Custom Properties

vCloud Automation Center 6.0.0

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition. To check for more recent editions of this document, see http://www.vmware.com/support/pubs.

EN-001352-01

vmware®

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

http://www.vmware.com/support/

The VMware Web site also provides the latest product updates.

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

docfeedback@vmware.com

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

Contents

Custom Properties Reference

Custom Properties Reference provides information about the custom properties, and their usage, that are available when using VMware vCloud® Automation Center™.

This documentation is intended to be used with the vCloud Automation Center product documentation set, including the following guides:

- *Installation and Configuration*
- *IaaS Configuration for Virtual Platforms*
- *IaaS Configuration for Cloud Platforms*
- *IaaS Configuration for Physical Machines*
- *IaaS Configuration for Multi-Machine Services*
- *IaaS Configuration for vCloud Director*
- *IaaS Integration for BMC BladeLogic*
- *IaaS Integration for HP Server Automation*
- *Tenant Administration*
- *System Administration*
- *Advanced Service Design*
- *Advanced Service Designer Configuration*

NOTE Not all features and capabilities of vCloud Automation Center are available in all editions. For a comparison of feature sets in each edition, see https://www.vmware.com/products/vcloud-automation-center/.

Intended Audience

This information is intended for IaaS administrators, fabric administrators, and business group managers of vCloud Automation Center. 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.

1 Using Custom Properties

VMware vCloud® Automation Center™ custom properties allow you to add attributes of the machines your site provisions, or to override their standard 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:

Specifying Custom Properties

Custom properties can be used to control many aspects of 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. Property names and property values should not be identical.

Custom Property Uses

A machine is defined by a number of properties, for example its operating system and memory size. Custom properties allow you to control aspects of the machines that users can provision.

Some properties are determined by standard settings that must be specified for all machines. For example, memory and disk size values are required for all blueprints. Additional properties can be specified individually or in build profiles in blueprints and in reservations. Custom properties can be used to add values or override existing or default values for the following sorts of information:

- machine operating system
- virtualization platform
- build settings such as disk size
- integration with an external system

When a property is added to a blueprint or a build profile, it can be marked as a required property. When 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.

Property values are recorded in the vCloud Automation Center database by the application's manager service. They are also recorded on the provisioned machine in the following files by the guest agent:

- *%SystemDrive%**VRM**Build*\Properties.ini
- *%SystemDrive%**VRM**Build*\Properties.xml

Custom Property Types

There are four types of custom properties.

Custom property types have the following unique characteristics:

- 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 impact 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 are determined by the template; they cannot be used for cloned machines. External and updated property types can be used for cloned machines.

All reserved custom properties, except read-only properties `VirtualMachine.Admin.AgentID`, `VirtualMachine.Admin.UUID`, and `Virtual–Machine.Admin.Name`, can be changed in the database using the applicable machine menu.

Create a Property Set

Fabric administrators can create their own groupings of related custom properties and then load them into vCloud Automation Center for use in build profiles.

Property sets are available to fabric administrators of all tenants.

Create a Property Set XML File

Property sets are defined in an XML file and then uploaded into vCloud Automation Center by a fabric administrator.

NOTE If you edit a property set that is already in use in a build profile, vCloud Automation Center does not automatically update the build profile. A fabric administrator must reload the property set into 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 inside the `<Doc>` element.

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

5 Define the attributes of the custom property you want 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 for each property you want to include in the property set.

7 Save and close the file.

A fabric administrator can now upload your property set XML file to vCloud Automation Center. See "Add a Property Set to vCloud Automation Center," on page 10.

Add a Property Set to vCloud Automation Center

After you create a property set XML file, a fabric administrator can upload the property set into vCloud Automation Center.

Prerequisites

- Log in to the vCloud Automation Center console as a **fabric administrator**.
- "Create a Property Set XML File," on page 9.

Procedure

1 Select **Infrastructure > Blueprints > Build Profiles**.

2 Click **Manage Property Sets**.

3 Click **Browse** to select the **Property set XML file name**.

4 Type 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

This section lists and described the vCloud Automation Center custom properties organized by function.

These functional groupings are also captured in context-relevant guides in the vCloud Automation Center product documentation set.

This chapter includes the following topics:

Custom Properties for Basic Workflow Blueprints

While there are a number of custom properties that you can apply to your basic workflow blueprint, there are some that are very common for basic workflow blueprints.

The Custom Properties for Basic Workflow Blueprints table describes the most commonly used custom properties for this blueprint type.

Table 2-1. Custom Properties for Basic Workflow Blueprints

Custom Property	Description
`VirtualMachine.CDROM.Attach`	Set to **`False`** to provision the machine without a CD-ROM device. The default is **`True`**.
`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.

For a full list of custom properties, see *Custom Properties Reference*.

Custom Properties for Clone Blueprints

You can apply many different custom properties to clone blueprints, but there are some that are very common and some that are required for certain configurations.

The Custom Properties for Clone Blueprints table describes the most commonly used custom properties for this blueprint type.

Table 2-2. Custom Properties for Clone Blueprints

Custom Property	Description
`VirtualMachine.Network`*N*`.ProfileName`	Specifies the name of a network profile from which to assign a static IP address to network device *N*or from which to obtain the range of static IP addresses that can be assigned to network device *N* of a cloned machine, where *N*=0 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 `VirtualMachine.Network`*N*`.Name` custom property.
`Linux.ExternalScript.Name`	Specifies the name of a customization script that the Linux 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.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.

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

Custom Property	Description
`Linux.ExternalScript.Server`	Specifies the name of the NFS server on which the Linux external customization script named in `Linux.ExternalScript.Name` is located.
`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.

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

Table 2-3. Custom Properties for Customizing Cloned Machines with a Guest Agent

Custom Property	Description
`VirtualMachine.Admin.UseGuestAgent`	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. Set to **`False`** to disable the guest agent.
`VirtualMachine.Disk`*N*`.Size`	Defines the size in GB of disk N. For example, to give a size of 150GB to a disk G, define the custom property `VirtualMachine.Disk0.Size` and enter a value of **`150`**. Disk numbering must be sequential. By default a machine has one disk referred to by `VirtualMachine.Disk0.Size`, where size is specified by the **Storage** option on the **Build Information** tab of the blueprint from which the machine is provisioned. The value in the **Storage** option on the **Build Information** tab overwrites the value in the `VirtualMachine.Disk0.Size` property. The `VirtualMachine.Disk0.Size` property is not available as a custom property because of its relationship with the**Storage** option on the **Build Information** tab. More disks can be added by specifying `VirtualMachine.Disk1.Size`, `VirtualMachine.Disk2.Size` and so on. `VirtualMachine.Admin.TotalDiskUsage` always represents the total of the `.DiskN.Size` properties plus the `VMware.Memory.Reservation` size allocation.
`VirtualMachine.Disk`*N*`.Label`	Specifies the label of a machine's disk . The disk label maximum is 32 characters. Disk numbering must be sequential.
`VirtualMachine.Disk`*N*`.Letter`	Specifies the 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 `VirtualMachine.Disk1.Letter` and enter the value `D`. Disk numbering must be sequential.
`VirtualMachine.Admin.CustomizeGuestOSDelay`	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.

Table 2-3. Custom Properties for Customizing Cloned Machines with a Guest Agent (Continued)

Custom Property	Description
`VirtualMachine.Customize.WaitComplete`	Set to **`True`** to hold the provisioning workflow until customizations are finished.
`VirtualMachine.Software0.ScriptPath`	Specifies the 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 `.bat` file. You can pass custom property values as parameters to the script by inserting {*CustomPropertyName*} in the path string. For example, if you have a custom property named `ActivationKey` whose value is **`1234`**, the script path is `D:\InstallApp.bat -key {ActivationKey}`. The guest agent runs the command `D:\InstallApp.bat -key 1234`. Your script file can then be programmed to accept and use this value.

For a full list of custom properties, see *Custom Properties Reference*.

Custom Properties for FlexClone Blueprints

You can apply a number of different custom properties to your FlexClone blueprint, but there are some that are very common and some that are required for certain configurations.

The Custom Properties for FlexClone Blueprints table describes the most commonly used custom properties for this blueprint type.

Table 2-4. Custom Properties for FlexClone Blueprints

Custom Property	Description
`VirtualMachine.Network`*N*`.ProfileName`	Specifies the name of a network profile from which to assign a static IP address to network device *N*or from which to obtain the range of static IP addresses that can be assigned to network device *N* of a cloned machine, where *N*=0 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 `VirtualMachine.Network`*N*`.Name` custom property.
`Linux.ExternalScript.Name`	Specifies the name of a customization script that the Linux 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.

Table 2-4. Custom Properties for FlexClone Blueprints (Continued)

Custom 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.Server`	Specifies the name of the NFS server on which the Linux external customization script named in `Linux.ExternalScript.Name` is located.
`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.

If you have 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 scenario.

Table 2-5. Custom Properties for Customizing FlexClone Machines with a Guest Agent

Custom Property	Description
`VirtualMachine.Admin.UseGuestAgent`	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. Set to **`False`** to disable the guest agent.
`VirtualMachine.Disk`*N*`.Size`	Defines the size in GB of disk N. For example, to give a size of 150GB to a disk G, define the custom property `VirtualMachine.Disk0.Size` and enter a value of **`150`**. Disk numbering must be sequential. By default a machine has one disk referred to by `VirtualMachine.Disk0.Size`, where size is specified by the **Storage** option on the **Build Information** tab of the blueprint from which the machine is provisioned. The value in the **Storage** option on the **Build Information** tab overwrites the value in the `VirtualMachine.Disk0.Size` property. The `VirtualMachine.Disk0.Size` property is not available as a custom property because of its relationship with the**Storage** option on the **Build Information** tab. More disks can be added by specifying `VirtualMachine.Disk1.Size`, `VirtualMachine.Disk2.Size` and so on. `VirtualMachine.Admin.TotalDiskUsage` always represents the total of the `.DiskN.Size` properties plus the `VMware.Memory.Reservation` size allocation.
`VirtualMachine.Disk`*N*`.Label`	Specifies the label of a machine's disk . The disk label maximum is 32 characters. Disk numbering must be sequential.
`VirtualMachine.Disk`*N*`.Letter`	Specifies the 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 `VirtualMachine.Disk1.Letter` and enter the value `D`. Disk numbering must be sequential.
`VirtualMachine.Admin.CustomizeGuestOSDelay`	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.

Table 2-5. Custom Properties for Customizing FlexClone Machines with a Guest Agent (Continued)

Custom Property	Description
`VirtualMachine.Customize.WaitComplete`	Set to **`True`** to hold the provisioning workflow until customizations are finished.
`VirtualMachine.Software0.ScriptPath`	Specifies the 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 `.bat` file. You can pass custom property values as parameters to the script by inserting {*CustomPropertyName*} in the path string. For example, if you have a custom property named `ActivationKey` whose value is **`1234`**, the script path is `D:\InstallApp.bat -key {ActivationKey}`. The guest agent runs the command `D:\InstallApp.bat -key 1234`. Your script file can then be programmed to accept and use this value.

For a full list of custom properties, see *Custom Properties Reference*.

Custom Properties for Linked Clone Blueprints

While there are a number of custom properties that you can apply to your linked clone blueprint, there are some that are very common for this blueprint type. *VirtualMachine.DiskN* properties are required when specifying volume properties for a linked clone blueprint.

The Custom Properties for Linked Clone Blueprints table describes the most commonly used custom properties for this blueprint type.

Table 2-6. Custom Properties for Linked Clone Blueprints

Custom Property	Description
`VirtualMachine.Disk`*N*`.Storage`	Specifies that datastore on which to place the machine disk *N*. This property is also used to add a single datastore to a linked clone blueprint. *N* is the index (starting at 0) of the volume to assign. Type the name of the datastore to assign to the volume in the **Value** text box. This is the datastore name as it appears in the Storage Path column on the Edit Compute Resource page. Disk numbering must be sequential.
`VirtualMachine.Disk`*N*`.StorageReservationPolicy`	Specifies a storage reservation policy to find storage for disk *N*. Also assigns a storage reservation policy to a volume. You can use this disk property only with linked clone blueprints. To use this property, substitute the volume number for *N* in the property name and specify a storage reservation policy as the value. This property is equivalent to the **Storage Reservation Policy** drop-down menu in the Volumes list on the **Build Information** tab of the blueprint page. Disk numbering must be sequential.

Table 2-6. Custom Properties for Linked Clone Blueprints (Continued)

Custom Property	Description
`VirtualMachine.Disk`*N*`.Size`	Defines the size in GB of disk N. For example, to give a size of 150GB to a disk G, define the custom property `VirtualMachine.Disk0.Size` and enter a value of **`150`**. Disk numbering must be sequential. By default a machine has one disk referred to by `VirtualMachine.Disk0.Size`, where size is specified by the **Storage** option on the **Build Information** tab of the blueprint from which the machine is provisioned. The value in the **Storage** option on the **Build Information** tab overwrites the value in the `VirtualMachine.Disk0.Size` property. The `VirtualMachine.Disk0.Size` property is not available as a custom property because of its relationship with the**Storage** option on the **Build Information** tab. More disks can be added by specifying `VirtualMachine.Disk1.Size`, `VirtualMachine.Disk2.Size` and so on. `VirtualMachine.Admin.TotalDiskUsage` always represents the total of the `.DiskN.Size` properties plus the `VMware.Memory.Reservation` size allocation.
`VirtualMachine.Disk`*N*`.Label`	Specifies the label of a machine's disk . The disk label maximum is 32 characters. Disk numbering must be sequential.
`VirtualMachine.Disk`*N*`.Letter`	Specifies the 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 `VirtualMachine.Disk1.Letter` and enter the value `D`. Disk numbering must be sequential.
`MaximumProvisionedMachines`	Specifies the maximum number of linked clones for one machine snapshot. The default is 20.
`Linux.ExternalScript.Name`	Specifies the name of a customization script that the Linux 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.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.Server`	Specifies the name of the NFS server on which the Linux external customization script named in `Linux.ExternalScript.Name` is located.
`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.

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

Table 2-7. Custom Properties for Customizing Cloned Machines with a Guest Agent

Custom Property	Description
`VirtualMachine.Admin.UseGuestAgent`	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. Set to **`False`** to disable the guest agent.
`VirtualMachine.Admin.CustomizeGuestOSDelay`	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.
`VirtualMachine.Customize.WaitComplete`	Set to **`True`** to hold the provisioning workflow until customizations are finished.
`VirtualMachine.Software0.ScriptPath`	Specifies the 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 `.bat` file. You can pass custom property values as parameters to the script by inserting {*CustomPropertyName*} in the path string. For example, if you have a custom property named `ActivationKey` whose value is **`1234`**, the script path is `D:\InstallApp.bat -key {ActivationKey}`. The guest agent runs the command `D:\InstallApp.bat -key 1234`. Your script file can then be programmed to accept and use this value.

For a full list of custom properties, see *Custom Properties Reference*.

Custom Properties for Linux Kickstart Blueprints

While there are a number of custom properties that you can apply to your Linux Kickstart blueprint, there are some that are very common and some that are required.

The Required Custom Properties for Linux Kickstart Blueprints table describes the required custom properties for this blueprint type.

Table 2-8. Required Custom Properties for Linux Kickstart Blueprints

Custom Property	Description
`VMware.VirtualCenter.OperatingSystem`	Specifies the vCenter Server guest operating system version (`VirtualMachineGuestOsIdentifier`) 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 build profiles using one of several property sets, for example, `VMware[OS_Version]Properties`, that are predefined to include the correct `VMware.VirtualCenter.OperatingSystem` values. This property is for virtual provisioning. For a list of currently accepted values, see the VMware® vCenter Server™ documentation.
`Image.ISO.Location`	Specifies the location of the ISO image to boot from. The format of this value depends on your platform. For details, see the documentation provided by your platform. This property is required for WIM-based provisioning, Linux kickstart and autoYaST provisioning, and SCCM-based provisioning.

Table 2-8. Required Custom Properties for Linux Kickstart Blueprints (Continued)

Custom Property	Description
`Image.ISO.Name`	Specifies the name of the ISO image from which to boot. The format of this value depends on your platform. For details, see the documentation provided by 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 *username@domain*. 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 `Image.ISO.UserName` property. For Dell iDRAC integrations where the image is located on a CIFS share that requires authentication to access.

The Common Custom Properties for Linux Kickstart Blueprints table describes the most commonly used custom properties for this blueprint type.

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

Custom Property	Description
`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.
`Machine.SSH`	Set to **`True`** to enable the **Connect Using SSH** option 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.

For a full list of custom properties, see *Custom Properties Reference*.

Custom Properties for SCCM Blueprints

While there are a number of custom properties that you can apply to your SCCM blueprint, there are some that are very common and some that are required.

The Required Custom Properties for SCCM Blueprints table describes the required custom properties for this blueprint type.

Table 2-10. Required Custom Properties for SCCM Blueprints

Custom Property	Description
`Image.ISO.Location`	Specifies the location of the ISO image to boot from. The format of this value depends on your platform. For details, see the documentation provided by your platform. This property is required for WIM-based provisioning, Linux kickstart and autoYaST provisioning, and SCCM-based provisioning.
`Image.ISO.Name`	Specifies the name of the ISO image from which to boot. The format of this value depends on your platform. For details, see the documentation provided by your platform. This property is required for WIM-based provisioning, Linux kickstart and autoYaST provisioning, and SCCM-based provisioning.

Table 2-10. Required Custom Properties for SCCM Blueprints (Continued)

Custom Property	Description
`Image.ISO.UserName`	Specifies the user name to access the CIFS share in the format *username@domain*. 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 `Image.ISO.UserName` 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.
`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 `SCCM.Server.UserName` property.

The Common Custom Properties for SCCM Blueprints table describes the most commonly used custom properties for this blueprint type.

Table 2-11. Common Custom Properties for SCCM Blueprints

Custom Property	Description
`SCCM.CustomVariable.`	Specifies the value of a custom variable, where *Name* 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.

For a full list of custom properties, see *Custom Properties Reference*.

Custom Properties for WIM Blueprints

While there are a number of custom properties that you can apply to your WIM blueprint, there are some that are very common and some that are required for certain integration types.

The Required Custom Properties for WIM Blueprints table describes the required custom properties for this blueprint type.

Table 2-12. Required Custom Properties for WIM Blueprints

Custom Property	Description
VMware.VirtualCenter.OperatingSystem	Specifies the vCenter Server guest operating system version (VirtualMachineGuestOsIdentifier) 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 build profiles using one of several property sets, for example, VMware[OS_Version]Properties, that are predefined to include the correct VMware.VirtualCenter.OperatingSystem values. This property is for virtual provisioning. For a list of currently accepted values, see the VMware® vCenter Server™ documentation.
Image.ISO.Location	Specifies the location of the ISO image to boot from. The format of this value depends on your platform. For details, see the documentation provided by your platform. This property is required for WIM-based provisioning, Linux kickstart and autoYaST provisioning, and SCCM-based provisioning.
Image.ISO.Name	Specifies the name of the ISO image from which to boot. The format of this value depends on your platform. For details, see the documentation provided by 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 *username@domain*. 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 Image.ISO.UserName 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, for example \\server\share$.
Image.WIM.Name	Specifies the name of the WIM file located by the Image.WIM.Path property.
Image.WIM.Index	Specifies the index used to extract the desired image from the WIM file.
Image.Network.User	Specifies the user name with which to map the WIM image path (Image.WIM.Path) 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 Image.Network.User property.

The Common Custom Properties for WIM Blueprints table describes the most commonly used custom properties for this blueprint type.

Table 2-13. Common Custom Properties for WIM Blueprints

Custom Property	Description
`SysPrep.`	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. *Section* represents the name of the section of the SysPrep answer file, for example **`GuiUnattended`** or **`UserData`**. *Key* 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`**. For a full list of sections, keys, and accepted values, see the System Preparation Utility for Windows documentation.
`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.

For a full list of custom properties, see *Custom Properties Reference*.

Custom Properties for vApp Blueprints

There are several custom properties that a tenant administrator or business group manager can apply to a vApp or vApp component blueprint.

You can add custom properties to a blueprint or build profile 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 build profile to assign custom properties to the vApp component blueprint associated with the Windows machines.

The following table describes the most commonly used custom properties for this blueprint type.

Table 2-14. Custom Properties for vApp Blueprints

Custom Property	Description
`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.

Table 2-14. Custom Properties for vApp Blueprints (Continued)

Custom Property	Description
`VCloud.Template.MakeIdenticalCopy`	Set to **`True`** to clone an identical copy of the vApp template in vCloud Director and provision the results in vCloud Automation Center. 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. Set to **`False`** to clone a copy of the vApp template with settings specified by the vApp and vApp component blueprints. The storage path specified in a vApp component blueprint, operating system or when requesting a vApp machine, is used if the **Make Identical Copy** option in the vApp template properties was selected.
`VMware.SCSI.Type`	For vApps, specifies the SCSI machine type using one of the following case-sensitive values: ■ **`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. ■ **`VirtualSCSI`** Use para-virtualization emulation for the virtual disk. ■ **`none`** Use if a SCSI controller does not exist for this machine.

For a full list of custom properties, see *Custom Properties Reference*.

Custom Properties for Networking

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

NOTE This information does not apply to Amazon Web Services.

By default a virtual machine has one network device configured with custom properties named *VirtualMachine.Network0.**. For example, the name of the network for the default network device is specified in the *VirtualMachine.Network0.Name* property. Additional network devices can be configured on machines by specifying properties named *VirtualMachine.Network1.**, and so on.

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 2-15. Custom Properties for Networking Configuration

Custom Property	Description
`VirtualMachine.Network`*N*`.Address`	Specifies the IP address of a network device `N` in a machine provisioned with a static IP address.
`VirtualMachine.Network`*N*`.MacAddressType`	Indicates whether the MAC address of network device *N* is auto-generated or user-defined. This property is available for cloning. The default value is **`generated`**. If the value is **`static`**, you must also use `VirtualMachine.Network`*N*`.MacAddress` to specify the MAC address.

Table 2-15. Custom Properties for Networking Configuration (Continued)

Custom Property	Description
`VirtualMachine.Network`*N*`.MacAddress`	Specifies the MAC address of a network device *N*. This property is available for cloning. If the value of `VirtualMachine.Network`*N*`.MacAddressType` is **`generated`**, this property contains the generated address. If the value of `VirtualMachine.Network.`*N*`.MacAddressType` is **`static`**, this poperty 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.
`VirtualMachine.Network`*N*`.Name`	Specifies the network to which a network device *N* in a virtual machine is attached. By default, a network is assigned from the network paths available on the reservation on which the machine is provisioned. 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.
`VirtualMachine.Network`*N*`.PortID`	Specifies the port ID to use for network device *N* when using a dvPort group with a vSphere distributed switch.
`VirtualMachine.Network`*N*`.ProfileName`	Specifies the name of a network profile from which to assign a static IP address to network device *N*or from which to obtain the range of static IP addresses that can be assigned to network device *N* of a cloned machine, where *N*=0 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.
■ `VirtualMachine.Network`*N*`.SubnetMask` ■ `VirtualMachine.Network`*N*`.Gateway` ■ `VirtualMachine.Network`*N*`.PrimaryDns` ■ `VirtualMachine.Network`*N*`.SecondaryDns` ■ `VirtualMachine.Network`*N*`.PrimaryWins` ■ `VirtualMachine.Network`*N*`.SecondaryWins` ■ `VirtualMachine.Network`*N*`.DnsSuffix` ■ `VirtualMachine.Network`*N*`.DnsSearchSuffixes`	Configures attributes of the network profile specified in `VirtualMachine.Network`*N*`.ProfileName`.

For a full list of custom properties, see *Custom Properties Reference*.

Custom Properties for PXE Provisioning

While PXE is the only provisioning method supported for Cisco UCS Manager, you can initiate WIM, SCCM, or Linux Kickstart provisioning by using the NBP and calling the appropriate custom properties. You can also call your own PowerShell scripts using custom properties.

Table 2-16. Calling Custom PowerShell Scripts

Custom Property	Description
`Pxe.Setup.ScriptName`	Specifies a custom EPI PowerShell script to run on the machine before it is booted using the PXE network boot program. The value is the name assigned to the script when it is uploaded to the model manager.
`Pxe.Clean.ScriptName`	Specifies the name of a EPI PowerShell script installed in the vCloud Automation Center 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.

Table 2-17. 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.
`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 `SCCM.Server.UserName` property.
`SCCM.CustomVariable.`	Specifies the value of a custom variable, where *Name* 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.

Table 2-18. 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 K.
`Image.WIM.Path`	Specifies the UNC path to the WIM file from which an image is extracted during WIM-based provisioning, for example `\\server\share$`.
`Image.WIM.Name`	Specifies the name of the WIM file located by the `Image.WIM.Path` property.
`Image.WIM.Index`	Specifies the index used to extract the desired image from the WIM file.
`Image.Network.User`	Specifies the user name with which to map the WIM image path (`Image.WIM.Path`) to a network drive on the provisioned machine. This is typically a domain account with access to the network share.

Table 2-18. PXE and WIM Provisioning (Continued)

Custom Property	Description
`Image.Network.Password`	Specifies the password associated with the `Image.Network.User` property.
`SysPrep.`	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. *Section* represents the name of the section of the SysPrep answer file, for example **`GuiUnattended`** or **`UserData`**. *Key* 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`**. For a full list of sections, keys, and accepted values, see the System Preparation Utility for Windows documentation.

Linux Kickstart provisioning does not require any custom properties.

For a full list of custom properties, see *Custom Properties Reference*.

Custom Properties for BMC BladeLogic Configuration Manager Integration

Some custom properties are required for BMC BladeLogic integrations and some are optional and commonly used with these types of blueprints.

Required Custom Properties for BMC BladeLogic Integrations

The Custom Properties Required for BMC BladeLogic Integrations table describes all custom properties that are required for a blueprint to work with BMC BladeLogic.

Table 2-19. Custom Properties Required for BMC BladeLogic Integrations

Custom Property	Description
`VirtualMachine.EPI.Type`	Specifies the type of external provisioning infrastructure.
`BMC.Software.Install`	Set to **`True`** to enable BMC BladeLogic integration.
`EPI.Server.Name`	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.
`BMC.Service.Profile`	Specifies the name of the default authentication profile on the BMC BladeLogic server.

Table 2-19. Custom Properties Required for BMC BladeLogic Integrations (Continued)

Custom Property	Description
`BMC.Software.BatchLocation`	Specifies the location in BMC BladeLogic of software jobs to be 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.
`VMware.VirtualCenter.OperatingSystem`	Specifies the vCenter Server guest operating system version (`VirtualMachineGuestOsIdentifier`) 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 build profiles using one of several property sets, for example, `VMware[OS_Version]Properties`, that are predefined to include the correct `VMware.VirtualCenter.OperatingSystem` values. This property is for virtual provisioning. For a list of currently accepted values, see the VMware® vCenter Server™ documentation.

Custom Properties That Make BMC BladeLogic Software Jobs Available

Your fabric administrator can configure BMC BladeLogic jobs for vCloud Automation Center integrations by either making all software jobs available to machine requesters to select from, or by specifying a software job to apply to all machines provisioned from your blueprint.

Table 2-20. Custom Properties to Make Software Jobs Available

Custom Property	Description
`LoadSoftware`	Set to **`True`** to make the software jobs listed in `Website\Software.txt` available to the user requesting the machine.

`VRM.Software.Id`*NNNN*	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 Integrations

The Optional Custom Properties for BMC BladeLogic Integrations table describes optional custom properties commonly used with BMC BladeLogic blueprints. These custom properties are not required.

Table 2-21. Optional Custom Properties for BMC BladeLogic Integrations

Property	Definition
BMC.AddServer.Delay	Specifies the number of seconds to wait before adding the machine to BMC BladeLogic. 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

Some custom properties are required for HP Server Automation integrations, while others are optional and commonly used with these types of blueprints.

Required Custom Properties for HP Server Automation Integration

The Custom Properties Required for HP Server Automation Integrations table describes all of the custom properties that are required for a blueprint to work with HP Server Automation.

Table 2-22. Required Custom Properties for HP Server Automation Integration

Property	Definition
VMware.VirtualCenter.OperatingSystem	Specifies the vCenter Server guest operating system version (VirtualMachineGuestOsIdentifier) 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 build profiles using one of several property sets, for example, VMware[OS_Version]Properties, that are predefined to include the correct VMware.VirtualCenter.OperatingSystem values. This property is for virtual provisioning.
VirtualMachine.EPI.Type	Specifies the type of external provisioning infrastructure.
EPI.Server.Name	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.
Opsware.Software.Install	Set to **True** to allow HP Server Automation to install software.
Opsware.Server.Name	Specifies the fully qualified name of the server automation server.
Opsware.Server.Username	Specifies the user name provided when a password file in the agent directory was created. This user name requires administrative access to the HP Server Automation instance, for example **opswareadmin**.
Opsware.BootImage.Name	Specifies the boot image value as defined in HP Server Automation, for example **winpe32** for the 32-bit WinPE image. The property is not required when provisioning by cloning.
Opsware.Customer.Name	Specifies a customer name value as defined in HP Server Automation, for example **MyCompanyName**.
Opsware.Facility.Name	Specifies a facility name value as defined in HP server automation, for example **Cambridge**.

Table 2-22. Required Custom Properties for HP Server Automation Integration (Continued)

Property	Definition
`Opsware.Machine.Password`	Specifies the default local administrator password for an operating system sequence WIM image such as `Opsware.OSSequence.Name` as defined in HP Server Automation, for example **`P@ssword1`**.
`Opsware.OSSequence.Name`	Specifies the operating system sequence name value as defined in HP Server Automation, for example **`Windows 2003 WIM`**.
`Opsware.Realm.Name`	Specifies the realm name value as defined in HP Server Automation, for example **`Production`**.
`Opsware.Register.Timeout`	Specifies the time, in seconds, to wait for creation of a provisioning job to complete.
`Opsware.WOL.Enabled`	Set to **`False`** for virtual provisioning.
`Opsware.WOL.Delay`	Specifies the time, in seconds, to wait before running Wake-On-LAN. This setting is ignored if `Opsware.WOL.Enabled` is set to **`False`**.
`VirtualMachine.CDROM.Attach`	Set to **`False`** to provision the machine without a CD-ROM device. The default is **`True`**.
`Linux.ExternalScript.Name`	Specifies the name of a customization script that the Linux 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.
`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.

Optional Custom Properties for HP Server Automation Integration

The Optional Custom Properties for HP Server Automation Integrations table describes all of the custom properties that are optional for a blueprint to work with HP Server Automation.

Table 2-23. Optional Custom Properties for HP Server Automation Integration

Property	Definition
`Opsware.ProvFail.Notify`	(Optional) Specifies the notification email address for HP Server Automation to use in the event of provisioning failure.
`Opsware.ProvFail.Notify`	(Optional) Specifies the HP Server Automation user to assign ownership to if provisioning fails, for example **`opswareadmin`**.
`Opsware.ProvSuccess.Notify`	(Optional) Specifies the notification email address for HP Server Automation to use if provisioning is successful.
`Opsware.ProvSuccess.Owner`	(Optional) Specifies the HP Server Automation user to which to assign ownership if provisioning is successful, for example **`opswareadmin`**.

Custom Properties That Make HP Server Automation Software Jobs Available

Depending on how your fabric administrator configures HP Server Automation jobs for vCloud Automation Center integration, you might have a choice between making all software jobs available to machine requesters to select, or you can specify specific jobs to apply to all machines provisioned from your blueprint. The Custom Properties to Make Software Jobs Available table describes these properties.

Table 2-24. Custom Properties to Make Software Jobs Available

Property	Definition
`LoadSoftware`	Set to **`True`** to make the software jobs listed in `Website\Software.txt` available to the user requesting the machine.
`VRM.Software.Id`	Optionally specify anHP 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 3

This section lists and described the vCloud Automation Center custom properties organized by property name.

These properties are also grouped functionally in another section of this guide and in context-relevant guides in the vCloud Automation Center product documentation set.

This chapter includes the following topics:

Custom Properties A Table

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

Table 3-1. Custom Properties A Table

Property	Description	Relevance
`AD.Lookup.Department`	Specifies the cost center value that is included in email sent to approvers to notify them of pending approvals.	
`Amazon.ElasticLoadBalancer.Names`	Assigns machines that are provisioned by a blueprint to the elastic load balancers that match the specified values.	
`Amazon.Instance.Id`	Specifies the Amazon instance ID of a machine provisioned on an Amazon EC2 endpoint.	

Custom Properties B Table

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

Table 3-2. Custom Properties B Table

Property	Definition	Relevance
`BMC.Service.Profile`	Specifies the name of the default authentication profile on the BMC BladeLogic server.	
`BMC.AddServer.Delay`	Specifies the number of seconds to wait before adding the machine to BMC BladeLogic. 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.	
`BMC.Software.Install`	Set to **`True`** to enable BMC BladeLogic integration.	
`BMC.Software.BatchLocation`	Specifies the location in BMC BladeLogic of software jobs to be 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.	

Custom Properties C Table

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

Table 3-3. Custom Properties C Table

Property	Definition	Relevance
`Cisco.Organization.DN`	The distinguished name of the Cisco UCS Manager organization in which Cisco USC machines provisioned by the provisioning group are placed, for example org-root/org-Engineering. If the specified organization does not exist in the Cisco UCS Manager instance managing the machine, provisioning fails. This property is available for provisioning groups only.	
`CloneFrom`	Specifies the name of an existing machine or virtualization platform object to clone from, for example a template in vCenter Server. The default is the value specified by the **Clone from** setting on the blueprint's **Build Information** tab.	

Table 3-3. Custom Properties C Table (Continued)

Property	Definition	Relevance
`CloneSpec`	Specifies a cloned machine, for example a predefined `SysPrep` object in vCenter Server. The default is the value specified by the customization setting on the blueprint's **Build Information** tab.	
`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 custom properties that begin with the letter E.

Table 3-4. Custom Properties E Table

Property	Definition	Relevance
`EPI.Server.Name`	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. 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. If only dedicated Opsware EPI agents for specific HP server automation servers were installed, this value must exactly match the server name configured for one of these agents. If at least one general EPI agent of the appropriate type (`VirtualMachine.EPI.Type`) 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.	
`EPI.Server.Port`	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.	
`EPI.Server.VDiskName`	Specifies the name of the Citrix provisioning vDisk to provision from.	
`EPI.Server.Store`	Specifies the name of the Citrix provisioning store that contains the vDisk identified by the `EPI.Server.VDiskName` property.	

Table 3-4. Custom Properties E Table (Continued)

Property	Definition	Relevance
`EPI.Server.Collection`	Specifies the name of the Citrix provisioning collection to which the machine will be registered.	
`EPI.Server.Site`	Specifies the name of the Citrix provisioning site that contains the collection and store identified by the `EPI.Server.Collection` and `EPI.Server.Store` properties.	

Custom Properties H Table

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

Table 3-5. Custom Properties E Table

Property	Definition	Relevance
`Hyperv.Network.Type`	Specifies the network adapter type of the virtual machine. This property is valid for use with Hyper-V only. The following values are available: ■ Synthetic (default) ■ Legacy. This value is not available with Windows XP or Windows Server 2003 64-bit guest operating systems.	
`Hostname`	Specifies the machine name, overriding the generated machine name contained in the `VirtualMachine.Admin.Name` property. If `Hostname` is not used, the `VirtualMachine.Admin.Name` value is used as the machine name.	

Custom Properties I Table

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

Table 3-6. Custom Properties I Table

Property	
`Image.ISO.Location`	Specifies the location of the ISO image to boot from. The format of this value depends on your platform. For details, see the documentation provided by your platform. This property is required for WIM-based provisioning, Linux kickstart and autoYaST provisioning, and SCCM-based provisioning. 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. For physical provisioning, this specifies the HTTP URL of the web-accessible location of the image.
`Image.ISO.Name`	Specifies the name of the ISO image from which to boot. The format of this value depends on your platform. For details, see the documentation provided by your platform. This property is required for WIM-based provisioning, Linux kickstart and autoYaST provisioning, and SCCM-based provisioning. For virtual provisioning with vCenter Server, this svalue specifies the path to the image in the datastore specified by `Image.ISO.Location`, for example */MyISOs*`/Microsoft/MSDN/win2003.iso`. 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 `Image.ISO.Location`. In virtual provisioning with Hyper-V, this value specifies the full local path to the image. For physical provisioning, this value specifies the file name of the image.
`Image.ISO.UserName`	Specifies the user name to access the CIFS share in the format *username@domain*. 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 `Image.ISO.UserName` property. For Dell iDRAC integrations where the image is located on a CIFS share that requires authentication to access.
`Image.WIM.Path`	Specifies the UNC path to the WIM file from which an image is extracted during WIM-based provisioning, for example `\\server\share$`.
`Image.WIM.Name`	Specifies the name of the WIM file located by the `Image.WIM.Path` property.
`Image.WIM.Index`	Specifies the index used to extract the desired image from the WIM file.
`Image.Network.User`	Specifies the user name with which to map the WIM image path (`Image.WIM.Path`) 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 `Image.Network.User` property.
`Image.Network.Letter`	Specifies the drive letter to which the WIM image path is mapped on the provisioned machine. The default value is K.

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

Property	
`Infrastructure.Admin.MachineObjectOU`	Specifies the organizational unit (OU) of the machine. When machines are placed in the required OU by the provisioning group OU setting, this property is not required.
`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.
`Infrastructure.Admin.DefaultDomain`	Specifies the default domain on the machine.
`Infrastructure.Admin.ADUser`	Specifies the user name that the machine uses to query Active Directory users and groups when an anonymous bind cannot be used.
`Infrastructure.Admin.ADPassword`	Specifies the password associated with the `Infrastructure.Admin.ADUser` user name.

Custom Properties L Table

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

Table 3-7. Custom Properties L Table

Property	Description	Relevance
`Linux.ExternalScript.Name`	Specifies the name of a customization script that the Linux 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.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.Server`	Specifies the name of the NFS server on which the Linux external customization script named in `Linux.ExternalScript.Name` is located.	
`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.	
`LoadSoftware`	Set to **`True`** to make the software jobs listed in `Website\Software.txt` available to the user requesting the machine.	

Custom Properties M Table

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

Table 3-8. Custom Properties M Table

Property	Description	Relevance
`MaximumProvisionedMachines`	Specifies the maximum number of linked clones for one machine snapshot. The default is 20.	
`Machine.SSH`	Set to **`True`** to enable the **Connect Using SSH** option 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.	
`MaximumProvisionedMachines`	Specifies the number of machines that you can provision across all users based on a given blueprint. Most blueprint types are unlimited by default. Linked clone blueprints have a default maximum of 20 machines.	

Custom Properties O Table

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

Table 3-9. Custom Properties O Table

Property	Description	Relevance
`Opsware.Software.Install`	Set to **`True`** to allow HP Server Automation to install software.	
`Opsware.Server.Name`	Specifies the fully qualified name of the server automation server.	
`Opsware.Server.Username`	Specifies the user name provided when a password file in the agent directory was created. This user name requires administrative access to the HP Server Automation instance, for example **`opswareadmin`**.	
`Opsware.BootImage.Name`	Specifies the boot image value as defined in HP Server Automation, for example **`winpe32`** for the 32-bit WinPE image. The property is not required when provisioning by cloning.	
`Opsware.Customer.Name`	Specifies a customer name value as defined in HP Server Automation, for example **`MyCompanyName`**.	
`Opsware.Facility.Name`	Specifies a facility name value as defined in HP server automation, for example **`Cambridge`**.	
`Opsware.Machine.Password`	Specifies the default local administrator password for an operating system sequence WIM image such as `Opsware.OSSequence.Name` as defined in HP Server Automation, for example **`P@ssword1`**.	
`Opsware.OSSequence.Name`	Specifies the operating system sequence name value as defined in HP Server Automation, for example **`Windows 2003 WIM`**.	
`Opsware.Realm.Name`	Specifies the realm name value as defined in HP Server Automation, for example **`Production`**.	
`Opsware.Register.Timeout`	Specifies the time, in seconds, to wait for creation of a provisioning job to complete.	

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

Property	Description	Relevance
`Opsware.WOL.Enabled`	Set to **`False`** for virtual provisioning.	
`Opsware.WOL.Delay`	Specifies the time, in seconds, to wait before running Wake-On-LAN. This setting is ignored if `Opsware.WOL.Enabled` is set to **`False`**.	
`Opsware.ProvFail.Notify`	(Optional) Specifies the notification email address for HP Server Automation to use in the event of provisioning failure.	
`Opsware.ProvFail.Owner`	(Optional) Specifies the HP Server Automation user to assign ownership to if provisioning fails, for example **`opswareadmin`**.	
`Opsware.ProvSuccess.Notify`	(Optional) Specifies the notification email address for HP Server Automation to use if provisioning is successful.	
`Opsware.ProvSuccess.Owner`	(Optional) Specifies the HP Server Automation user to which to assign ownership if provisioning is successful, for example **`opswareadmin`**.	

Custom Properties P Table

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

Table 3-10. Custom Properties P Table

Property	Description	Relevance
`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.UserName`	Specifies an Active Directory account user name with sufficient privileges to delete, disable, rename, or move Active Directory accounts. The value must be in *domain\username* format. This property is required if the vCloud Automation Center 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 `Plugin.AdMachineCleanup.UserName` property. For security, select the **Encrypt** check box in the user interface.	
`Plugin.AdMachineCleanup.Delete`	Set to **`True`** to delete the accounts of destroyed machines, instead of disabling them.	
`Plugin.AdMachineCleanup.Domain`	Specifies the Active Directory domain name that contains the machine account to be 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 *ou=OU, dc=dc* format .	
`Plugin.AdMachineCleanup.RenamePrefix`	Renames the accounts of destroyed machines by adding a prefix. The value is the prefix to prepend, for example **`destroyed_`**.	

Table 3-10. Custom Properties P Table (Continued)

Property	Description	Relevance
Pxe.Setup.ScriptName	Specifies a custom EPI PowerShell script to run on the machine before it is booted using the PXE network boot program. The value is the name assigned to the script when it is uploaded to the model manager.	
Pxe.Clean.ScriptName	Specifies the name of a EPI PowerShell script installed in the vCloud Automation Center 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.	

Custom Properties R Table

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

Table 3-11. Custom Properties R Table

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

Custom Properties S Table

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

Table 3-12. Custom Properties S Table

Property	Description	Relevance
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.	

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

Property	Description	Relevance
`SysPrep.`*Section.Key*	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. *Section* represents the name of the section of the SysPrep answer file, for example **`GuiUnattended`** or **`UserData`**. *Key* 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`**. For a full list of sections, keys, and accepted values, see the System Preparation Utility for Windows documentation. The following *Section.Key* combinations can be specified for WIM-based provisioning: ■ GuiUnattended ■ AdminPassword ■ EncryptedAdminPassword ■ TimeZone ■ UserData ■ ProductKey ■ FullName ■ ComputerName ■ OrgName ■ Identification ■ DomainAdmin ■ DomainAdminPassword ■ JoinDomain ■ JoinWorkgroup	
`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.	
`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 `SCCM.Server.UserName` property.	

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

Property	Description	Relevance
`SCCM.CustomVariable.`*Name*	Specifies the value of a custom variable, where *Name* 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.	
`Snapshot.Policy.AgeLimit`	Sets the age limit, in days, for snapshots that can be applied to machines.	
`Snapshot.Policy.Limit`	Sets the depth limit of snapshots that can be created for machines.	

Custom Properties V Table

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

Table 3-13. Custom Properties V Table

Property	Description	Relevance
`VirtualMachine.Admin.UseGuestAgent`	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. Set to **`False`** to disable the guest agent.	
`VirtualMachine.Admin.NameCompletion`	Specifies the domain name to include in the fully qualified domain name of the machine that the RDP files generate for the **Connect Using RDP** option. For example, set the value to **`myCompany.com`** to generate the FDQN *`my-machine-name`*`.myCompany.com` in the RDP file.	
`VirtualMachine.Admin.ConnectionAddress`	Specifies the RDP connection address of the machine to which an RDP file will be downloaded when the **Connect Using RDP** option is used or attached to automatic emails. Do not use in a blueprint or build profile unless **Prompt User** is enabled and no default value is supplied.	
`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.	
`VirtualMachine.Admin.CustomizeGuestOSDelay`	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.	

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

Property	Description	Relevance
`VirtualMachine.Admin.NetworkInterfaceType`	Indicates the network interface type. Use to modify the network settings of a newly provisioned virtual machine. The following options are available: ■ E1000 (default) ■ VirtIO ■ RTL8139 ■ RTL8139 VirtIO This property is for virtual provisioning.	
`VirtualMachine.Admin.OverrideVncConsole`	Specifies how the consoles of virtual machines are accessed. The default display type is VNC. Set to **`True`** to use the KVM (RHEV) display type, Spice. This property is available for KVM (RHEV) blueprints	
`VirtualMachine.Admin.Name`	Specifies the generated machine name. You can use this property can be used when creating custom workflows or plug-ins. The value in the blueprint or build profile has no effect on this property.	
`VirtualMachine.Admin.UUID`	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 build profile has no effect on this property.	
`VirtualMachine.Admin.AgentID`	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 build profile has no effect on this property.	
`VirtualMachine.Admin.Owner`	Specifies the user name of the machine owner.	
`VirtualMachine.Admin.Approver`	Specifies the user name of the group manager who approved the machine request.	
`VirtualMachine.Admin.Description`	Specifies the description of the machine as entered or modified by its owner or an administrator.	
`VirtualMachine.Admin.AdministratorEmail`	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 JoeAdmin@VMware.com,WeiMgr@VMware.com.	
`VirtualMachine.Host.TpmEnabled`	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`**. All hosts in a cluster must have a Trust Protection Module (TPM) device installed. If no acceptable hosts or clusters are found, the machine cannot be provisioned until this property is removed.	
`VirtualMachine.Admin.Hostname`	Specifies the name of the compute resource on which the machine resides.	
`VirtualMachine.Admin.ClusterName`	Specifies the name of the cluster that contains the compute resource on which the machine resides.	
`VirtualMachine.Admin.ApplicationID`	List the application IDs that can be assigned to a machine.	

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

Property	Description	Relevance
`VirtualMachine.Admin.AddOwnerToAdmins`	Set to **`True`** to add the machine's owner, as specified by the `VirtualMachine.Admin.Owner` property, to the local administrators group on the machine. This property is not available for provisioning by cloning.	
`VirtualMachine.Admin.AllowLogin`	Set to **`True`** (default) to add to the local remote desktop users group the machine's owner, as specified by the `VirtualMachine.Admin.Owner` property.	
`VirtualMachine.Agent.CopyToDisk`	Set to **`True`** (default) to copy the guest agent executable file to `%SystemDrive%\VRM\Build\Bin` on the machine's disk.	
`VirtualMachine.Agent.GuiRunOnce`	Set to **`False`** for the Linux agent to stop the provisioning workflow. This property is for kickstart provisioning with Linux or AutoYaST.	
`VirtualMachine.Agent.Reboot`	Set to **`True`** (default) to specify that the guest agent restarts the machine following installation of the guest operating system.	
`VirtualMachine.CDROM.Attach`	Set to **`False`** to provision the machine without a CD-ROM device. The default is **`True`**.	
`VirtualMachine.Customize.WaitComplete`	Set to **`True`** to hold the provisioning workflow until customizations are finished.	
`VirtualMachine.Admin.DiskInterfaceType`	Indicates the type of disk drivers. The following disk drivers are supported: ■ **`IDE`** (default) ■ **`VirtIO`** This property is for virtual provisioning.	
`VirtualMachine.Disk`*N*`.Letter`	Specifies the 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 `VirtualMachine.Disk1.Letter` and enter the value `D`. Disk numbering must be sequential.	
`VirtualMachine.Disk`*N*`.Size`	Defines the size in GB of disk N. For example, to give a size of 150GB to a disk G, define the custom property `VirtualMachine.Disk0.Size` and enter a value of **`150`**. Disk numbering must be sequential. By default a machine has one disk referred to by `VirtualMachine.Disk0.Size`, where size is specified by the **Storage** option on the **Build Information** tab of the blueprint from which the machine is provisioned. The value in the **Storage** option on the **Build Information** tab overwrites the value in the `VirtualMachine.Disk0.Size` property. The `VirtualMachine.Disk0.Size` property is not available as a custom property because of its relationship with the**Storage** option on the **Build Information** tab. More disks can be added by specifying `VirtualMachine.Disk1.Size`, `VirtualMachine.Disk2.Size` and so on. `VirtualMachine.Admin.TotalDiskUsage` always represents the total of the `.DiskN.Size` properties plus the `VMware.Memory.Reservation` size allocation.	

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

Property	Description	Relevance
`VirtualMachine.Disk`*N*`.Label`	Specifies the label of a machine's disk . The disk label maximum is 32 characters. Disk numbering must be sequential.	
`VirtualMachine.Disk`*N*`.Active`	Set to **`True`** (default) to specify that the machine's disk *N* is active. Set to **`False`** to specify that the machine's disk *N* is not active.	
`VirtualMachine.Disk`*N*`.FS`	Specifies the file system of the machine's disk *N*. The options are NTFS (default), FAT and FAT32.	
`VirtualMachine.Disk`*N*`.Percent`	Specifies the percentage of the disk *N* to be formatted by a guest agent for the machine's use. That machine cannot use the remaining portion of the disk .	
`VirtualMachine.Disk`*N*`.StorageReservationPolicy`	Specifies a storage reservation policy to find storage for disk *N*. Also assigns a storage reservation policy to a volume. You can use this disk property only with linked clone blueprints. To use this property, substitute the volume number for *N* in the property name and specify a storage reservation policy as the value. This property is equivalent to the **Storage Reservation Policy** drop-down menu in the Volumes list on the **Build Information** tab of the blueprint page. Disk numbering must be sequential.	
`VirtualMachine.Disk`*N*`.StorageReservationPolicyMode`	Allocates disk *N* on the best available storage reservation policy.	
`VirtualMachine.Rdp.File`	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.Setting`*N*. The file must be located in *`vCAC_installation_dir`*`\`*`Website`*`\Rdp` where *vCAC_install_dir* is the server install directory, for example `%SystemDrive%\Program Files x86\VMware\vCAC\Server\Rdp\console.rdp`. You must create the `Rdp` directory.	
`VirtualMachine.CPU.Count`	Specifies the number of CPUs allocated to a machine. The default is the value specified by the **# CPUs** setting on the blueprint's **Build Information** tab.	
`VirtualMachine.Memory.Size`	Specifies the size of the machine's memory in MB. The default is the value specified by the **Memory** option on the blueprint's **Build Information** tab.	
`VirtualMachine.Admin.TotalDiskUsage`	Specifies the total disk space that the machine uses, including all disks as specified by the `VirtualMachine.DiskN.Size` properties and the swap file as specified by the `VMware.Memory.Reservation` property.	
`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.	

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

Property	Description	Relevance
`VirtualMachine.Disk`*N*`.Storage`	Specifies that datastore on which to place the machine disk *N*. This property is also used to add a single datastore to a linked clone blueprint. *N* is the index (starting at 0) of the volume to assign. Type the name of the datastore to assign to the volume in the **Value** text box. This is the datastore name as it appears in the Storage Path column on the Edit Compute Resource page. Disk numbering must be sequential.	
`VirtualMachine.Disk`*N*`.VMwareType`	Specifies the VMware disk mode of the machine's disk*N*. The following options are available: ■ persistent ■ independent_persistent ■ independent_nonpersistent For details, see `VirtualDeviceDeviceBackingOption` data object help.	
`VirtualMachine.EPI.Type`	Specifies the type of external provisioning infrastructure. Set to **`BMC`** for BMC BladeLogic integration. Set to **`CitrixProvisioning`** for Citrix provisioning server integration.	
`VirtualMachine.Network`*N*`.Address`	Specifies the IP address of a network device `N` in a machine provisioned with a static IP address.	
`VirtualMachine.Network`*N*`.MacAddressType`	Indicates whether the MAC address of network device *N* is auto-generated or user-defined. This property is available for cloning. The default value is **`generated`**. If the value is **`static`**, you must also use `VirtualMachine.Network`*N*`.MacAddress` to specify the MAC address.	
`VirtualMachine.Network`*N*`.MacAddress`	Specifies the MAC address of a network device *N*. This property is available for cloning. If the value of `VirtualMachine.Network`*N*`.MacAddressType` is **`generated`**, this property contains the generated address. If the value of `VirtualMachine.Network.`*N*`.MacAddressType` is **`static`**, this poperty 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.	
`VirtualMachine.Network`*N*`.Name`	Specifies the network to which a network device *N* in a virtual machine is attached. By default, a network is assigned from the network paths available on the reservation on which the machine is provisioned. 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.	
`VirtualMachine.Network`*N*`.PortID`	Specifies the port ID to use for network device *N* when using a dvPort group with a vSphere distributed switch.	

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

Property	Description	Relevance
`VirtualMachine.Network`*N*`.ProfileName`	Specifies the name of a network profile from which to assign a static IP address to network device *N*or from which to obtain the range of static IP addresses that can be assigned to network device *N* of a cloned machine, where *N*=0 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 `VirtualMachine.Network`*N*`.Name` custom property. With WIM-based provisioning for physical machines, you must specify the network profile with the `VirtualMachine.Network`*N*`.ProfileName` custom property. You must also specify a value with the `VirtualMachine.Network`*N*`.Name` custom property, although vCloud Automation Center does not use the value. The following attributes of the network profile are available to enable static IP assignment in a cloning blueprint: ■ VirtualMachine.Network*N*.SubnetMask ■ VirtualMachine.Network*N*.Gateway ■ VirtualMachine.Network*N*.PrimaryDns ■ VirtualMachine.Network*N*.SecondaryDns ■ VirtualMachine.Network*N*.PrimaryWins ■ VirtualMachine.Network*N*.SecondaryWins ■ VirtualMachine.Network*N*.DnsSuffix ■ VirtualMachine.Network*N*.DnsSearchSuffixes	
■ `VirtualMachine.Network`*N*`.SubnetMask` ■ `VirtualMachine.Network`*N*`.Gateway` ■ `VirtualMachine.Network`*N*`.PrimaryDns` ■ `VirtualMachine.Network`*N*`.SecondaryDns` ■ `VirtualMachine.Network`*N*`.PrimaryWins` ■ `VirtualMachine.Network`*N*`.SecondaryWins` ■ `VirtualMachine.Network`*N*`.DnsSuffix` ■ `VirtualMachine.Network`*N*`.DnsSearchSuffixes`	Configures attributes of the network profile specified in `VirtualMachine.Network`*N*`.ProfileName`.	
`VirtualMachine.Rdp.Setting`*N*	Configures specific RDP settings. *N* 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 `VirtualMachine.Rdp.Setting1` and set the value **`authentication level:i:3`**. For a list of available settings and correct syntax, see the Microsoft Windows RDP documentation.	

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

Property	Description	Relevance
`VirtualMachine.Software0.ScriptPath`	Specifies the 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 `.bat` file. You can pass custom property values as parameters to the script by inserting {*CustomPropertyName*} in the path string. For example, if you have a custom property named `ActivationKey` whose value is **`1234`**, the script path is `D:\InstallApp.bat -key {ActivationKey}`. The guest agent runs the command `D:\InstallApp.bat -key 1234`. Your script file can then be programmed to accept and use this value.	
`VbScript.PreProvisioning.Name`	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.	
`VbScript.PostProvisioning.Name`	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.	
`VbScript.UnProvisioning.Name`	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.	
`VCloud.Template.MakeIdenticalCopy`	Set to **`True`** to clone an identical copy of the vApp template in vCloud Director and provision the results in vCloud Automation Center. 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. Set to **`False`** to clone a copy of the vApp template with settings specified by the vApp and vApp component blueprints. The storage path specified in a vApp component blueprint, operating system or when requesting a vApp machine, is used if the **Make Identical Copy** option in the vApp template properties was selected.	

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

Property	Description	Relevance
`VCNS.SecurityGroup.Names.`*name*	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. 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 might list security groups intended for general use, for the sales force, and for support: ■ VCNS.SecurityGroup.Names ■ VCNS.SecurityGroup.Names.sales ■ VCNS.SecurityGroup.Names.spport	
`VCNS.LoadBalancerEdgePool.Names.`*name*	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 *edge*/*pool* name or a list of *edge*/*pool* names separated by commas. Names are case-sensitive. 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: ■ VCNS.LoadBalancerEdgePool.Names ■ VCNS.LoadBalancerEdgePool.Names.moderate ■ VCNS.LoadBalancerEdgePool.Names.high ■ VCNS.LoadBalancerEdgePool.Names.low	
`VMware.VirtualCenter.OperatingSystem`	Specifies the vCenter Server guest operating system version (`VirtualMachineGuestOsIdentifier`) 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 build profiles using one of several property sets, for example, `VMware[OS_Version]Properties`, that are predefined to include the correct `VMware.VirtualCenter.OperatingSystem` values. This property is for virtual provisioning. When this property has a non-Windows value, the **Connect Using RDP** option is disabled. The property can be used in a virtual, cloud or physical blueprint. For a list of currently accepted values, see the VMware® vCenter Server™ documentation.	

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

Property	Description	Relevance
`VMware.SCSI.Type`	For vApps, specifies the SCSI machine type using one of the following case-sensitive values: ■ **`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. ■ **`VirtualSCSI`** Use para-virtualization emulation for the virtual disk. ■ **`none`** Use if a SCSI controller does not exist for this machine.	
`VMware.SCSI.Type`	For vSphere only, specifies the SCSI machine type using one of the following case-sensitive values: ■ **`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 `VirtualSCSISharing` ENUM value.	
`VMware.Memory.Reservation`	Specifies the size of the machine's swap file.	
`VMware.VirtualCenter.Folder`	Name of the inventory folder in the datacenter in which the machine is created. The default is **`VRM`**. Can be a path with multiple folders, for example production\email servers\. A proxy agent creates the folder in vCenter Server if it does not exist. VMware and vSphere only	

`VRM.Software.Id`*NNNN*	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`	

`VRM.Software.Id`*NNNN*	Optionally specify anHP Server Automation policy to be applied to all machines provisioned from the blueprint. NNNN is a number from 1000 to 1999.	

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

Property	Description	Relevance
`VRM.DataCenter.Policy`	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. 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. 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.	
`VMware.Network.Type`	Specifies a network adapter type. The following adapter type values are available: ■ **`Flexible`** (default) ■ **`VirtualPcNet32`**. This type is not compatible with vSphere ■ **`E1000`** or **`VirtualE1000`** ■ **`VMXNET`** or **`VirtualVMXNET`** ■ **`VMXNET2`** ■ **`VMXNET3`** 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 in physical provisioning.	
`VirtualMachine.VDI.Type`	Specifies the type of virtual desktop infrastructure. For XenDesktop. provisioning, set to **`XenDesktop`**.	
`VDI.Server.Website`	Specifies the server name of the Citrix Web interface site to use in connecting to the machine. If the value of `VDI.Server.Name` 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 `VDI.Server.Name` property determines the desktop delivery controller to connect to, which must be the name of a server that hosts a desktop delivery controller.	

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

Property	Description	Relevance
VDI.Server.Name	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. If the value is a 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. 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.	
VDI.Server.Group	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 *group_name;catalog_name* format. For XenDesktop 4, specifies the name of the XenDesktop group to which machines are to be added. XenDesktop 4 preassigned groups are supported.	
VDI.ActiveDirectory.Interval	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).	
VDI.ActiveDirectory.Interval	Specifies an optional timeout value in time span format for virtual desktop infrastructure machine Active Directory registration check. The default value is 00:30:00 (30 minutes.)	
VDI.ActiveDirectory.Delay	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).	

Custom Properties Z Table

This section lists custom properties that begin with the letter Z.

Table 3-14. Custom Properties Z Table

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

Using the Properties Dictionary 4

The property dictionary enables you to define characteristics of custom properties to tailor their display in the user interface.

You can customize property display in the user interface, as in the following examples:

- 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.

Create and deploy a new custom property definition using this high-level procedure:

1. Add a new property and associate it with a property control type.
2. Add any 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 by using the Properties tab on the blueprint form. Alternatively you could add the properties to a build profile. Note that a tenant administrator or business group manager role can create or edit blueprints. See the applicable IaaS Configuration guide in VMware vCloud® Automation Center™Product Documentation for information about adding properties to blueprints and build profiles.
4. Optionally add a control layout to specify where the properties appear on the machine confirmation page.

New custom properties are typically used by custom workflows that specify the logic to execute based on the value of that property. You can define these custom workflows using either vCloud Automation Center Designer. For more information about working with custom workflows, see *Extension Development*.

This chapter includes the following topics:

Adding Property Definitions

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

Several property control types are available as you create new properties. After you create a property you can optionally add attributes to it property, based on its control type. For example, a DropDown control type can be assigned a ValueExpression attribute, which when selected, will open 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.

The following property definition control types are available.

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 have also been 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

Use this procedure for creating a new property definition.

After you add a new property definition, and select a definition control type for that property, you can optionally add attributes to configure additional 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 vCloud Automation Center console as a **fabric administrator**.

Procedure

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

2 Click **New Property Definition**.

3 Type 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 Type the name that will appear in the user interface in the **Display Name** text box.

5 (Optional) Type 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. The following control types are available:

- **CheckBox**

  Associate a check box with the new property.

- **DateTimeEdit**

  Add a date and time control type that adheres to a *YYYY-MM-DD* or *MM/DD/YYYY* format and a time in *HH:MM* format (24-hour clock or followed by AM or PM). You can further refine the allowed value by adding property attributes such as MinValue and HelpText.

- **DropDown**

  This control type allows the user to type a value in a text box. You can further refine this property type by adding attributes such as ValueList and Relationship.

- **DropDownList**

  This control type requires the user to select from values in a drop-down menu. You can further refine this property type by adding attributes such as ValueList and Relationship.

- **Integer**

  This control type allows for an integer value. You can further refine the allowed value by adding attributes such as MinValue and MaxValue or Integer.

- **Label**

  You can add a read-only text label. You can further refine the allowed value by adding attributes such as MinValue and HelpText.

- **Link**

  You can display 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 OrderIndex attribute.

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

Several property attribute types are available to further enhance property definitions in the dictionary.

Property attributes types are available for a custom property definition based on the control type used when the property was defined.

The following table shows which attribute values are available for each of the property definition control types.

Table 4-2. Property attributes available for property definition control types

Property Definition Control Type	Available Property Attribute Types
CheckBox	■ HelpText ■ OrderIndex
DateTimeEdit	■ HelpText ■ MinValue ■ OrderIndex
DropDown	■ HelpText ■ OrderIndex ■ Relationship ■ ValueExpression ■ ValueList
DropDownList	■ HelpText ■ OrderIndex ■ Relationship ■ ValueExpression ■ ValueList
Integer	■ HelpText ■ Interval ■ MaxValue ■ MinValue ■ OrderIndex
Label	■ HelpText ■ OrderIndex
Link	OrderIndex

Add a Help Text Attribute

Add help text that is displayed when the user hovers over the property name.

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

Prerequisites

Log in to the vCloud Automation Center 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 Type 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

Add an order index attribute to control how the property name is displayed in the user interface.

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

Prerequisites

Log in to the vCloud Automation Center 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 Type 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 ordering 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

Add a relationship attribute to define a relationship between a drop-down property and another property such that the value of one 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 depending on that parent property value.

Prerequisites

Log in to the vCloud Automation Center 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 Type an attribute name in the **Name** text box.

This name is not visible in the user interface.

6 Type the needed value or value expression in the **Value** text box. For more information, see "Example - Creating a Relationship Between Two Properties and Adding to a Blueprint," on page 60.

7 Click the **Save** icon ( ).

8 (Optional) Add additional attributes.

9 Click **OK**.

Add a Value Expression Attribute

Add a value expressions attribute. A value expression is an XML string that maps the values of the parent property and the child property in a property relationship.

A ValueExpression attribute is available for the DropDown and DropDownList property value control types.

Prerequisites

Log in to the vCloud Automation Center 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 Type an attribute name in the **Name** text box.

This name is not visible in the user interface.

6 Type the values that you want to use in the **Value** text box, for example an XML value expression or snippet that you have formatted as a single line string with no line breaks. For more information, see "Example - Creating a Relationship Between Two Properties and Adding to a Blueprint," on page 60.

7 Click the **Save** icon ( ).

8 (Optional) Add additional attributes.

9 Click **OK**.

Add a Value List Attribute

Add a value list attribute.

A ValueList attribute is available for the DropDown and DropDownList property value control types.

Prerequisites

Log in to the vCloud Automation Center 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 Type an attribute name in the **Name** text box.

This name is not visible in the user interface.

6 Click in the **Value** text box and type a comma-separated list of values to appear in the drop-down list, for example `Option 1,Option 2,Option 3`. For more information, see "Example - Creating a Relationship Between Two Properties and Adding to a Blueprint," on page 60.

7 Click the **Save** icon ( ).

8 (Optional) Add additional attributes.

9 Click **OK**.

Add a Minimum Value Attribute

Add a minimum value attribute.

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

Prerequisites

Log in to the vCloud Automation Center 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 Type an attribute name in the **Name** text box.

This name is not visible in the user interface.

6 Click in the **Value** text box and then type the minimum value of the property. 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 (24-hour clock or 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

Add a maximum value attribute.

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

Prerequisites

Log in to the vCloud Automation Center 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 Type an attribute name in the **Name** text box.

This name is not visible in the user interface.

6 Click in the **Value** text box, and then type an integer for the maximum value of the property.

7 Click the **Save** icon ( ).

8 (Optional) Add additional attributes.

9 Click **OK**.

Add an Interval Attribute

Add an interval attribute. The interval is the 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 vCloud Automation Center 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 Type 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 - Creating a Relationship Between Two Properties and Adding to a Blueprint

This example describes how to define a relationship so that the value of a new parent property determines the child property values that are available in a drop-down list. The example illustrates the use of the ValueList, ValueExpression, and Relationship property attributes in combination with the DropDownList property definition control type.

In this example, a new `VirtualMachine.Network.Environment` property is used to filter 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.

The high-level overview process is as follows:

1 Create two property definitions, the parent and child.

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 build profile.

Instead of specifying a ValueList attribute for the child property, you could also 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. Both attribute types are used in this example.

This example is a logical progression from the following topics:

- "Add a Property Definition," on page 54
- "Add a Relationship Attribute," on page 57
- "Add a Value List Attribute," on page 58
- "Add a Value Expression Attribute," on page 58

Table 4-3. Relationship between `VirtualMachine.Network.Environment` property and `VirtualMachine.Network0.Name` property menu options

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

This example assumes you are logged in as a tenant administrator and are on the **Infrastructure > Blueprints > Property Dictionary** page.

1 Create the parent property.

   a Click **New Property Definition** on the Property Dictionary page.

   b Type **`VirtualMachine.Network.Environment`** in the **Name** text box.

   c Type **`Environment`** in the **Display Name** text box.

   d Click in the **Control Type** box and select **DropDownList**.

   e Click the **Required** check box to make the property required at request time ad then click **Save.**

2 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 Type **`Values`** in the **Name** text box.

   e Type **`Development,Test,Production`** in the **Value** text box.

   f Click **Save** and click **OK**.

3 Create the child property.

   a Click **New Property Definition** on the Property Dictionary page.

b Type **`VirtualMachine.Network0.Name`** in the **Name** text box.

c Type **`Select Network`** in the **Display Name** text box.

d Click in the **Control Type** box and select **DropDownList**.

e Click **Save.**

4 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 Type **Parent** in the **Name** text box.

e Type **`VirtualMachine.Network.Environment`** in the **Value** text box.

f Click **Save** and click **OK**.

5 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>
    <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>
```

6 Add the value expression (XML file content) that you just created to the child property.

a In a text editor, format the value expression (XML file content) 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 Type **`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**.

7 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 Type **`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.

8 Click the **Prompt User** check box.

9 Click **OK**.

10 Click **New Property**.

11 Type **`VirtualMachine.Network0.Name`** in the **Name** text box.

12 Click **OK**.

When a user requests a machine using this blueprint, 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.

Control layouts enable you to group properties together and specify the order in which they are displayed to users on the machine confirmation page. You need to add both the control layout and the individual properties that it contains to the blueprint for them to appear. The layout only defines the order of the properties that display in the form; it does not include the properties by default.

Add a Property Control Layout

Create a property control layout to determine how properties are displayed to users.

Prerequisites

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

This task assumes that you have already created the new property definitions that you want to add to the new property control layout. After you have created the layout, a tenant administrator or business administrator can add it to a blueprint by using the Properties tab. In this example, the two existing property definition names are `VirtualMachine.Network.Environment` and `VirtualMachine.Network0.Name`.

Procedure

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

2 Click **New Property Layout**.

3 Type 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 named NetworkLayout is now available for a tenant administrator or business group manager to add to a blueprint or build profile. The name to specify on the Properties tab page on the blueprint is NetworkLayout.

Index

W