioning with XenServer, this value specifies the name of the image in the storage repository specified by <code>Image.ISO.Location</code>. In virtual provisioning with Hyper-V, this value specifies the full local path to the image.</p> <p>For physical provisioning, this value specifies the file name of the image.</p>
<code>Image.ISO.UserName</code>	<p>Specifies the user name to access the CIFS share in the format <code>username@domain</code>. For Dell iDRAC integrations where the image is located on a CIFS share that requires authentication to access.</p>
<code>Image.ISO.Password</code>	<p>Specifies the password associated with the <code>Image.ISO.UserName</code> property. For Dell iDRAC integrations where the image is located on a CIFS share that requires authentication to access.</p>
<code>Image.WIM.Path</code>	<p>Specifies the UNC path to the WIM file from which an image is extracted during WIM-based provisioning. The path format is <code>\\server\share\$</code> format, for example <code>\\lab-ad\dfs\$</code>.</p>
<code>Image.WIM.Name</code>	<p>Specifies the name of the WIM file, for example <code>win2k8.wim</code>, as located by the <code>Image.WIM.Path</code> property.</p>
<code>Image.WIM.Index</code>	<p>Specifies the index used to extract the correct image from the WIM file.</p>
<code>Image.Network.User</code>	<p>Specifies the user name with which to map the WIM image path (<code>Image.WIM.Path</code>) to a network drive on the provisioned machine. This is typically a domain account with access to the network share.</p>
<code>Image.Network.Password</code>	<p>Specifies the password associated with the <code>Image.Network.User</code> property.</p>
<code>Image.Network.Letter</code>	<p>Specifies the drive letter to which the WIM image path is mapped on the provisioned machine. The default value is K.</p>
<code>Infrastructure.Admin.MachineObjectOU</code>	<p>Specifies the organizational unit (OU) of the machine. When machines are placed in the required OU by the business group OU setting, this property is not required.</p>
<code>Infrastructure.Admin.ADUser</code>	<p>Specifies the domain administrator user ID. This identifier is used to query Active Directory users and groups when an anonymous bind cannot be used.</p>

Table 3-6. Custom Properties I Table (Continued)

Property	Definition
Infrastructure.Admin.ADPassword	Specifies the password associated with the Infrastructure.Admin.ADUser domain administrator user ID.
Infrastructure.Admin.DefaultDomain	Specifies the default domain on the machine.
Infrastructure.ResourcePool.Name	Specifies the resource pool to which the machine belongs, if any. The default is the value specified in the reservation from which the machine was provisioned.

Custom Properties L Table

This section lists vRealize Automation custom properties that begin with the letter L.

Table 3-7. Custom Properties L Table

Property	Description
Linux.ExternalScript.LocationType	Specifies the location type of the customization script named in the Linux.ExternalScript.Name property. This can be either local or nfs. You must also specify the script location using the Linux.ExternalScript.Path property. If the location type is nfs, also use the Linux.ExternalScript.Server property.
Linux.ExternalScript.Name	Specifies the name of an optional customization script, for example config.sh, that the Linux guest agent runs after the operating system is installed. This property is available for Linux machines cloned from templates on which the Linux agent is installed. If you specify an external script, you must also define its location by using the Linux.ExternalScript.LocationType and Linux.ExternalScript.Path properties.
Linux.ExternalScript.Path	Specifies the local path to the Linux customization script or the export path to the Linux customization on the NFS server. The value must begin with a forward slash and not include the file name, for example /scripts/linux/config.sh.
Linux.ExternalScript.Server	Specifies the name of the NFS server, for example lab-ad.lab.local, on which the Linux external customization script named in Linux.ExternalScript.Name is located.
LoadSoftware	Set to True to enable software install options and make the software jobs listed in Website\Software.txt available to the user requesting the machine.

Custom Properties M Table

This section lists vRealize Automation custom properties that begin with the letter M.

Table 3-8. Custom Properties M Table

Property	Description
MaximumProvisionedMachines	Specifies the maximum number of linked clones for one machine snapshot. The default is unlimited.
Machine.SSH	Set to True to enable the Connect Using SSH option, on the vRealize Automation Items page, for Linux machines provisioned from this blueprint. If set to True and the Connect using RDP or SSH machine operation is enabled in the blueprint, all Linux machines that are provisioned from the blueprint display the Connect Using SSH option to entitled users.

Custom Properties O Table

This section lists vRealize Automation custom properties that begin with the letter O.

Table 3-9. Custom Properties O Table

Property	Description
Opware.BootImage.Name	Specifies the boot image value as defined in HP Server Automation for the 32-bit WinPE image, for example winpe32. The property is not required when provisioning by cloning.
Opware.Customer.Name	Specifies a customer name value as defined in HP Server Automation, for example MyCompanyName.
Opware.Facility.Name	Specifies a facility name value as defined in HP Server Automation, for example Cambridge.
Opware.Machine.Password	Specifies the default local administrator password for an operating system sequence WIM image such as Opware.OSSequence.Name as defined in HP Server Automation, for example P@ssword1.
Opware.OSSequence.Name	Specifies the operating system sequence name value as defined in HP Server Automation, for example Windows 2008 WIM.
Opware.ProvFail.Notify	(Optional) Specifies the notification email address for HP Server Automation to use in the event of provisioning failure, for example provisionfail@lab.local.
Opware.ProvFail.Owner	(Optional) Specifies the HP Server Automation user to whom ownership is assigned if provisioning fails.
Opware.ProvSuccess.Notify	(Optional) Specifies the notification email address for HP Server Automation to use if provisioning is successful.
Opware.ProvSuccess.Owner	(Optional) Specifies the HP Server Automation user to whom ownership is assigned if provisioning is successful.
Opware.Realm.Name	Specifies the realm name value as defined in HP Server Automation, for example Production.
Opware.Register.Timeout	Specifies the time, in seconds, to wait for creation of a provisioning job to complete.
Opware.Server.Name	Specifies the fully qualified name of the HP Server Automation server.

Table 3-9. Custom Properties O Table (Continued)

Property	Description
Opware.Server.Username	Specifies the user name provided when a password file in the agent directory was created, for example opswreadmin. This user name requires administrative access to the HP Server Automation instance.
Opware.Software.Install	Set to True to allow HP Server Automation to install software.

Custom Properties P Table

This section lists vRealize Automation custom properties that begin with the letter P.

Table 3-10. Custom Properties P Table

Property	Description
Plugin.AdMachineCleanup.Delete	Set to True to delete the accounts of destroyed machines, instead of disabling them.
Plugin.AdMachineCleanup.Execute	Set to True to enable the Active Directory cleanup plug-in. By default, each machine's account is disabled when it is destroyed.
Plugin.AdMachineCleanup.MoveToOu	Moves the account of destroyed machines to a new Active Directory organizational unit. The value is the organization unit to which you are moving the account. This value must be in <i>ou=OU, dc=dc</i> format, for example <i>ou=trash,cn=computers,dc=lab,dc=local</i> .
Plugin.AdMachineCleanup.UserName	Specifies an Active Directory account user name with sufficient privileges to perform Active Directory actions such as delete, disable, rename, or move Active Directory accounts. The value must be in <i>domain\username</i> format, for example <i>lab\administrator</i> . This property is required if the vRealize Automation manager service does not have these rights in a domain, which can occur when you provision machines in more than one domain.
Plugin.AdMachineCleanup.Password	Specifies the password associated to the <code>Plugin.AdMachineCleanup.UserName</code> property.
Plugin.AdMachineCleanup.Domain	Specifies the Active Directory domain name that contains the machine account to be destroyed.
Plugin.AdMachineCleanup.RenamePrefix	Renames the accounts of destroyed machines by adding a prefix. The value is the prefix string to prepend, for example <i>destroyed_</i> .
Pxe.Clean.ScriptName	Specifies the name of an EPI PowerShell script installed in the vRealize Automation Model Manager, to run on the machine after it is provisioned. The value is the name assigned to the script when it is uploaded to the Model Manager, for example <i>clean.ps1</i> .
Pxe.Setup.ScriptName	Specifies a custom EPI PowerShell script to run on the machine before it is started by using the PXE network boot program. The value is the name assigned to the script when it is uploaded to the model manager, for example <i>setup.ps1</i> .

Custom Properties R Table

This section lists vRealize Automation custom properties that begin with the letter R.

Table 3-11. Custom Properties R Table

Property	Description
RDP.File.Name	Specifies an RDP file from which to obtain settings, for example <code>My_RDP_Settings.rdp</code> . The file must reside in the <code>Website\Rdp</code> subdirectory of the vRealize Automation installation directory.

Custom Properties S Table

This section lists vRealize Automation custom properties that begin with the letter S.

Table 3-12. Custom Properties S Table

Property	Description
SysPrep. <i>Section.Key</i> ■ SysPrep.GuiUnattended.AdminPassword ■ SysPrep.GuiUnattended.EncryptedAdminPassword ■ SysPrep.GuiUnattended.TimeZone	Specifies information to be added to the SysPrep answer file on machines during the WinPE stage of provisioning. Information that already exists in the SysPrep answer file is overwritten by these custom properties. <i>Section</i> represents the name of the section of the SysPrep answer file, for example <code>GuiUnattended</code> or <code>UserData</code> . <i>Key</i> represents a key name in the section. For example, to set the time zone of a provisioned machine to West Pacific Standard Time, define the custom property <code>GuiUnattended.UserData.TimeZone</code> and set the value to 275. For a full list of sections, keys, and accepted values, see the System Preparation Utility for Windows documentation. The following <i>Section.Key</i> combinations can be specified for WIM-based provisioning: <ul style="list-style-type: none"> ■ GuiUnattended <ul style="list-style-type: none"> ■ AdminPassword ■ EncryptedAdminPassword ■ TimeZone ■ UserData <ul style="list-style-type: none"> ■ ProductKey ■ FullName ■ ComputerName ■ OrgName ■ Identification <ul style="list-style-type: none"> ■ DomainAdmin ■ DomainAdminPassword ■ JoinDomain ■ JoinWorkgroup
Sysprep.Identification.DomainAdmin	Specifies a user name with administrator-level access to the target domain in Active Directory. Do not include the user domain in the credentials that you send to vCloud Director.
Sysprep.Identification.DomainAdminPassword	Specifies the password to associate with the <code>Sysprep.Identification.DomainAdmin</code> property.
Sysprep.Identification.JoinDomain	Specifies the name of the domain to join in Active Directory.

Table 3-12. Custom Properties S Table (Continued)

Property	Description
<code>Sysprep.Identification.JoinWorkgroup</code>	Specifies the name of the workgroup to join if not using a domain.
<code>SysPrep.UserData.ComputerName</code>	Specifies a machine name, for example lab-client005.
<code>SysPrep.UserData.FullName</code>	Specifies the full name of a user.
<code>SysPrep.UserData.OrgName</code>	Specifies the organization name of the user.
<code>SysPrep.UserData.ProductKey</code>	Specifies the Windows product key.
<code>SCCM.Collection.Name</code>	Specifies the name of the SCCM collection that contains the operating system deployment task sequence.
<code>SCCM.CustomVariable.Name</code>	Specifies the value of a custom variable, where <i>Name</i> is the name of any custom variable to be made available to the SCCM task sequence after the provisioned machine is registered with the SCCM collection. The value is determined by your choice of custom variable. If your integration requires it, you can use <code>SCCM.RemoveCustomVariablePrefix</code> to remove the <code>SCCM.CustomVariable.</code> prefix from your custom variable.
<code>SCCM.Server.Name</code>	Specifies the fully qualified domain name of the SCCM server on which the collection resides, for example lab-sccm.lab.local.
<code>SCCM.Server.SiteCode</code>	Specifies the site code of the SCCM server.
<code>SCCM.Server.UserName</code>	Specifies a user name with administrator-level access to the SCCM server.
<code>SCCM.Server.Password</code>	Specifies the password associated with the <code>SCCM.Server.UserName</code> property.
<code>SCCM.RemoveCustomVariablePrefix</code>	Set to <i>true</i> to remove the prefix <code>SCCM.CustomVariable.</code> from SCCM custom variables you created by using the custom property <code>SCCM.CustomVariable.Name</code> .
<code>Scvmm.Generation2</code>	When set to true, specifies that the blueprint be allowed to provision a Generation-2 machine on a Hyper-V (SCVMM) 2012 R2 resource. Generation-2 provisioning also requires that the blueprint includes the <code>Hyperv.Network.Type = synthetic</code> property setting.
<code>Snapshot.Policy.AgeLimit</code>	Sets the age limit, in days, for snapshots that can be applied to machines. This property applies to vSphere provisioning. When a snapshot exceeds the age limit, the Apply option is no longer available. When the snapshot age limit is reached, the snapshot remains but you can no longer revert to it. You can delete the snapshot using the vSphere client.

Table 3-12. Custom Properties S Table (Continued)

Property	Description
Snapshot.Policy.Limit	<p>Sets the number of snapshots allowed per machine. The default setting is one snapshot per machine. This property applies to vSphere provisioning. When set to 0, the blueprint option to create a snapshot is hidden for all users except for support and manager roles.</p> <p>Snapshots are shown in a hierarchical structure.</p> <ul style="list-style-type: none"> ■ Depth – Maximum is 31. ■ Width – There is no limit.
Snapshot.Policy.Disable	<p>When set to true, the ability to create a snapshot is disabled for all vRealize Automation user roles and the snapshot option is hidden from the Items tab.</p>

Custom Properties V Table

This section lists vRealize Automation custom properties that begin with the letter V.

Table 3-13. Custom Properties V Table

This section lists vRealize Automation custom properties that begin with the letter V.	
Property	Description
VbScript.PreProvisioning.Name	<p>Specifies the full path of a Visual Basic script to be run before a machine is provisioned. For example, %System-Drive%\Program Files(x86)\VMware\VCAC Agents\EPI_Agent\Scripts\SendEmail.vbs. The script file must reside on the system on which the Visual Basic script EPI agent is installed.</p>
VbScript.PostProvisioning.Name	<p>Specifies the full path of a Visual Basic script to be run after a machine is provisioned. For example, %System-Drive%\Program Files(x86)\VMware\VCAC Agents\EPI_Agent\Scripts\SendEmail.vbs. The script file must reside on the system on which the Visual Basic script EPI agent is installed.</p>
VbScript.UnProvisioning.Name	<p>Specifies the full path of a Visual Basic script to be run when a machine is destroyed. For example, %System-Drive%\Program Files (x86)\VMware\VCAC Agents\EPI_Agent\Scripts\SendEmail.vb. The script file must reside on the system on which the Visual Basic script EPI agent is installed.</p>
VCloud.Lease.Sync.TimeBufferMins	<p>Specifies a threshold integer value for a compute resource such that lease synchronization between vCloud Director and vRealize Automation will only occur for vCloud Director or vCloud Air provisioned machines that are set to expire in vCloud Director or vCloud Air in that time period. If a conflict is found, the lease value is synchronized to match the lease length defined in vRealize Automation. The default VCloud.Lease.Sync.TimeBufferMins value is 720 minutes, which is 12 hours. If VCloud.Lease.Sync.TimeBufferMins is not present, the default value is used. For example, if the default values are used, vRealize Automation will run the lease synchronization check workflow every 45 minutes, which is the workflow default, and only the leases of machines that are set to expire within 12 hours are changed to match the lease length defined in vRealize Automation.</p>

Table 3-13. Custom Properties V Table (Continued)

This section lists vRealize Automation custom properties that begin with the letter V.

Property	Description
VCloud.Template.MakeIdenticalCopy	<p>Set to True to clone an identical copy of the vApp template in vCloud Director and provision the results in vRealize Automation. This ignores all settings specified in the blueprints except the name of the vApp and its virtual machines. The storage path specified in the vApp template during cloning is used, even if a different storage path is specified in a vApp component blueprint or when requesting a vApp machine.</p> <p>Set to False to clone a copy of the vApp template with settings specified by the vApp and vApp component blueprints.</p>
VCNS.LoadBalancerEdgePool.Names.name	<p>Specifies the vCloud Networking and Security load balancing pools to which the virtual machine is assigned during provisioning. The virtual machine is assigned to all service ports of all specified pools. The value is an <i>edge/pool</i> name or a list of <i>edge/pool</i> names separated by commas. Names are case-sensitive.</p> <p>Appending a name allows you to create multiple versions of a custom property. For example, the following properties might list load balancing pools set up for general use and machines with high, moderate, and low performance requirements:</p> <ul style="list-style-type: none"> ■ VCNS.LoadBalancerEdgePool.Names ■ VCNS.LoadBalancerEdgePool.Names.moderate ■ VCNS.LoadBalancerEdgePool.Names.high ■ VCNS.LoadBalancerEdgePool.Names.low <p>NOTE In a multi-machine service, this custom property is supported only for VM components with Routed Virtual Wire. It is also not supported for pre-defined static IP addresses for VM components in the context of a multi-machine blueprint.</p>
VCNS.SecurityGroup.Names.name	<p>Specifies the vCloud Networking and Security security group or groups to which the virtual machine is assigned during provisioning. The value is a security group name or a list of names separated by commas. Names are case-sensitive.</p> <p>Appending a name allows you to create multiple versions of the property, which can be used separately or in combination. For example, the following properties can list security groups intended for general use, for the sales force, and for support:</p> <ul style="list-style-type: none"> ■ VCNS.SecurityGroup.Names ■ VCNS.SecurityGroup.Names.sales ■ VCNS.SecurityGroup.Names.support
VCNS.SecurityGroup.Names.blueprint_name	<p>When using vCloud Networking and Security, specifies the Edge Pool with which to associate the blueprint.</p>

Table 3-13. Custom Properties V Table (Continued)

This section lists vRealize Automation custom properties that begin with the letter V.

Property	Description
VCNS.SecurityTag.Names.name	<p>Specifies the vCloud Networking and Security security tag or tags to which the virtual machine is associated during provisioning. The value is a security tag name or a list of names separated by commas. Names are case-sensitive.</p> <p>Appending a name allows you to create multiple versions of the property, which can be used separately or in combination. For example, the following properties can list security tags intended for general use, for the sales force, and for support:</p> <ul style="list-style-type: none"> ■ VCNS.SecurityTag.Names ■ VCNS.SecurityTag.Names.sales ■ VCNS.SecurityTag.Names.support
VirtualMachine.Admin.UseGuestAgent	<p>If the guest agent is installed as a service on a template for cloning, set to True on the machine blueprint to enable the guest agent service on machines cloned from that template. When the machine is started, the guest agent service is started. Set to False to disable the guest agent. If set to False, the enhanced clone workflow will not use the guest agent for guest operating system tasks, reducing its functionality to VMwareCloneWorkflow. If not specified or set to anything other than False, the enhanced clone workflow will send work items to the guest agent.</p>
VirtualMachine.Admin.NameCompletion	<p>Specifies the domain name to include in the fully qualified domain name of the machine that the RDP or SSH files generate for the user interface options Connect Using RDP or Connect Using SSH option. For example, set the value to myCompany.com to generate the fully qualified domain name <i>my-machine-name.myCompany.com</i> in the RDP or SSH file.</p>
VirtualMachine.Admin.ConnectAddress	<p>Specifies the RDP connection address of the machine to which an RDP file is downloaded when the user interface option Connect Using RDP is used or attached to automatic emails. Do not use in a blueprint or property group unless you require the user to be prompted and you have not supplied a default value.</p>
VirtualMachine.Admin.ConnectAddress.Regex	<p>Used by a vRealize Automation administrator to define a regular expression to match an IP address for terminal connections, such as an RDP connection. If matched, the IP address is saved under the <code>VirtualMachine.Admin.ConnectAddress</code> custom property. Otherwise, the first available IP address is designated.</p> <p>For example, setting the property value to <code>10.10.0.</code> allows selection of an IP address from a <code>10.10.0.*</code> subnet that is assigned to the virtual machine. If the subnet has not been assigned, the property is ignored.</p> <p>This property is available for use with Openstack.</p>
VirtualMachine.Admin.ThinProvision	<p>Determines whether thin provisioning is used on ESX compute resources using local or iSCSI storage. Set to True to use thin provisioning. Set to False to use standard provisioning. This property is for virtual provisioning.</p>

Table 3-13. Custom Properties V Table (Continued)

This section lists vRealize Automation custom properties that begin with the letter V.

Property	Description
<code>VirtualMachine.Admin.CustomizeGuestOSDelay</code>	Specifies the time to wait after customization is complete and before starting the guest operating system customization. The value must be in HH:MM:SS format. If the value is not set, the default value is one minute (00:01:00). If you choose not to include this custom property, provisioning can fail if the virtual machine reboots before guest agent work items are completed.
<code>VirtualMachine.Admin.NetworkInterfaceType</code>	Indicates the network interface type. Use to modify the network settings of a newly provisioned virtual machine. The following options are available: <ul style="list-style-type: none"> ■ E1000 (default) ■ VirtIO ■ RTL8139 ■ RTL8139 VirtIO
<code>VirtualMachine.Admin.Name</code>	Specifies the generated machine name for vSphere, for example CodyVM01. When creating custom workflows or plug-ins for customizing a virtual machine name, set this property to match the name of the virtual machine. This is an internal input property for the agent to name the virtual machine. <p>NOTE This property is for vSphere only. The value specified in the blueprint has no effect on this property. This property is not intended to be used to prompt the user. Use the <code>HostName</code> property to prompt the user. If the property is set at runtime, the container name that is created in the hypervisor might not match the item record name.</p>
<code>VirtualMachine.Admin.UUID</code>	Specifies the UUID of the machine. The value is recorded by the guest agent when the machine is created, then it becomes read-only. The value in the blueprint or property group has no effect on this property.
<code>VirtualMachine.Admin.AgentID</code>	Specifies the UUID of the guest agent. The value is recorded by the guest agent when the machine is created, then it becomes read-only. The value in the blueprint or property group has no effect on this property.
<code>VirtualMachine.Admin.Owner</code>	Specifies the user name of the machine owner.
<code>VirtualMachine.Admin.Approver</code>	Specifies the user name of the group manager who approved the machine request.
<code>VirtualMachine.Admin.Description</code>	Specifies the description of the machine as entered or modified by its owner or an administrator.
<code>VirtualMachine.Admin.EncryptPasswords</code>	If set to True, specifies that the administrator passwords are encrypted.
<code>VirtualMachine.Admin.AdministratorEmail</code>	Specifies the manager email addresses or Active Directory accounts for the business group of the provisioning blueprint. Multiple email addresses are separated by a comma, for example <code>JoeAdmin@VMware.com,WeiLeeMgr@VMware.com</code> .
<code>VirtualMachine.Admin.TotalDiskUsage</code>	Specifies the total disk space that the machine uses, including all disks as specified by the <code>VirtualMachine.DiskN.Size</code> properties and the swap file as specified by the <code>VMware.Memory.Reservation</code> property.

Table 3-13. Custom Properties V Table (Continued)

This section lists vRealize Automation custom properties that begin with the letter V.

Property	Description
<code>VirtualMachine.Admin.Hostname</code>	<p>Informs the administrator which host is used for provisioning the machine on the endpoint. The specified value is implemented on the machine and is populated during data collection. For example, if the compute resource of a machine is changed, a proxy agent updates the value of the machine's <code>VirtualMachine.Admin.Hostname</code> property.</p> <p>NOTE This is an internal output property from the agent that is populated during the data collection process and identifies the host on which a machine resides.</p>
<code>VirtualMachine.Admin.ClusterName</code>	<p>Informs the administrator which cluster contains the compute resource for the machine to use.</p> <p>NOTE This is an internal output property from the agent that is populated during the data collection process and identifies the cluster in which a machine resides.</p>
<code>VirtualMachine.Admin.ApplicationID</code>	List the application IDs that can be assigned to a machine.
<code>VirtualMachine.Admin.AddOwnerToAdmins</code>	<p>Set to True (default) to add the machine's owner, as specified by the <code>VirtualMachine.Admin.Owner</code> property, to the local administrators group on the machine.</p> <p>This property is not available for provisioning by cloning.</p>
<code>VirtualMachine.Admin.AllowLogin</code>	Set to True (default) to add the machine owner to the local remote desktop users group, as specified by the <code>VirtualMachine.Admin.Owner</code> property.
<code>VirtualMachine.Admin.DiskInterfaceType</code>	<p>Indicates the type of disk drivers. The following disk drivers are supported:</p> <ul style="list-style-type: none"> ■ IDE (default) ■ VirtIO <p>This property is for virtual provisioning.</p>
<code>VirtualMachine.Admin.ForceHost</code>	<p>Specifies the name of the ESX host. The property is only honored if <code>VirtualMachine.Admin.HostSelectionPolicy</code> is set to EXACT_MATCH.</p> <p>NOTE This property is for vSphere only.</p> <p>When provisioning against a vSphere cluster, you can use the <code>VirtualMachine.Admin.ForceHost</code> property to specify the host on which a machine is to be provisioned. This property is used only if DRS is not set to automatic for the cluster. If the cluster has DRS enabled and is set to Automatic, vSphere relocates the provisioned machine when the machine is restarted.</p>
<code>VirtualMachine.Admin.HostSelectionPolicy</code>	<p>Optionally set to EXACT_MATCH to require the machine to be placed on the host specified by the <code>VirtualMachine.Admin.ForceHost</code> property. If the host is unavailable, the request results in a failure. If a host is not specified, the next best available host is selected. If set to EXACT_MATCH, an error occurs if the specified host does not have enough memory or is in maintenance mode.</p> <p>NOTE This property applies to vSphere only.</p>

Table 3-13. Custom Properties V Table (Continued)

This section lists vRealize Automation custom properties that begin with the letter V.

Property	Description
<code>VirtualMachine.Admin.ExportProperties</code>	Optionally set to True when using an enhanced clone workflow to specify that the guest agent export the virtual machine's custom properties to the following files in the guest OS: <ul style="list-style-type: none"> ■ <code>C:\VRMGuestAgent\site\workitem.xml</code> ■ <code>C:\VRMGuestAgent\site\workitem.ini</code> Guest scripts and utilities can then consume these files.
<code>VirtualMachine.Agent.CopyToDisk</code>	Set to True (default) to copy the guest agent executable file to <code>%System-Drive%\VRM\Build\Bin</code> on the machine's disk.
<code>VirtualMachine.Agent.GuiRunOnce</code>	Set to True to include guest agent execution in the <code>SysPrep.inf</code> <code>runonce</code> section. Set to False for the Linux agent to stop the provisioning workflow.
<code>VirtualMachine.Agent.Reboot</code>	Set to True (default) to specify that the guest agent restarts the machine following installation of the guest operating system.
<code>VirtualMachine.CDRom.Attach</code>	Set to False to provision the machine without a CD-ROM device. The default is True.
<code>VirtualMachine.CPU.Count</code>	Specifies the number of CPUs, for example 2, allocated to a machine. The default is the value specified by the CPU setting on the blueprint. NOTE This custom property value is overridden by the CPU value on the blueprint when the machine is first provisioned.
<code>VirtualMachine.Customize.WaitComplete</code>	Set to True to prevent the provisioning workflow from sending work items to the guest agent until all customizations have been completed.
<code>VirtualMachine.DiskN.Letter</code>	Specifies the drive letter or mount point of a machine's disk N. The default is C. For example, to specify the letter D for Disk 1, define the custom property as <code>VirtualMachine.Disk1.Letter</code> and enter the value D. Disk numbering must be sequential. When used in conjunction with a guest agent, this value specifies the drive letter or mount point under which an additional disk N is mounted by the guest agent in the guest operating system.
<code>VirtualMachine.DiskN.Size</code>	Defines the size in GB of disk N. For example, to give a size of 150 GB to a disk G, define the custom property <code>VirtualMachine.Disk0.Size</code> and enter a value of 150. Disk numbering must be sequential. By default a machine has one disk referred to by <code>VirtualMachine.Disk0.Size</code> , where size is specified by the storage value on the blueprint from which the machine is provisioned. The storage value on the blueprint user interface overwrites the value in the <code>VirtualMachine.Disk0.Size</code> property. The <code>VirtualMachine.Disk0.Size</code> property is not available as a custom property because of its relationship with the storage option on the blueprint. More disks can be added by specifying <code>VirtualMachine.Disk1.Size</code> , <code>VirtualMachine.Disk2.Size</code> and so on. <code>VirtualMachine.Admin.TotalDiskUsage</code> always represents the total of the <code>.DiskN.Size</code> properties plus the <code>VMware.Memory.Reservation</code> size allocation.

Table 3-13. Custom Properties V Table (Continued)

This section lists vRealize Automation custom properties that begin with the letter V.

Property	Description
VirtualMachine.DiskN.IsFixed	<p>Disables the editing of a specific disk when reconfiguring a machine. Set to True to disable display of the edit capacity option for a specific volume. The True value is case-sensitive. The <i>N</i> value is the 0-based index of the disk.</p> <p>Alternatively, you can set the VirtualMachine.DiskN.IsFixed custom property to True in the VirtualMachineProperties table in the database or use the Repository API to specify a URI value such as <code>.../Repository/Data/ManagementModelEntities.svc/VirtualMachines(guid'60D93A8A-F541-4CE0-A6C6-78973AC0F1D2')/VirtualMachineProperties</code>.</p>
VirtualMachine.DiskN.Label	<p>Specifies the label for a machine's disk. The disk label maximum is 32 characters. Disk numbering must be sequential. When used in conjunction with a guest agent, specifies the label of a machine's disk <i>N</i> inside the guest operating system.</p>
VirtualMachine.DiskN.Active	<p>Set to True (default) to specify that the machine's disk <i>N</i> is active. Set to False to specify that the machine's disk <i>N</i> is not active.</p>
VirtualMachine.DiskN.FS	<p>Specifies the file system of the machine's disk <i>N</i>. The options are NTFS (default), FAT and FAT32.</p>
VirtualMachine.DiskN.Percent	<p>Specifies the percentage of the disk <i>N</i> to be formatted by a guest agent for the machine's use. That machine cannot use the remaining portion of the disk.</p>
VirtualMachine.DiskN.StorageReservationPolicy	<p>Specifies the storage reservation policy to use to find storage for disk <i>N</i>. Also assigns the named storage reservation policy to a volume. To use this property, substitute the volume number for <i>N</i> in the property name and specify a storage reservation policy name as the value. This property is equivalent to the storage reservation policy name specified on the blueprint. Disk numbering must be sequential. This property is valid for all Virtual and vCloud reservations. This property is not valid for Physical or Cloud reservations, other than vCloud reservations.</p>
VirtualMachine.DiskN.StorageReservationPolicyMode	<p>Allocates disk <i>N</i> to the best available storage reservation policy.</p>
VirtualMachine.DiskN.Storage	<p>Specifies the datastore on which to place the machine disk <i>N</i>, for example DATASTORE01. This property is also used to add a single datastore to a linked clone blueprint. <i>N</i> is the index (starting at 0) of the volume to assign. Enter the name of the datastore to assign to the volume. This is the datastore name as it appears in the Storage Path on the Edit Compute Resource page. Disk numbering must be sequential.</p>

Table 3-13. Custom Properties V Table (Continued)

This section lists vRealize Automation custom properties that begin with the letter V.

Property	Description
VirtualMachine.DiskN.VMwareType	<p>Specifies the VMware disk mode of the machine's diskN. This property applies to vSphere. The following options are available:</p> <ul style="list-style-type: none"> ■ persistent ■ independent_persistent ■ independent_nonpersistent <p>If you set this custom property to independent_persistent or independent_nonpersistent, users must power off their machine before using vRealize Automation to take a snapshot.</p> <p>NOTE This property applies to vSphere only.</p> <p>For details, see VirtualDeviceDeviceBackingOption data object help in <i>VMware vSphere Web Services SDK Documentation</i>.</p>
VirtualMachine.Host.TpmEnabled	<p>Limits virtual machine placement to hosts that have a Trust Protection Module (TPM) device installed and recognized by ESX and vSphere. The default value is False.</p> <p>All hosts in a cluster must have a Trust Protection Module device installed. If no acceptable hosts or clusters are found, the machine cannot be provisioned until this property is removed.</p>
VirtualMachine.Rdp.File	<p>Specifies the RDP file that contains settings to be used when opening an RDP link to the machine. Can be used together with, or as an alternative to, VirtualMachine.Rdp.SettingN. The file must be located in the vRealize Automation server installation directory, for example %SystemDrive%\Program Files x86\VMware\VCAC\Server\Rdp\console.rdp. You must create the Rdp directory.</p>
VirtualMachine.Memory.Size	<p>Specifies the size of the machine's memory in MB, such as 1024. The default is the value specified by the memory setting on the blueprint.</p> <p>NOTE This custom property setting is overridden by the memory setting on the blueprint when the machine is first provisioned.</p>
VirtualMachine.EPI.Type	<p>Specifies the type of external provisioning infrastructure. Set to BMC for BMC BladeLogic integration. Set to CitrixProvisioning for Citrix provisioning server integration.</p>
VirtualMachine.NetworkN.Address	<p>Specifies the IP address of network device N in a machine provisioned with a static IP address.</p> <p>VirtualMachine.NetworkN custom properties are designed to be specific to individual blueprints and machines. When a machine is requested, network and IP address allocation is performed before the machine is assigned to a reservation. Because blueprints are not guaranteed to be allocated to a specific reservation, do not use this property on a reservation.</p>

Table 3-13. Custom Properties V Table (Continued)

This section lists vRealize Automation custom properties that begin with the letter V.

Property	Description
VirtualMachine.NetworkN.AddressM	<p>Defines additional <i>M</i> IP address allocated for an Openstack instance for network <i>N</i>, excluding the IP address set specified by the <code>VirtualMachine.NetworkN.Address</code> property. Additional addresses are displayed on the Network tab in the Additional Addresses column.</p> <p>This property is used by Openstack machine state data collection.</p> <p>While this property is only data-collected by the OpenStack endpoint, it is not specific to OpenStack and can be used for lifecycle extensibility by other endpoint types.</p>
VirtualMachine.NetworkN.MacAddressType	<p>Indicates whether the MAC address of network device <i>N</i> is generated or user-defined (static). This property is available for cloning.</p> <p>The default value is generated. If the value is static, you must also use <code>VirtualMachine.NetworkN.MacAddress</code> to specify the MAC address.</p> <p><code>VirtualMachine.NetworkN</code> custom properties are designed to be specific to individual blueprints and machines. When a machine is requested, network and IP address allocation is performed before the machine is assigned to a reservation. Because blueprints are not guaranteed to be allocated to a specific reservation, do not use this property on a reservation.</p>
VirtualMachine.NetworkN.MacAddress	<p>Specifies the MAC address of a network device <i>N</i>. This property is available for cloning.</p> <p>If the value of <code>VirtualMachine.NetworkN.MacAddressType</code> is generated, this property contains the generated address.</p> <p>If the value of <code>VirtualMachine.Network.N.MacAddressType</code> is static, this property specifies the MAC address. For virtual machines provisioned on ESX server hosts, the address must be in the range specified by VMware. For details, see vSphere documentation.</p> <p><code>VirtualMachine.NetworkN</code> custom properties are designed to be specific to individual blueprints and machines. When a machine is requested, network and IP address allocation is performed before the machine is assigned to a reservation. Because blueprints are not guaranteed to be allocated to a specific reservation, do not use this property on a reservation.</p>

Table 3-13. Custom Properties V Table (Continued)

This section lists vRealize Automation custom properties that begin with the letter V.

Property	Description
VirtualMachine.NetworkN.Name	<p>Specifies the name of the network to connect to, for example the network device <i>N</i> to which a machine is attached.</p> <p>By default, a network is assigned from the network paths available on the reservation on which the machine is provisioned.</p> <p>You can ensure that a network device is connected to a specific network by setting the value of this property to the name of a network on an available reservation.</p> <p>VirtualMachine.NetworkN custom properties are designed to be specific to individual blueprints and machines. When a machine is requested, network and IP address allocation is performed before the machine is assigned to a reservation. Because blueprints are not guaranteed to be allocated to a specific reservation, do not use this property on a reservation.</p> <p>You can add this property to a vApp (vCloud) blueprint or a vApp (vCloud) Component blueprint.</p>
VirtualMachine.NetworkN.PortID	<p>Specifies the port ID to use for network device <i>N</i> when using a dvPort group with a vSphere distributed switch.</p> <p>VirtualMachine.NetworkN custom properties are designed to be specific to individual blueprints and machines. When a machine is requested, network and IP address allocation is performed before the machine is assigned to a reservation. Because blueprints are not guaranteed to be allocated to a specific reservation, do not use this property on a reservation.</p>

Table 3-13. Custom Properties V Table (Continued)

This section lists vRealize Automation custom properties that begin with the letter V.

Property	Description
VirtualMachine.NetworkN.ProfileName	<p>Specifies the name of a network profile from which to assign a static IP address to network device <i>N</i> or from which to obtain the range of static IP addresses that can be assigned to network device <i>N</i> of a cloned machine, where <i>N</i>=0 for the first device, 1 for the second, and so on.</p> <p>If a network profile is specified in the network path in the reservation on which the machine is provisioned, a static IP address is assigned from that network profile. You can ensure that a static IP address is assigned from a specific profile by setting the value of this property to the name of a network profile.</p> <p>Note that changing this property value after the network is assigned has no effect on the expected IP address values for the designated machines.</p> <p>With WIM-based provisioning for virtual machines, you can use this property to specify a network profile and network interface or you can use the Network section of the Virtual Reservation page. You can also assign the network interface to a virtual network using the VirtualMachine.NetworkN.Name custom property.</p> <p>With WIM-based provisioning for physical machines, you must specify the network profile with the VirtualMachine.NetworkN.ProfileName custom property. You must also specify a value with the VirtualMachine.NetworkN.Name custom property, although vRealize Automation does not use the value.</p> <p>The following attributes of the network profile are available to enable static IP assignment in a cloning blueprint:</p> <ul style="list-style-type: none"> ■ VirtualMachine.NetworkN.SubnetMask ■ VirtualMachine.NetworkN.Gateway ■ VirtualMachine.NetworkN.PrimaryDns ■ VirtualMachine.NetworkN.SecondaryDns ■ VirtualMachine.NetworkN.PrimaryWins ■ VirtualMachine.NetworkN.SecondaryWins ■ VirtualMachine.NetworkN.DnsSuffix ■ VirtualMachine.NetworkN.DnsSearchSuffixes <p>VirtualMachine.NetworkN custom properties are designed to be specific to individual blueprints and machines. When a machine is requested, network and IP address allocation is performed before the machine is assigned to a reservation. Because blueprints are not guaranteed to be allocated to a specific reservation, do not use this property on a reservation.</p>
<ul style="list-style-type: none"> ■ VirtualMachine.NetworkN.SubnetMask ■ VirtualMachine.NetworkN.Gateway ■ VirtualMachine.NetworkN.PrimaryDns ■ VirtualMachine.NetworkN.SecondaryDns ■ VirtualMachine.NetworkN.PrimaryWins ■ VirtualMachine.NetworkN.SecondaryWins ■ VirtualMachine.NetworkN.DnsSuffix ■ VirtualMachine.NetworkN.DnsSearchSuffixes 	<p>Configures attributes of the network profile specified in VirtualMachine.NetworkN.ProfileName.</p> <p>VirtualMachine.NetworkN custom properties are designed to be specific to individual blueprints and machines. When a machine is requested, network and IP address allocation is performed before the machine is assigned to a reservation. Because blueprints are not guaranteed to be allocated to a specific reservation, do not use this property on a reservation.</p>

Table 3-13. Custom Properties V Table (Continued)

This section lists vRealize Automation custom properties that begin with the letter V.

Property	Description
VirtualMachine.Rdp.SettingN	<p>Configures specific RDP settings. <i>N</i> is a unique number used to distinguish one RDP setting from another. For example, to specify the Authentication Level so that no authentication requirement is specified, define the custom property <code>VirtualMachine.Rdp.Setting1</code> and set the value to <code>authentication level:i:3</code>. Use to open an RDP link to specify settings.</p> <p>For a list of available settings and correct syntax, see the Microsoft Windows RDP documentation.</p>
VirtualMachine.Request.Layout	<p>Specifies the property layout to be used in the virtual machine request page. The value must match the name of the layout to be used.</p>
VirtualMachine.Reconfigure.DisableHotCpu	<p>Set to True to specify that the reconfigure machine action restarts the specified machine. By default, the reconfigure machine action does not restart the machine.</p> <p>Performing a hot add of CPU, or memory, or storage causes the reconfigure machine action to fail and to not restart the machine unless the Hot Add setting is enabled in vSphere for the machine or template. You can add <code>VirtualMachine.Reconfigure.DisableHotCpu=true</code> to a vRealize Automation blueprint to force the machine to restart regardless of the vSphere Hot Add setting.</p>
VirtualMachine.SoftwareN.Name	<p>Specifies the descriptive name of a software application or script to install or run during provisioning. This is an optional and information-only property. It serves no real function for the enhanced clone workflow or the guest agent but it is useful for a custom software selection in a user interface or for software usage reporting.</p>
VirtualMachine.SoftwareN.ScriptPath	<p>Specifies the full path to an application's install script. The path must be a valid absolute path as seen by the guest operating system and must include the name of the script file name.</p> <p>You can pass custom property values as parameters to the script by inserting <code>{CustomPropertyName}</code> in the path string. For example, if you have a custom property named <code>ActivationKey</code> whose value is 1234, the script path is <code>D:\InstallApp.bat -key {ActivationKey}</code>. The guest agent runs the command <code>D:\InstallApp.bat -key 1234</code>. Your script file can then be programmed to accept and use this value.</p> <p>You can also pass custom property values as parameters to the script by inserting <code>{YourCustomProperty}</code> in the path string. For example, entering the value <code>\\vra-scripts.mycompany.com\scripts\changeIP.bat</code> runs the <code>changeIP.bat</code> script from a shared location, but entering the value <code>\\vra-scripts.mycompany.com\scripts\changeIP.bat {VirtualMachine.Network0.Address}</code> runs the <code>changeIP</code> script but also passes the value of the <code>VirtualMachine.Network0.Address</code> property to the script as a parameter.</p> <p>Insert <code>{Owner}</code> to pass the machine owner name to the script.</p>

Table 3-13. Custom Properties V Table (Continued)

This section lists vRealize Automation custom properties that begin with the letter V.

Property	Description
VirtualMachine.ScriptPath.Decrypt	<p>Allows vRealize Automation to obtain an encrypted string that is passed as a properly formatted VirtualMachine.SoftwareN.ScriptPath custom property statement to the gagent command line.</p> <p>You can provide an encrypted string, such as your password, as a custom property in a command line argument. This allows you to store encrypted information that the guest agent can decrypt and understand as a valid command line argument. For example, the VirtualMachine.Software0.ScriptPath = c:\dosomething.bat <i>password</i> custom property string is not secure as it contains an actual password.</p> <p>To encrypt the password, you can create a vRealize Automation custom property, for example MyPassword = <i>password</i>, and enable encryption by selecting the available checkbox. The guest agent decrypts the [MyPassword] entry to the value in the custom property MyPassword and executes the script as c:\dosomething.bat <i>password</i>.</p> <ul style="list-style-type: none"> ■ Create custom property MyPassword = <i>password</i> where <i>password</i> is the value of your actual password. Enable encryption by selecting the available checkbox. ■ Set custom property VirtualMachine.ScriptPath.Decrypt as VirtualMachine.ScriptPath.Decrypt = true. ■ Set custom property VirtualMachine.Software0.ScriptPath as VirtualMachine.Software0.ScriptPath = c:\dosomething.bat [MyPassword]. <p>If you set VirtualMachine.ScriptPath.Decrypt to false, or don't create the VirtualMachine.ScriptPath.Decrypt custom property, then the string inside the square brackets ([and]) is not decrypted.</p>
VirtualMachine.SoftwareN.ISOName	<p>Specifies the path and file name of the ISO file relative to the data store root. The format is <i>/folder_name/subfolder_name/file_name.iso</i>. If a value is not specified, the ISO is not mounted.</p>
VirtualMachine.SoftwareN.ISOLocation	<p>Specifies the storage path that contains the ISO image file to be used by the application or script. Format the path as it appears on the host reservation, for example <i>netapp-1:it_nfs_1</i>. If a value is not specified, the ISO is not mounted.</p>
VirtualMachine.SoftwareN.PostInstallDelay	<p>Specifies a time delay value, in TimeSpan string format, to wait before sending the next InstallSoftware work item. If a value is not specified, there is no time delay between InstallSoftware work items.</p>
VirtualMachine.SoftwareN.FailsBuild	<p>Optionally set to True to specify that a failure when executing the work item causes the virtual machine build to fail.</p>

Table 3-13. Custom Properties V Table (Continued)

This section lists vRealize Automation custom properties that begin with the letter V.

Property	Description
<code>VirtualMachine.SoftwareN.NoQuotes</code>	<p>Specifies that expanding the script path does not enclose values that contain spaces in double quotes.</p> <p>Example 1: When using the following statement in a <code>testprop = test param</code> format:</p> <pre>VirtualMachine.Software0.ScriptPath = C:\path0 {testprop}, VirtualMachine.Software0.NoQuotes = false</pre> <p><code>Vrm.Software.Command</code> expands to <code>C:\path0 "test param"</code>.</p> <p>Example 2: When using the following statement in a <code>testprop = test param</code> format:</p> <pre>VirtualMachine.Software0.ScriptPath = C:\path0 {testprop}, VirtualMachine.Software0.NoQuotes = true</pre> <p><code>Vrm.Software.Command</code> expands to <code>C:\path0 test param</code>.</p>
<code>VirtualMachine.Storage.Name</code>	Identifies the storage path on which the machine resides. The default is the value specified in the reservation that was used to provision the machine.
<code>VirtualMachine.Storage.AllocationType</code>	Stores collected groups to a single datastore. A distributed environment stores disks round-robin style.
<code>VirtualMachine.Storage.ReserveMemory</code>	<p>Set to True to manage vSwap storage allocation to ensure availability and set allocation in the reservation. vSwap allocation is considered when you create or reconfigure a virtual machine. vSwap allocation checking is only available for vCenter Server endpoints. For multi-machine blueprints, add the property to the component blueprints.</p> <p>NOTE If you do not specify the <code>VirtualMachine.Storage.ReserveMemory</code> custom property when you create or provision the machine from vRealize Automation, swap space availability is not ensured. If you add the property for an already provisioned machine, and the allocated reservation is full, the storage allocated in the reservation might exceed the actual allocated storage.</p>
<code>VirtualMachine.VDI.Type</code>	Specifies the type of virtual desktop infrastructure. For XenDesktop provisioning, set to XenDesktop.
<code>VMware.AttributeN.Name</code>	Specifies the name of an attribute in vRealize Orchestrator. For example, it specifies the value of the attribute used in the <code>VMware.AttributeN.Name</code> property. Replace the letter <i>N</i> with a number, starting with 0 and increasing for each attribute to set.
<code>VMware.AttributeN.Value</code>	Specifies the value of the attribute used in the <code>VMware.AttributeN.Name</code> property. Replace the letter <i>N</i> with a number, starting with 0 and increasing for each attribute to set.
<code>VMware.Endpoint.Openstack.IdentityProvider.Version</code>	For 6.2.4 and 6.2.5, specifies the version of Openstack Identity provider (Keystone) to use when authenticating an Openstack endpoint. Configure a value of 3 to authenticate with Keystone version 3 OpenStack Identity Provider. If you use any other value, or do not use this custom property, authentication defaults to Keystone version 2.

Table 3-13. Custom Properties V Table (Continued)

This section lists vRealize Automation custom properties that begin with the letter V.	Description
VMware.Endpoint.Openstack.Release	Specifies the OpenStack release, for example Havana or Icehouse, when creating an OpenStack endpoint. Required for 6.2.1, 6.2.2, and 6.2.3 OpenStack provisioning. Deprecated as of 6.2.4.
VMware.Hardware.Version	Specifies the VM hardware version to be used for vSphere settings. Supported values are currently vmx-04, vmx-07, vmx-08, vmx-09 and vmx-10. This property is applicable for VM Create and VM Update workflows and is available only for basic workflow blueprints.
VMware.VirtualCenter.OperatingSystem	<p>Specifies the vCenter Server guest operating system version (<code>VirtualMachineGuestOsIdentifier</code>) with which vCenter Server creates the machine. This operating system version must match the operating system version to be installed on the provisioned machine.</p> <p>Administrators can create property groups using one of several property sets, for example, <code>VMware[OS_Version]Properties</code>, that are predefined to include the correct <code>VMware.VirtualCenter.OperatingSystem</code> values. This property is for virtual provisioning.</p> <p>When this property has a non-Windows value, the Connect Using RDP user interface option is disabled. The property can be used in a virtual, cloud or physical blueprint.</p> <p>For related information, see the enumeration type <code>VirtualMachineGuestOsIdentifier</code> in vSphere API/SDK Documentation. For a list of currently accepted values, see the VMware vCenter Server™ documentation.</p>
VMware.SCSI.Type	<p>For vApps and vSphere, specifies the SCSI machine type using one of the following case-sensitive values:</p> <ul style="list-style-type: none"> ■ buslogic Use BusLogic emulation for the virtual disk. ■ lsilogic Use LSILogic emulation for the virtual disk (default). ■ lsilogicsas Use LSILogic SAS 1068 emulation for the virtual disk. ■ pvscsi Use para-virtualization emulation for the virtual disk. ■ none Use if a SCSI controller does not exist for this machine.
VMware.SCSI.Sharing	Specifies the sharing mode of the machine's VMware SCSI bus. Possible values are based on the <code>VirtualSCSISharingEnum</code> value and include <code>noSharing</code> , <code>physicalSharing</code> , and <code>virtualSharing</code> .
VMware.Memory.Reservation	Specifies the size of the machine's swap file, for example 1024.

Table 3-13. Custom Properties V Table (Continued)

This section lists vRealize Automation custom properties that begin with the letter V.

Property	Description
VMware.Network.Type	<p>Specifies a network adapter type. The following adapter type values are available:</p> <ul style="list-style-type: none"> ■ Flexible (default) ■ VirtualPCNet32 (not compatible with vSphere). ■ E1000 or VirtualE1000 ■ VMXNET or VirtualVMXNET ■ VMXNET2 ■ VMXNET3 <p>Set to E1000 when provisioning Windows 32-bit virtual machines on ESX server hosts to ensure that machines are created with the correct network adapter. This property is not used for physical provisioning.</p>
VMware.VCenterOrchestrator.EndpointName	<p>Overrides a specified endpoint setting or specifies that a particular endpoint be used during the vRealize Automation IaaS provisioning process. The value of this property can be set to an applicable vRealize Orchestrator endpoint, such as external VRO, available in the environment.</p>
VMware.VirtualCenter.Folder	<p>Specifies the name of the inventory folder in the datacenter in which to put the virtual machine. The default is VRM, which is also the vSphere folder in which vRealize Automation places provisioned machines if the property is not used. This value can be a path with multiple folders, for example <code>production\email servers</code>. A proxy agent creates the specified folder in vSphere if the folder does not exist. Folder names are case sensitive. This property is available for virtual provisioning.</p>
VDI.Server.Website	<p>Specifies the server name of the Citrix Web interface site to use in connecting to the machine. If the value of <code>VDI.Server.Name</code> is a XenDesktop farm, this property must have an appropriate value or the machine owner cannot connect to the machine using XenDesktop. If this property is not specified, the <code>VDI.Server.Name</code> property determines the desktop delivery controller to connect to, which must be the name of a server that hosts a desktop delivery controller.</p> <p>NOTE If the Citrix Web Interface (WI) has been replaced with StoreFront (SF), you can use this property instead of <code>VDI.Server.Name</code> to connect to the XenDesktop server. An example value is <code>VDI.Server.Website=sqa-xddc-7.sqa.local/Citrix/StoreWeb</code>. See <code>VDI.Server.Name</code> for more information.</p>

Table 3-13. Custom Properties V Table (Continued)

This section lists vRealize Automation custom properties that begin with the letter V.

Property	Description
VDI.Server.Name	<p>Specifies the server name, which hosts the desktop delivery controller, to register with, or the name of a XenDesktop farm that contains desktop delivery controllers with which to register.</p> <p>If the value is a XenDesktop farm name, the VDI.Server.Website property value must be the URL of an appropriate Citrix web interface site to use in connecting to the machine.</p> <p>If the value is a server name, and at least one general XenDesktop VDI agent was installed without specifying a desktop delivery controller server, this value directs the request to the desired server. If the value is a server name, and only dedicated XenDesktop VDI agents for specific DDC servers were installed, this value must exactly match the server name configured for a dedicated agent.</p> <p>NOTE For more information about how to make StoreFront the default page in IIS, see Citrix documentation. See also VDI.Server.Website.</p> <p>NOTE Changes in the Citrix web interface protocol have impacted how the VDI.Server.Name default value is recognized. The value of the VDI.Server.Name property is used as the default connection string to open the Citrix web interface when users connect to a virtual desktop. It is always the DNS/IP of the XD server. If that value does not connect to the Citrix interface, you are unable to access your VMs. However, you can use the VDI.Server.Website custom property when the Citrix web interface is hosted on a server other than the XenDesktop server. When this property is present on the VM, it is used instead of VDI.Server.Name.</p>
VDI.Server.Group	<p>For XenDesktop 5, specifies the name of the XenDesktop group to add machines to and the name of the catalog to which the group belongs, in the <i>group_name;catalog_name</i> format.</p> <p>For XenDesktop 4, specifies the name of the XenDesktop group to which machines are to be added. XenDesktop 4 preassigned groups are supported.</p>
VDI.ActiveDirectory.Interval	<p>Specifies an optional interval value in time span format for virtual desktop infrastructure machine Active Directory registration check. The default value is 00:00:15 (15 seconds).</p>
VDI.ActiveDirectory.Timeout	<p>Specifies an optional timeout value to wait before retrying Active Directory registration. The default value is 00:00:15 (30 minutes).</p>
VDI.ActiveDirectory.Delay	<p>Specifies an optional delay time value in time span format between successfully adding a machine to Active Directory and initiation of XenDesktop registration. The default value is 00:00:05 (5 seconds).</p>

Table 3-13. Custom Properties V Table (Continued)

This section lists vRealize Automation custom properties that begin with the letter V.

Property	Description
VRM.DataCenter.Policy	<p>Specifies whether provisioning must use a compute resource associated with a particular location, or if any location is suitable. To enable this feature you must add data center to a location file. Associate each compute resource with a location.</p> <p>Set to Exact to provision a requested machine on a compute resource associated with the location specified on the blueprint. If a compute resource with sufficient capacity and associated with that location is not available, then provisioning fails.</p> <p>Set to NonExact (default) to provision a requested machine on a compute resource with sufficient capacity and associated with the location specified on the blueprint. If that compute resource is not available, then use the next available compute resource with sufficient capacity without regard to location.</p>
VRM.Software.IdNNNN This row is specific to BMC BladeLogic.	<p>Specifies a software job or policy to be applied to all machines provisioned from the blueprint. Set the value to <code>job_type=job_path</code>, where <code>job_type</code> is the numeral that represents the BMC BladeLogic job type and <code>job_path</code> is the location of the job in BMC BladeLogic, for example <code>4=/Utility/putty</code>. NNNN is a number from 1000 to 1999.</p> <ul style="list-style-type: none"> 1 – AuditJob 2 – BatchJob 3 – ComplianceJob 4 – DeployJob 5 – FileDeployJob 6 – NSHScriptJob 7 – PatchAnalysisJob 8 – SnapshotJob
VRM.Software.IdNNNN This row is specific to HP Server Automation.	<p>(Optional) Specifies an HP Server Automation policy to be applied to all machines provisioned from the blueprint. NNNN is a number from 1000 to 1999.</p>

Custom Properties X Table

This section lists vRealize Automation custom properties that begin with the letter X.

Table 3-14. Custom Properties X Table

Property	Description
Xen.Platform.Viridian	<p>For virtual provisioning, set to False when you provision Windows virtual machines on a XenServer host or pool. The default is True. This property is not used in physical provisioning.</p>

Using the Property Dictionary

You can use the property dictionary to define new custom properties.

You can customize how properties appear in the user interface.

- Associate a property name with a user control, such as a check box or drop-down menu.
- Specify constraints such as minimum and maximum values or validation against a regular expression.
- Provide descriptive display names for properties or add label text.
- Group sets of property controls together and specify the order in which they appear.

Use the following workflow when you define and use a custom property definition.

- 1 Add a new property and associate it with a property control type.
- 2 Add relevant property attributes to the property.

Attributes contain additional information such as minimum and maximum values or help text to display with the property. Available attribute types vary depending on the control type.

- 3 Add the properties to a blueprint either individually or by using a property group.

A tenant administrator or business group manager can create or edit blueprints.

- 4 (Optional) Add a control layout to specify where the properties appear on the machine confirmation page.

New custom properties are typically used by custom workflows to specify how to run a command based. You can use vRealize Automation Designer to define these custom workflows. For more information about working with custom workflows, see *Extension Development*.

This chapter includes the following topics:

- [“Adding Property Definitions,”](#) on page 78
- [“Adding Property Attributes,”](#) on page 79
- [“Example of Creating a Relationship Between Two Properties and Adding Them to a Blueprint,”](#) on page 84
- [“Creating Property Control Layouts,”](#) on page 87

Adding Property Definitions

A property definition contains an associated property control type, and optionally a property attribute, which define its behavior.

Property control types are available as you create new properties. After you create a property, you can add attributes, based on its control type. For example, you can assign a `DropDown` control type to a `ValueExpression` attribute, which when selected, opens a text box in which the user can type a value. You can also create placement layouts to control where the properties appear when presented to users.

Table 4-1. Property Definition Control Types and Attributes

Control Type	Available Attributes	Description
CheckBox	HelpText OrderIndex	Specifies a check box for specifying true or false values.
DateTimeEdit	HelpText OrderIndex MinValue	Specifies a calendar and time control that enables users to specify a date and time.
DropDown	HelpText OrderIndex Relationship ValueExpression ValueList	Specifies a drop-down text box. A user can enter any text or select from the list of options.
DropDownList	HelpText OrderIndex Relationship ValueExpression ValueList	Specifies a drop list of options. A user must select a value from the list.
Integer	HelpText OrderIndex Interval MinValue MaxValue	Specifies a numeric box. If attributes are also assigned, a user can enter an integer value between a defined minimum and maximum value or step up and down by a defined interval value.
Label	HelpText OrderIndex	Displays a read-only text label.
Link	OrderIndex	Displays a link with the property display name as the link text and the property value as the URL.

Add a Property Definition

You can define new property definitions to allow for additional customization.

After you add a property definition and select a definition control type for that property, you can add attributes to configure property controls. A tenant administrator or business group manager can add properties to blueprints or build profiles. You can also create a property control layout to organize display of the new properties.

Prerequisites

Log in to the vRealize Automation console as a **fabric administrator**.

Procedure

- 1 Select **Infrastructure > Blueprints > Property Dictionary**.

- 2 Click **New Property Definition**.
- 3 Enter the new property name in the **Name** text box.
This name must exactly match the property name that is used in the blueprint or build profile.
- 4 Enter the name that will appear in the user interface in the **Display Name** text box.
- 5 (Optional) Enter a property description in the **Description** text box.
This text is visible to the property dictionary administrator but is not visible to users.
- 6 Select a definition control type to associate with this property from the **Control Type** drop-down menu.

Option	Description
CheckBox	Associate a check box with the new property.
DateTimeEdit	Adds a date and time control type that adheres to a <i>YYYY-MM-DD</i> or <i>MM/DD/YYYY</i> format and a time in <i>HH:MM</i> format, 24-hour clock or followed by AM or PM. You can further refine the allowed value by adding property attributes such as <i>MinValue</i> and <i>HelpText</i> .
DropDown	Allows the user to type a value in a text box. You can further refine this property type by adding attributes such as <i>ValueList</i> and <i>Relationship</i> .
DropDownList	Requires the user to select from values in a drop-down menu. You can further refine this property type by adding attributes such as <i>ValueList</i> and <i>Relationship</i> .
Integer	Allows for an integer value. You can further refine the allowed value by adding attributes such as <i>MinValue</i> and <i>MaxValue</i> or <i>Integer</i> .
Label	Adds a read-only text label. You can further refine the allowed value by adding attributes such as <i>MinValue</i> and <i>HelpText</i> .
Link	Displays a link with the property display name as the link text and the property value as the URL. You can control the position of the new property by adding the <i>OrderIndex</i> attribute.
Notes	Provides for field in which to enter notes.
Password	Provides a field in which to enter a password value.
TextBox	Provides a text box in which to enter text characters.

- 7 Click **Required** if the machine owner must specify a value for this property.
- 8 Click **Save**.

The property is created and is visible on the Property Dictionary page.

Adding Property Attributes

Certain attribute values are available for each of the property definition control types.

Table 4-2. Property Attributes for Property Definition Control Types

Property Definition Control Type	Available Property Attribute Types
CheckBox	<ul style="list-style-type: none"> ■ HelpText ■ OrderIndex
DateTimeEdit	<ul style="list-style-type: none"> ■ HelpText ■ MinValue ■ OrderIndex

Table 4-2. Property Attributes for Property Definition Control Types (Continued)

Property Definition Control Type	Available Property Attribute Types
DropDown	<ul style="list-style-type: none"> ■ HelpText ■ OrderIndex ■ Relationship ■ ValueExpression ■ ValueList
DropDownList	<ul style="list-style-type: none"> ■ HelpText ■ OrderIndex ■ Relationship ■ ValueExpression ■ ValueList
Integer	<ul style="list-style-type: none"> ■ HelpText ■ Interval ■ MaxValue ■ MinValue ■ OrderIndex
Label	<ul style="list-style-type: none"> ■ HelpText ■ OrderIndex
Link	OrderIndex

Add a Help Text Attribute


You can use the HelpText attribute to add text that appears when the user points to the property name.

The HelpText attribute is available for all property value control types except Link.

Prerequisites

Log in to the vRealize Automation console as a **fabric administrator**.

Procedure

- 1 Select **Infrastructure > Blueprints > Property Dictionary**.
- 2 Click **Edit** in the Property Attributes column of the property row.
- 3 Click **New Property Attribute**.
- 4 Select **HelpText** from the **Type** drop-down menu.
- 5 Enter an attribute name in the **Name** text box.
This name is not visible in the user interface.
- 6 In the **Value** text box, type the help text that you want to display when the user pauses on the property display name.
- 7 Click the **Save** icon ()
- 8 (Optional) Add additional attributes.
- 9 Click **OK**.

Add an Order Index Attribute


You can use the OrderIndex attribute to control how the property name appears in the user interface.

The OrderIndex attribute is available for all property value control types except Link.

Prerequisites

Log in to the vRealize Automation console as a **fabric administrator**.

Procedure

- 1 Select **Infrastructure > Blueprints > Property Dictionary**.
- 2 Click **Edit** in the Property Attributes column of the property row.
- 3 Click **New Property Attribute**.
- 4 Select **OrderIndex** from the **Type** drop-down menu.
- 5 Enter an attribute name in the **Name** text box.
This name is not visible in the user interface.
- 6 In the **Value** text box, type the index number to use when you order the property display name in the user interface.
- 7 Click the **Save** icon ()
- 8 (Optional) Add additional attributes.
- 9 Click **OK**.

Add a Relationship Attribute

You can add a Relationship attribute to define a relationship between a drop-down property and another property, such that the value of the other property determines the possible values of the drop-down property.

The Relationship attribute is available for the DropDown and DropDownList property value control types.


In a property relationship, a parent property value determines the behavior of a child drop-down property. The child drop-down values are populated based on the value of the parent. If the value of the parent property changes, the child property updates with a list of possible values that are specific to the new parent property.

For more information, see [“Example of Creating a Relationship Between Two Properties and Adding Them to a Blueprint,”](#) on page 84.

Prerequisites

Log in to the vRealize Automation console as a **fabric administrator**.

Procedure

- 1 Select **Infrastructure > Blueprints > Property Dictionary**.
- 2 Click **Edit** in the Property Attributes column of the property row.
- 3 Click **New Property Attribute**.
- 4 Select **Relationship** from the **Type** drop-down menu.
- 5 Enter an attribute name in the **Name** text box.
This name is not visible in the user interface.
- 6 Type the property value or value expression in the **Value** text box.
- 7 Click the **Save** icon ()
- 8 (Optional) Add additional attributes.

- 9 Click **OK**.

Add a Value Expression Attribute


You can use the ValueExpression attribute to add an XML string to map the values of a parent property and a child property in a property relationship.

A ValueExpression attribute is available for the DropDown and DropDownList property value control types. For more information about property relationships, see [“Example of Creating a Relationship Between Two Properties and Adding Them to a Blueprint,”](#) on page 84.

Prerequisites

Log in to the vRealize Automation console as a **fabric administrator**.

Procedure

- 1 Select **Infrastructure > Blueprints > Property Dictionary**.
- 2 Click **Edit** in the Property Attributes column of the property row.
- 3 Click **New Property Attribute**.
- 4 Select **ValueExpression** from the **Type** drop-down menu.
- 5 Enter an attribute name in the **Name** text box.
This name is not visible in the user interface.
- 6 Enter the values that you want to use in the **Value** text box, for example an XML value expression or snippet that you formatted as a single line string with no line breaks.
- 7 Click the **Save** icon ()
- 8 (Optional) Add additional attributes.
- 9 Click **OK**.

Add a Value List Attribute

You can use the ValueList attribute to create a list of available values for a drop-down menu.


A ValueList attribute is available for the DropDown and DropDownList property value control types. See [“Example of Creating a Relationship Between Two Properties and Adding Them to a Blueprint,”](#) on page 84.

Prerequisites

Log in to the vRealize Automation console as a **fabric administrator**.

Procedure

- 1 Select **Infrastructure > Blueprints > Property Dictionary**.
- 2 Click **Edit** in the Property Attributes column of the property row.
- 3 Click **New Property Attribute**.
- 4 Select **ValueList** from the **Type** drop-down menu.
- 5 Enter an attribute name in the **Name** text box.
This name is not visible in the user interface.
- 6 Enter a comma-separated list of values in the **Value** text box to appear in the drop-down list.
For example, enter **Option 1,Option 2,Option 3**.

- 7 Click the **Save** icon ()
- 8 (Optional) Add additional attributes.
- 9 Click **OK**.

Add a Minimum Value Attribute

You can use the `MinValue` attribute to define a minimum time value or a minimum integer value in a range.

A `MinValue` attribute is available for the `DateTimeEdit` and `Integer` property value control types.

Prerequisites

Log in to the vRealize Automation console as a **fabric administrator**.


Procedure

- 1 Select **Infrastructure > Blueprints > Property Dictionary**.
- 2 Click **Edit** in the Property Attributes column of the property row.
- 3 Click **New Property Attribute**.
- 4 Select **MinValue** from the **Type** drop-down menu.
- 5 Enter an attribute name in the **Name** text box.

This name is not visible in the user interface.

- 6 Type the minimum value of the property in the **Value** text box.

For example, if you are using this attribute with the `DateTimeEdit` control, type a minimum earliest possible date, and optionally a minimum time, value using a `YYYY-MM-DD` or `MM/DD/YYYY` format and a time in `HH:MM` format followed by `AM` or `PM`. If you are using this attribute with the `Integer` control, type a minimum integer value.

- 7 Click the **Save** icon ()
- 8 (Optional) Add additional attributes.
- 9 Click **OK**.

Add a Maximum Value Attribute

Use the `MaxValue` attribute to define a maximum integer value.

A `MaxValue` attribute is available for the `Integer` property value control type.


Prerequisites

Log in to the vRealize Automation console as a **fabric administrator**.

Procedure

- 1 Select **Infrastructure > Blueprints > Property Dictionary**.
- 2 Click **Edit** in the Property Attributes column of the property row.
- 3 Click **New Property Attribute**.
- 4 Select **MaxValue** from the **Type** drop-down menu.
- 5 Enter an attribute name in the **Name** text box.

This name is not visible in the user interface.

- 6 Type an integer for the maximum value of the property in the **Value** text box.
- 7 Click the **Save** icon ().
- 8 (Optional) Add additional attributes.
- 9 Click **OK**.

Add an Interval Attribute


You can use the Interval attribute to specify an integer increment by which the property value is increased or decreased.

An Interval attribute is available for the Integer property value control type.

Prerequisites

Log in to the vRealize Automation console as a **fabric administrator**.

Procedure

- 1 Select **Infrastructure > Blueprints > Property Dictionary**.
- 2 Click **Edit** in the Property Attributes column of the property row.
- 3 Click **New Property Attribute**.
- 4 Select **Interval** from the **Type** drop-down menu.
- 5 Enter an attribute name in the **Name** text box.
This name is not visible in the user interface.
- 6 In the **Value** text box, type the increment integer by which the property value is increased or decreased when a user clicks the up or down arrows.
- 7 Click the **Save** icon ().
- 8 (Optional) Add additional attributes.
- 9 Click **OK**.

Example of Creating a Relationship Between Two Properties and Adding Them to a Blueprint

Define a relationship so that the value of a parent property determines the child property values that are available in a drop-down list.

In this example, a new `VirtualMachine.Network.Environment` property filters available network options based on the machine environment. The `VirtualMachine.Network.Environment` property acts as a filter for the networks that a user can select when confirming a machine request. An associated child property, `VirtualMachine.Network0.Name`, is also created and a relationship is formed between the two properties. The user must first select the environment before selecting from the list of networks that are applicable to the selected environment.

Table 4-3. Relationship Between `VirtualMachine.Network.Environment` Property and `VirtualMachine.Network0.Name` Property Menu Options

Values for the <code>VirtualMachine.Network.Environment</code> Property (parent)	Values for the <code>VirtualMachine.Network0.Name</code> Property (child)
Development	Development Network
Test	Test Network 1 Test Network 2
Production	Production Network Failover Network

Workflow Overview

You use the following workflow to create the relationship and add the properties to the blueprint.

- 1 Create the parent and child property definitions.
- 2 Add a Relationship attribute to the child property whose value is the name of the parent property.
- 3 Create a value expression that describes which values to display in the child drop-down list for each value of the parent property.
- 4 Add the value expression as an attribute of the child property.
- 5 Add both properties to a blueprint or property group.

The following example is a logical progression from the following topics.

- [“Add a Property Definition,”](#) on page 78
- [“Add a Relationship Attribute,”](#) on page 81
- [“Add a Value List Attribute,”](#) on page 82
- [“Add a Value Expression Attribute,”](#) on page 82

Instead of specifying a `ValueList` attribute for the child property, you can use an XML string in a `ValueExpression` attribute to specify the mapping between the values of the parent property and the possible values of the child property.

This example uses the `ValueList`, `ValueExpression`, and `Relationship` property attributes with the `DropDownList` property definition control type.

Example Procedure

- 1 Log in to vRealize Automation as a tenant administrator.
- 2 Select **Administration > Property Dictionary > Property Definitions**.
- 3 Create the parent property.
 - a Click the Add icon on the Property Definitions page.
 - b Enter `VirtualMachine.Network.Environment` in the **ID** text box.
 - c Enter `Environment` in the **Label** text box.
 - d Select a data type from the **Data type** drop-down menu.
 - e Enter a description that describes the intent of the property and any information that might help the consumer best use the property.
 - f Do not select the **Multiple Values** checkmark option.
 - g Select **Dropdown** from the **Display Advice** drop-down menu.
 - h Click **Save**.

- 4 Define the values for the parent property.
 - a In the `VirtualMachine.Network.Environment` property name row, click **Edit** in the Property Attributes column.
 - b Click **New Property Attribute**.
 - c Select **ValueList** from the **Type** drop-down menu.
 - d Enter **Values** in the **Name** text box.
 - e Enter **Development, Test, Production** in the **Value** text box.
 - f Click **Save**.
 - g Click **OK**.
- 5 Create the child property.
 - a Click **New Property Definition** on the Property Dictionary page.
 - b Enter `VirtualMachine.Network0.Name` in the **Name** text box.
 - c Enter **Select Network** in the **Display Name** text box.
 - d Select **DropDownList** from the **Control Type** drop-down menu.
 - e Click **Save**.
- 6 Define the relationship between the child and parent properties.
 - a In the `VirtualMachine.Network0.Name` property name row, click **Edit** in the Property Attributes column.
 - b Click **New Property Attribute**.
 - c Select **Relationship** from the **Type** drop-down menu.
 - d Enter **Parent** in the **Name** text box.
 - e Enter `VirtualMachine.Network.Environment` in the **Value** text box.
 - f Click **Save** and click **OK**.
- 7 In an XML editor, create the following value expression, which specifies the values of the child property based on the values of the parent property, and save it as an XML file:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<ArrayOfPropertyValue xmlns:xsi=
  "http://www.w3.org/2001/XMLSchema-instance">
  <PropertyValue>
    <FilterName>VirtualMachine.Network.Environment</FilterName>
    <FilterValue>Development</FilterValue>
    <Value>Development Network</Value>
  </PropertyValue>
  <PropertyValue>
    <FilterName>VirtualMachine.Network.Environment</FilterName>
    <FilterValue>Test</FilterValue>
    <Value>Test Network 1</Value>
  </PropertyValue>
  <PropertyValue>
    <FilterName>VirtualMachine.Network.Environment</FilterName>
    <FilterValue>Test</FilterValue>
    <Value>Test Network 2</Value>
  </PropertyValue>
  <FilterName>VirtualMachine.Network.Environment</FilterName>
```

```

    <FilterValue>Production</FilterValue>
    <Value>Production Network</Value>
  </PropertyValue>
  <PropertyValue>
    <FilterName>VirtualMachine.Network.Environment</FilterName>
    <FilterValue>Production</FilterValue>
    <Value>Failover Network</Value>
  </PropertyValue>
</ArrayOfPropertyValue>

```

- 8 Add the value expression, specifically the XML file content, that you just created to the child property.
 - a In a text editor, format the value expression so that it is a single line string with no line breaks.
 - b In the `VirtualMachine.Network0.Name` property name row, click **Edit** in the Property Attributes column.
 - c Click **New Property Attribute**.
 - d Select **ValueExpression** from the **Type** drop-down menu.
 - e Enter **Expression** in the **Name** text box.
 - f Copy the value expression from the text editor and paste it into the **Value** text box.
 - g Click **Save** and click **OK**.
- 9 Add both properties to a blueprint.
 - a Create a new blueprint or edit an existing blueprint.
 - b Click the **Properties** tab on the blueprint page.
 - c Click **New Property**.
 - d Enter **VirtualMachine.Network.Environment** in the **Name** text box. This name must be an exact character match for the property name you created for the parent property in this example.
 - e Leave the **Value** text box blank.
- 10 Select the **Prompt User** check box.
- 11 Click **OK**.
- 12 Click **New Property**.
- 13 Enter **VirtualMachine.Network0.Name** in the **Name** text box.
- 14 Click **OK**.

When a user uses this blueprint to request a machine, the **Environment** and **Select Network** drop-down menus now appear on the Confirm Machine Request page and are initially empty. The user can select an environment, which then narrows the list of networks that they can select for the machine that they are requesting.

Creating Property Control Layouts

A property control layout acts as a named container to which you can add properties and specify the order in which the properties appear in a blueprint or build profile. The layout defines the order of the properties that display in the form.

With control layouts you can group properties together and specify the order in which they are displayed to users on the machine confirmation page. Add the control layout and the individual properties that it contains to the blueprint.

Add a Property Control Layout

You create a property control layout to define how properties are displayed to users.


After you create the layout, a tenant administrator or business group manager can add it to a blueprint on the **Properties** tab.

Prerequisites

Log in to the vRealize Automation console as a **fabric administrator**.

Create the property definitions to add to the property control layout.

Procedure

- 1 Select **Infrastructure > Blueprints > Property Dictionary**.
- 2 Click **New Property Layout**.
- 3 Enter a property layout name in the **Name** text box.
This is the name that is added to the blueprint or build profile. For example, type **NetworkLayout**.
- 4 Click the **Save** icon ()
- 5 Click **Edit** in the Property Instances column of the property layout name row, for example the **NetworkLayout** name row.
- 6 Type **1** in the **Order** text box.
- 7 From the **Property Definition** drop-down menu, select the name of the first property to display, for example **VirtualMachine.Network.Environment**.
- 8 Type **2** in the **Order** text box.
- 9 From the **Property Definition** drop-down menu, select the name of the second property to display, for example **VirtualMachine.Network0.Name**.
- 10 Click **OK**.

A custom layout is now available for a tenant administrator or business group manager to add to a blueprint or build profile.

Index

A

Amazon, available custom properties **49**
Application Services, adding custom properties
for use in application deployment **13**

B

blueprints, custom properties **45**
build profiles
creating **15**
creating property sets **16**
using property sets **14**

C

custom **10**
custom properties, a properties **49**
custom properties
creating build profiles **15**
creating property sets **16**
custom property types **13**
for FlexClone blueprints **25**
for basic workflow blueprints **20**
for clone blueprints **22, 37**
for linked clone blueprints **27**
Linux Kickstart provisioning **29**
listed by function **19**
listed by name **49**
machine provisioning **10**
networking **40**
o properties **55**
Openstack endpoints **20**
order of precedence **11**
properties.ini **9**
properties.xml **9**
PXE provisioning **42**
reasons for using **9**
related guides **5**
required for integrating HP Server
Automation **46**
required and optional **45**
SCCM provisioning **31**
specifying properties **10**
specifying for multi-machine services **12**
tailoring the user interface **77**
updated property information **7**
using the property dictionary **77**

vApp blueprints **35**

WIM provisioning **32**

custom properties, b properties **50**
custom properties, c properties **51**
custom properties, e properties **52**
custom properties, h properties **52**
custom properties, i properties **53**
custom properties, l properties **54**
custom properties, m properties **55**
custom properties, p properties **56**
custom properties, r properties **57**
custom properties, s properties **57**
custom properties, v properties **59**
custom properties, x properties **76**

G

guest agents, using custom properties **9**

H

HP Server Automation, required custom
properties **46**

L

lease synchronization, ensure machines do not
expire before defined duration **35**
Linux Kickstart provisioning, custom
properties **29**

M

machine provisioning, custom properties **10**
multi-machine services, specifying custom
properties **12**

O

Openstack, custom properties for endpoints **20**

P

property dictionary
adding an interval attribute **84**
adding an order index attribute **80**
adding help text **80**
adding property attributes **79**
adding property definitions **78**
adding a list of possible menu values **82**
adding a maximum value attribute **83**
adding a minimum value attribute **83**
adding a property control layout **88**

- adding a property definition **78**
- adding a relation attribute **81, 84**
- adding a dropdownlist control type **84**
- adding a valuelist attribute **84**
- adding a valuexpression attribute **84**
- adding properties to blueprints **84**
- creating a property control layout **87**
- property dictionary,adding a value expression attribute **82**
- property sets
 - creating **16**
 - creating the XML file **16**
 - using with build profiles **14**
- PXE provisioning, custom properties **42**

S

- SCCM provisioning, custom properties **31**

V

- vApp blueprints, adding custom properties **35**
- VirtualMachine.Network, specifying a network device **40**

W

- WIM provisioning, custom properties **32**