

# Embotics® vCommander® Veeam Integration Guide

vCommander version 5.7 and higher

Veeam version 1.1

## Contents

Introduction .....	3
Version support and compatibility .....	3
See also .....	3
Prerequisites .....	4
Accounts and permissions .....	4
Installation .....	4
1. Download the PowerShell modules .....	4
2. Specify that the downloaded files are "trusted" content .....	5
3. Install the PowerShell modules .....	5
4. Verify the module installation .....	5
5. Set permissions on folders .....	6
6. Enable PowerShell Remoting .....	6
7. Verify installation of the Veeam PowerShell Snap-in .....	7
Encrypt your authentication credentials .....	7
Constructing the Execute Script Step Command Line .....	8
Example command line .....	9
Script step command line syntax - 32-bit Windows .....	9
Script step command line syntax - 64-bit Windows .....	9
Setting the execution policy in the command line .....	9
Loading the modules in the command line .....	9
Using variables in the command line .....	10
Walk-through: Backing up a VM through a Command Workflow .....	10
Create custom attributes (optional) .....	10
Create a command workflow .....	11
Test the integration .....	12
Walk-through: Adding VMs to a Backup Job During Provisioning .....	14
Create a completion workflow .....	14
Test the integration .....	16
Walk-through: Decommissioning Workflow .....	16
Create a decommissioning workflow .....	16
Test the integration .....	18
Command Reference .....	19
Getting help .....	19

---

Add-VMToJob.....	19
Backup-VM.....	21
Remove-VMFromJob .....	24
Start-QuickBackup .....	25
Troubleshooting.....	27

## Introduction

If you use Veeam to back up your VMs, the Embotix® vCommander® integration with Veeam allows you to:

- back up one or more VMs immediately
- specify that a VM be backed up regularly by adding the VM to a Veeam backup job
- run an on-demand incremental backup for a specific VM
- remove a VM from a Veeam backup job. VMs are added to an exclusion list and are later removed by the Veeam retention policy.

The Veeam integration provides access to Veeam Backup and Replication through PowerShell.

You can use command workflows, completion workflows, or both to integrate with Veeam:

- Command workflows can be run on-demand by both vCommander users and Service Portal users. They can also be scheduled to run later. For example, a Service Portal user can select a VM and schedule a backup to run overnight.
- Completion workflows run automatically, as part of the provisioning or decommissioning process. For example, you can specify that all new VMs are added to a Veeam backup job that is scheduled to run weekly. Or, you can back up a VM and then remove the VM from a Veeam backup job during decommissioning.

You can also create custom attributes to store Veeam-related metadata.

## Version support and compatibility

Version 1.1 of the Veeam integration supports Veeam Backup and Replication version 8.0.

VeeamAdapterClient module version	Required vCommander module version	Compatible vCommander versions
1.1	1.1	5.7 and higher
1.0	1.0	5.5.x, 5.2.x, 5.1.x, 5.0.x, 4.7.x and 4.6.x

## See also

Consult the *vCommander REST API v2 Getting Started Guide* for general information on the REST API v2, including basic use as well as information on security, authentication, credential encryption, exception and error handling, and the data model.

## Prerequisites

The Veeam integration requires the following:

- PowerShell 3 or 4. You must be proficient in the use of Windows PowerShell to integrate vCommander with Veeam.
- .Net framework 3.5 (part of the PowerShell installation)
- A Windows domain account for connecting to Veeam (see [Accounts and permissions](#) below for details)
- A vCommander user account for the REST API v2 (see [Accounts and permissions](#) below for details)
- Veeam Backup Solution version 8.0 installed either on the same domain as vCommander, or in a separate domain, with trust between the two domains
- The Veeam PowerShell Snap-in installed on the Veeam host
- The Virtual Center(s) hosting the VMs must be added to Veeam Backup and Replication

**Note:** Backup of templates and out-of-inventory VMs is not supported.

## Accounts and permissions

The Veeam integration requires the following accounts:

- **Windows domain account for connecting to Veeam:** This is the account used to access the Veeam host; it exists on the Veeam host itself. This must be a domain account (not a local account), and must have permissions to execute the Veeam PowerShell Snap-in cmdlets and functions.
- **vCommander account for updating custom attributes for backed-up VMs.** This is the account you see on the **Configuration > Users and Roles** page of vCommander. This account must have permission to set custom attributes (Infrastructure role of Administrator and all Operator roles), and visibility of these VMs. To verify visibility, in vCommander, click **Configuration > Users and Roles**. Select the user and ensure that under Access Rights in the right pane, the user has visibility of the VMs.

Note: As part of the installation process, you will [encrypt your authentication credentials](#).

## Installation

### 1. Download the PowerShell modules

You need to download the following modules:

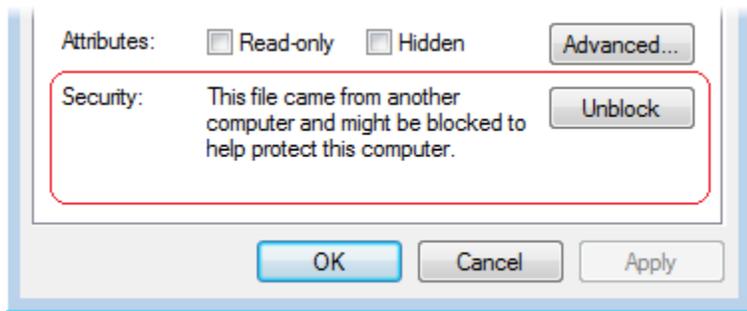
- vCommander REST Client module **version 2.4 or higher**. You can download the vCommander REST Client module here:  
<http://support.embotics.com/support/solutions/articles/8000035227-download-vcommander-rest-client>

See the *vCommander REST API v2 Getting Started Guide* for installation details.

- VCommander module **version 1.1** (used for credential encryption and handling of SSL certificates)  
**Note:** If you had previously installed version 1.0 of the VCommander module, you must replace it with version 1.1.
- VeeamAdapterClient module **version 1.1**  
You can download the VCommander and VeeamAdapterClient modules here:  
<http://ftp.embotics.com/downloads/kb/modules.zip>

## 2. Specify that the downloaded files are "trusted" content

Windows may block the use of files from external sources. To check the permissions on each downloaded .zip file, right-click it and select **Properties**. On the General tab of the Properties dialog, the following message indicates that the file is blocked:



To unblock the file, click **Unblock**.

**Note:** If you do not have full permissions on the folder where you saved the .zip file, the file will remain blocked. If this happens, move the .zip file to a location where you have full permissions (for example, your desktop), and unblock the .zip file there.

## 3. Install the PowerShell modules

Extract the three zip files into the PowerShell module directory of the machine where you will run the PowerShell client. See the *REST API v2 Getting Started Guide* for more information.

## 4. Verify the module installation

To verify the installation of the three modules, open PowerShell and enter the following command:

```
Get-Module -ListAvailable
```

You should see **VCommander**, **VCommanderRestClient** and **VeemAdapterClient** in the list of modules.

```
PS H:\> Get-Module -ListAvailable

Directory: C:\Windows\system32\WindowsPowerShell\v1.0\Modules

ModuleType Name                                     ExportedCommands
-----
Manifest AppLocker                                <Set-AppLockerPolicy, Get-AppLockerPolicy, Test-AppLockerPolicy, Get-...
Manifest BigHorn
Manifest BitsTransfer                          <Add-BitsFile, Remove-BitsTransfer, Complete-BitsTransfer, Get-BitsTr...
Manifest CimCmdlets                           <Get-CimAssociatedInstance, Get-CimClass, Get-CimInstance, Get-CimSes...
Script ISE                                       <New-IseSnippet, Import-IseSnippet, Get-IseSnippet>
Manifest Microsoft.PowerShell.Diagnostics      <Get-WinEvent, Get-Counter, Import-Counter, Export-Counter...>
Manifest Microsoft.PowerShell.Host            <Start-Transcript, Stop-Transcript>
Manifest Microsoft.PowerShell.Management      <Add-Content, Clear-Content, Clear-ItemProperty, Join-Path...>
Manifest Microsoft.PowerShell.Security        <Get-Acl, Set-Acl, Get-PfxCertificate, Get-Credential...>
Manifest Microsoft.PowerShell.Utility         <Format-List, Format-Custom, Format-Table, Format-Wide...>
Manifest Microsoft.WSMan.Management           <Disable-WSManCredSSP, Enable-WSManCredSSP, Get-WSManCredSSP, Set-WSM...
Script PowerShellLogging                       <Disable-LogFile, Disable-OutputSubscriber, Enable-LogFile, Enable-Ou...
Script PSDiagnostics                           <Disable-PSTrace, Disable-PSWSManCombinedTrace, Disable-WSManTrace, E...
Binary PSScheduledJob                          <New-JobTrigger, Add-JobTrigger, Remove-JobTrigger, Get-JobTrigger...>
Manifest PSWorkflow                            <New-PSWorkflowExecutionOption, New-PSWorkflowSession, nwsn>
Manifest PowerShellUtility                     Invoke-AsWorkflow
Manifest TroubleshootingPack                  <Get-TroubleshootingPack, Invoke-TroubleshootingPack>
Binary VCommander                               <Set-UcmdrUmAttribute, New-EncryptCredential, New-DecryptCredential>
Manifest VCommanderRestClient                  <Initialize-Configuration, Set-BaseServiceURI, Get-BaseServiceURI, Co...
Script VeemAdapterClient                       <Set-UcmdrUmAttribute, Backup-UM, Add-UMJob, Remove-UMFromJob...>
```

You should see the following subdirectories and files in the PowerShell Modules directory:

```
VCommander\VCommander.dll
VCommander\Security.psml
VCommander\VCommander.psd1
VCommander\VCommander.psml
VeemAdapterClient\VeemAdapterClient.psd1
VeemAdapterClient\VeemAdapterClient.psml
...
```

## 5. Set permissions on folders

If you installed the PowerShell client on the vCommander host, you need to grant Read and Execute permissions for the VeemAdapterClient folder.

The folder where you store the scripts calling the Veem integration must have Read, Write and Execute permissions.

## 6. Enable PowerShell Remoting

To allow vCommander to execute remote commands on the Veem Backup host, you must enable PowerShell Remoting on the Veem Backup host. By default, PowerShell Remoting is disabled. You only need to do this once.

To enable remote PowerShell commands:

1. Launch PowerShell with administrative rights (run as administrator).
2. Run the following command:  

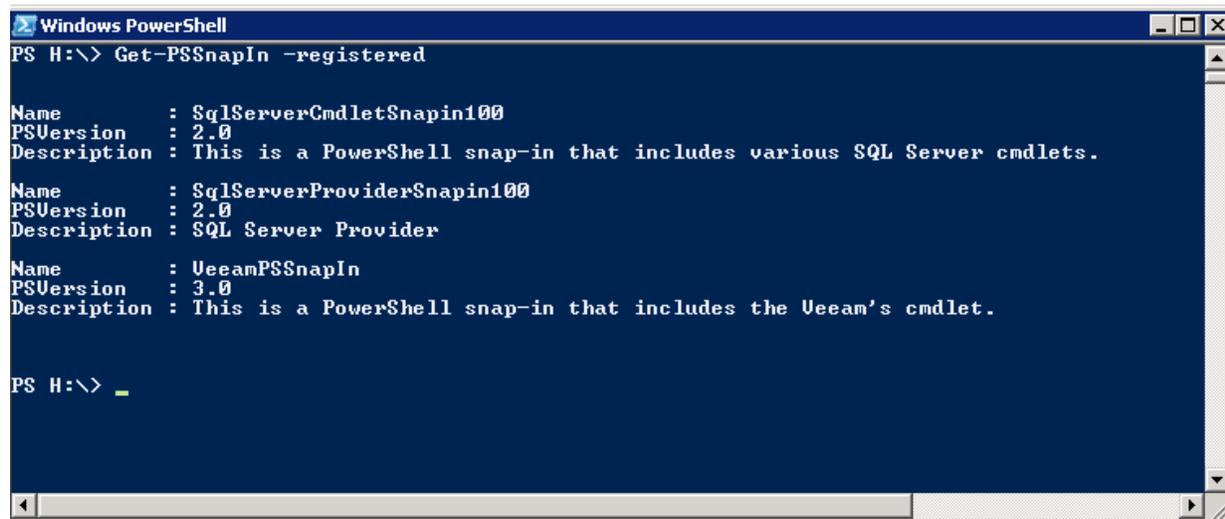
```
Enable-PSRemoting
```
3. Enter a ("Yes to All") when prompted.

## 7. Verify installation of the Veeam PowerShell Snap-in

To verify that the Veeam PowerShell snap-in is installed, in PowerShell, run this command:

```
Get-PSSnapIn -registered
```

You should see VeeamPSSnapIn in the list:



```
Windows PowerShell
PS H:\> Get-PSSnapIn -registered

Name       : SqlServerCmdletSnapin100
PSVersion  : 2.0
Description : This is a PowerShell snap-in that includes various SQL Server cmdlets.

Name       : SqlServerProviderSnapin100
PSVersion  : 2.0
Description : SQL Server Provider

Name       : VeeamPSSnapIn
PSVersion  : 3.0
Description : This is a PowerShell snap-in that includes the Veeam's cmdlet.

PS H:\> _
```

**Important:** Pay attention to the bit version of PowerShell. When vCommander executes commands remotely, it uses the default version of PowerShell. On 64-bit Windows, the default is 64-bit PowerShell; on 32-bit Windows, the default is 32-bit PowerShell. Veeam will install the snap-in for the 64-bit PowerShell on 64-bit Windows.

## Encrypt your authentication credentials

The Veeam integration uses two-stage encryption to protect credentials used in scripts. Passwords are first RSA-encrypted with a machine key. The encryption process produces an encrypted credential key file which can be stored on disk. During script executions, the Veeam integration decrypts this key and re-encrypts the password in a form that can be stored in memory. The second stage uses a standard Windows function to receive the password securely without storing it in memory as clear text.

You need to encrypt both the credentials used to log in to vCommander and the credentials used to log in to the Veeam host, so you must perform the procedure below twice. The VCommander module is used for encryption.

You will use these encrypted files to authenticate against the vCommander host and the Veeam host by adding the `-vCommanderCredential` and the `-VeeamCredential` parameters to your script, as shown in the walk-throughs below.

**Tip:** The best way to verify the user name and password for the Veeam account is to RDP into the Veeam host using this account. The best way to verify the user name and password for the vCommander account is to log into vCommander using this account.

To encrypt your credentials:

1. Open PowerShell as administrator.

2. Import the VCommander module into PowerShell:

```
import-module VCommander
```

3. If the user execution policy is set to Restricted, set it to Remote Signed with the following command:

```
Set-ExecutionPolicy RemoteSigned
```

4. Encrypt your credentials into a file for storage on disk:

```
New-EncryptCredential -destFilePath "<file-path>"
```

Make sure the destination directory exists. If the destination directory doesn't exist, an error will be thrown. If the destination file exists, it will be overwritten.

It's best practice to give this file an .xml extension.

For example:

```
New-EncryptCredential -destFilePath "C:\\Scripts\vcmdr_credential_key.xml"
```

5. Enter the username and password for the account.

The credentials are encrypted, and the resulting encrypted object is serialized into the specified file.

6. If you use separate credentials for vCommander and the Veeam host, run through the procedure again for the second set of credentials, specifying a different file path and the proper user name and password.

You will use these files to authenticate against the vCommander host and the Veeam host by adding the `-vCommanderCredential` and the `-VeeamCredential` parameters to your script, as shown below.

**Note:** Regenerate these files whenever the password changes.

## Constructing the Execute Script Step Command Line

This section provides important details on constructing the command line for the Execute Script steps that you add to the workflows. The walk-throughs that follow show you how to create the workflows from start to finish.

**Note:** Copying and pasting from a PDF may result in additional characters in your command line. Verify that your command line matches what you see below.

For more information on workflow script steps, see [Workflow Steps Reference](#) in the *vCommander User Guide*.

## Example command line

The following example backs up a VM using the 32-bit version of PowerShell on a 64-bit Windows machine. For command-line examples of adding a VM to a backup job and removing a VM from a backup job, see [Command Reference](#).

```
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy
RemoteSigned -Command "& { import-module veeamadapterclient; import-module vcommander;
```

```
#Add a -Debug switch to output debug logs
```

```
Backup-VM -computerName '#{system.address}' -vCommanderCredential
'C:\Scripts\vcmdr_credential_key.xml' -remoteComputerName 'myveeamhost' -
veeamCredential 'C:\Scripts\veeam_credential_key.xml' -hostName '#{target.host.name}'
-vmName '#{target.deployedName}' -vmRemoteId '#{target.remoteId}' -jobName
 '#{target.deployedName}-#{target.remoteId}-backup' -attribNameLastBackup 'Last Backup
Time' -attribNameJobName 'Backup Job Name' -attribNameNumRestorePoints 'Number of
Restore Points' }"
```

## Script step command line syntax - 32-bit Windows

```
C:\Windows\system32\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy
RemoteSigned -Command "& {cmdlet and its parameters}"
```

## Script step command line syntax - 64-bit Windows

For the 32-bit version of PowerShell:

```
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy
RemoteSigned -Command "& {cmdlet and its parameters}"
```

For the 64-bit version of PowerShell:

```
C:\Windows\system32\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy
RemoteSigned -Command "& {cmdlet and its parameters}"
```

## Setting the execution policy in the command line

The security settings built into PowerShell include the "execution policy", which determines whether and how PowerShell runs scripts. By default, PowerShell's execution policy is set to Restricted. You must specify the following switch in the workflow script step when calling PowerShell (as shown in the Example Script Step above):

```
-ExecutionPolicy RemoteSigned
```

## Loading the modules in the command line

You must load both the VeeamAdapterClient and the VCommander modules into PowerShell by including the following commands in your workflow script step (as shown in the Example Script Step above):

```
Import-Module VeeamAdapterClient
```

```
Import-Module VCommander
```

## Using variables in the command line

As shown in the [Example Comand Line](#) above, you can use vCommander variables in the command line. See [Command Reference](#) for a mapping of cmdlet parameters to variables.

## Walk-through: Backing up a VM through a Command Workflow

This walk-through shows you how to add a VM to a Veeam backup job as part of a completion workflow for a new VM request.

### Create custom attributes (optional)

For workflows in which you are backing up a VM, you can create custom attributes to store backup-related metadata.

In this step, create three custom attributes to contain the following backup-related metadata:

- the Veeam backup job name
- the last backup time
- the number of backups that have been performed on the VM (the number of restore points)

Values for these attributes are populated or updated if the cmdlet completes successfully.

1. Click **Configuration > Custom Attributes**, and click **Add**.
2. In the Configure Attribute wizard, enter a name for the attribute, such as "Backup Job Name". These custom attributes can be displayed on the Details panel of the VM's Summary tab, so you should create user-friendly names.
3. Specify that the custom attribute should apply to **Services**.

- Select the **Text** type and click **Finish**.

- Now create two more custom attributes in the same way:
  - the last backup time - we will use the name "Last Backup Time"
  - the number of backups that have been performed on the VM (the number of restore points) - we will use the name "Number of Restore Points"

For more information, see [Using Custom Attributes to Add Infrastructure Metadata](#) in the *vCommander User Guide*.

## Create a command workflow

To create a command workflow to back up VMs:

- Click **Configuration > Command Workflows**.
- Click **Add**.
- On the Name page, name the workflow "Back up VM Now".
- On the Steps page, click **Add > Execute Script**.
- Give the step a meaningful name, such as "Back up VM".
- Specify a timeout. The timeout depends on your environment. Make sure you allow enough time to complete a VM backup.

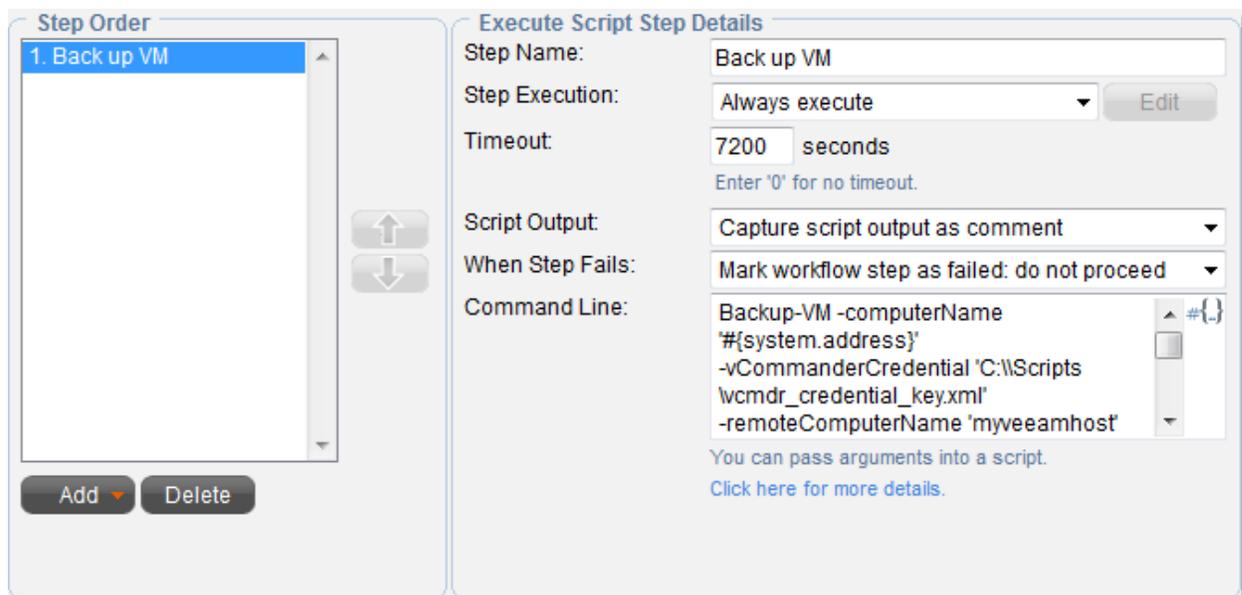
7. **IMPORTANT:** Specify that the script output should be captured as comment. Otherwise, you will not have any details in case of script failure.
8. Enter the following command line (this example uses a 32-bit version of PowerShell on 64-bit Windows):

```
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy RemoteSigned -Command "& { import-module veeamadapterclient; import-module vcommander;

#Add a -Debug switch to output debug logs

Backup-VM -computerName '#{system.address}' -vCommanderCredential 'C:\\Scripts\vcmdr_credential_key.xml' -remoteComputerName 'myveeamhost' -VeeamCredential 'C:\\Scripts\veeam_credential_key.xml' -hostName '#{target.host.name}' -vmName '#{target.deployedName}' -vmRemoteId '#{target.remoteId}' -jobName '#{target.deployedName}-#{target.remoteId}-backup' -attribNameLastBackup 'Last Backup Time' -attribNameJobName 'Backup Job Name' -attribNameNumRestorePoints 'Number of Restore Points' }"
```

9. The Steps page looks like this:



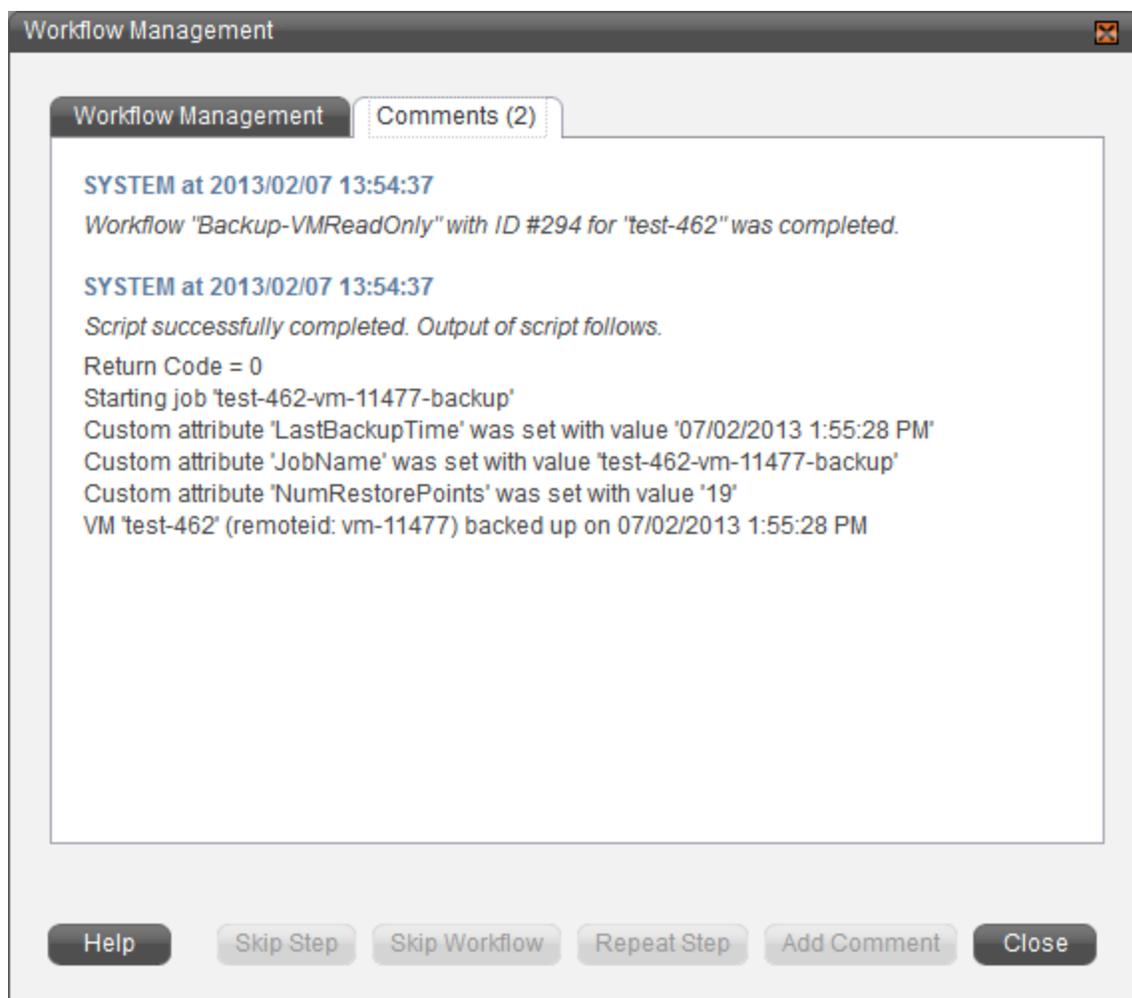
10. On the Permissions page, specify that anyone can access this workflow.
11. Click **Next** and **Finish**.

For more information, see [Workflows](#) in the *vCommander User Guide*.

## Test the integration

To test the command workflow, in vCommander or the Service Portal, right-click a VM and select **Run Workflow**. In the Select A Workflow dialog, select the workflow and click **Run**.

The Workflows tab at the bottom of your screen displays workflow status. Double-click the workflow in the list and click the **Comments** tab to view script output. (Later, once the workflow has disappeared from the Workflows tab, you can click **Tools > Workflow Status**, select a workflow, click **Workflow Management**, and click the **Comments** tab.)



Note that script output will not appear until the step completes; it will not appear if the command is never executed (that is, if the step failed at the command-line portion).

You can display the values for the Veeam-related custom attributes in the Details panel of the VM Summary tab. See [Customizing the vCommander™ Display](#) in the *vCommander User Guide*.

To see debug log statements, add the `-Debug` switch to your command line. See also [Troubleshooting](#).

## Walk-through: Adding VMs to a Backup Job During Provisioning

This walk-through shows you how to automatically add newly deployed VMs to a Veeam backup job as part of a completion workflow for a new service request.

### Create a completion workflow

To create a completion workflow to automatically add new VMs to a Veeam backup job:

1. Click **Configuration > Service Request Configuration** and select the Completion Workflow tab.
2. Click **Add**.
3. On the Name page, name the workflow "Add VM to backup job".
4. In the Apply this workflow drop-down menu, select **after a VM is deployed**.
5. On the Steps page, click **Add > Execute Script**.
6. Give the step a meaningful name, such as "Add VM to backup job".
7. For timeout, specify 300 seconds (5 minutes).
8. **IMPORTANT:** Specify that the script output should be captured as comment. Otherwise, you will not have any details in case of script failure.
9. Enter the following command line (this example uses a 32-bit version of PowerShell on 64-bit Windows):

```
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy
RemoteSigned -Command "& { import-module veeamadapterclient; import-module
vcommander;

#Add a -Debug switch to output debug logs

Add-VMToJob -remoteComputerName 'veeam' -VeeamCredential
'C:\\Scripts\encrypted_cred_key.xml' -hostName '#{target.host.name}' -vmName
 '#{target.deployedName}' -vmRemoteId '#{target.remoteId}' -jobName 'Vcommander' }"
```

10. The Steps page looks like this:

The screenshot shows a configuration window for a workflow step. On the left, under 'Step Order', there is a list with one item: '1. Add VM to backup job'. Below the list are 'Add' and 'Delete' buttons. On the right, under 'Execute Script Step Details', the following fields are visible: 'Step Name' is 'Add VM to backup job'; 'Step Execution' is set to 'Always execute'; 'Timeout' is '300 seconds'; 'Script Output' is 'Capture script output as comment'; 'When Step Fails' is 'Mark workflow step as failed: do not proceed'; and 'Command Line' is a PowerShell command: 'C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy RemoteSigned -Command "& { import-module veeamadapterclient; import-module'.

11. On the Assigned Components page, specify what components in the Service Catalog should be added to this backup job as part of provisioning. In this example, we specify that all VMs deployed from a request for our Windows Server 2008 R2 component will be added to the backup job:

The screenshot shows the 'Completion Workflow Configuration' dialog box, specifically the 'Assigned Components' tab. The title bar reads 'Completion Workflow Configuration'. Below the title, there is a section for 'Assigned Components' with the instruction: 'Choose the services that will trigger this workflow's actions once deployed.' There are three radio button options: 'Make this the default workflow (there can be only one default workflow for each type)', 'Apply this workflow to the selected components:', and 'Do not apply this workflow to any component'. The 'Apply this workflow to the selected components:' option is selected. Below this, a tree view shows 'Service Components' expanded, with 'Windows Server 2008 R2' selected and checked. At the bottom, there are 'Help', 'Back', 'Next', and 'Cancel' buttons.

12. Click **Next** and **Finish**.

For more information, see [Workflows](#) in the *vCommander User Guide*.

## Test the integration

To test the completion workflow, in vCommander or the Service Portal, request a service that contains the VM component you assigned the workflow to.

If you have set up automated provisioning, the VM will be deployed automatically; otherwise, approve and deploy the service request manually.

Open the Request Details dialog and navigate to the completion workflow in the request tree. Click the Comments tab to view script output.

**Note:** To see debug log statements, add the *-Debug* switch to your command line. See also [Troubleshooting](#).

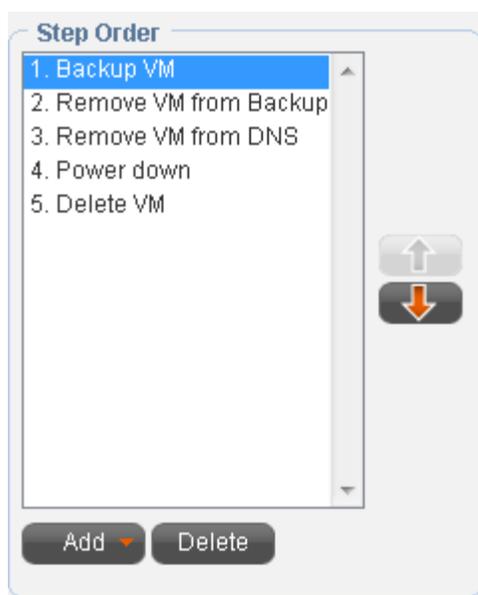
## Walk-through: Decommissioning Workflow

This walk-through shows you how to back up a VM and then remove it from a Veeam backup job as part of a decommissioning workflow.

### Create a decommissioning workflow

To create a completion workflow to back up VMs and remove them from a backup job during decommissioning:

1. Click **Configuration > Service Request Configuration** and select the Completion Workflow tab.
2. Click **Add**.
3. On the Name page, name the workflow "Decommission VM".
4. Specify that this workflow should be applied **after a Change Request is fulfilled**.
5. On the Steps page, add a total of five steps:



6. First, add a step to back up the VM. This step is covered in [Walk-through: Backing up a VM through a Command Workflow](#).
7. To add a step to remove the VM from a Veeam backup job, click **Add > Execute Script**.

Name the step "Remove VM from Backup Job".

For timeout, specify 300 seconds (5 minutes).

**Note:** To remove a VM from a Veeam backup job, a timeout of five minutes (300 seconds) is sufficient.

**IMPORTANT:** Specify that the script output should be captured as comment. Otherwise, you will not have any details in case of script failure.

Enter the following command line (this example uses a 32-bit version of PowerShell on 64-bit Windows):

```
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy RemoteSigned -Command "& { import-module veeamadapterclient; import-module vcommander;
```

```
#Add a -Debug switch to output debug logs
```

```
Remove-VMFromJob -remoteComputerName 'veeam' -veeamCredential 'C:\\Scripts\encrypted_cred_key.xml' -vmName '#{target.deployedName}' -vmRemoteId '#{target.remoteId}' -jobName 'Vcommander' }"
```

The screenshot displays the vCommander workflow editor interface. On the left, the 'Step Order' list contains five steps: 1. Back up VM, 2. Remove VM from Backup (highlighted), 3. Remove VM from DNS, 4. Power down VM, and 5. Delete VM. Below the list are 'Add' and 'Delete' buttons. On the right, the 'Execute Script Step Details' panel is shown for the selected step. The 'Step Name' is 'Remove VM from Backup Job'. The 'Step Execution' is set to 'Always execute'. The 'Timeout' is '300 seconds'. The 'Script Output' is set to 'Capture script output as comment'. The 'When Step Fails' is set to 'Mark workflow step as failed: do not proceed'. The 'Command Line' field contains the PowerShell command: `C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy RemoteSigned -Command "& { import-module veeamadapterclient; import-module`. Below the command line, there is a note: 'You can pass arguments into a script. Click here for more details.'

8. Add steps to remove the VM from the DNS, power down the VM, and delete the VM. For more information, see [Workflow Steps Reference](#) in the *vCommander User Guide*.
9. Click **Next**.

10. On the Assigned Forms page, assign this workflow to the Decommissioning Request form.

11. Click **Next** and **Finish**.

## Test the integration

To test the completion workflow, in vCommander or the Service Portal, decommission a VM.

The **Workflows** tab at the bottom of your screen displays workflow status. Double-click the workflow in the list and click the **Comments** tab to view script output. (Later, once the workflow has disappeared from the Workflows tab, you can click **Tools > Workflow Status**, select a workflow, click **Workflow Management**, and click the **Comments** tab.)

You can display the values for the Veeam-related custom attributes in the Details panel of the VM Summary tab. See [Customizing the vCommander™ Display](#) in the *vCommander User Guide*.

To see debug log statements, add the `-Debug` switch to your command line. See also [Troubleshooting](#).

## Command Reference

### Getting help

To list the functions and cmdlets available from the Veeam Adapter Client, run the following commands in PowerShell:

```
PS C:\> Import-Module VeeamAdapterClient
```

```
PS C:\> get-command -module VeeamAdapterClient
```

To get help on a cmdlet or function, run the following command in PowerShell:

```
get-help <command>
```

For example:

```
get-help Backup-VM
```

To get usage examples, run the following command in PowerShell:

```
get-help <command> -example
```

For example:

```
get-help Backup-VM -example
```

**Note:** Commands and their parameters are case insensitive.

### Add-VMToJob

#### Synopsis

Adds a VM to a Veeam backup job. If the specified VM is already in the backup job, the command will fail. The backup job must exist at the time of this request or an error is thrown. Once the VM is added to the job, the actual backup occurrence is dependent on the schedule of the job.

#### Syntax

```
Add-VMToJob -remoteComputerName <veeam-host> -veeamCredential <path> -hostName <vm-host> -vmName <vm-name> -vmRemoteId <unique-id> -jobName <job-name> [other_options]
```

#### Example command line

```
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy RemoteSigned -Command "& { import-module veeamadapterclient; import-module vcommander;
```

```
#Add a -Debug switch to output debug logs
```

```
Add-VMToJob -remoteComputerName 'veeam.mycompany.com' -veeamCredential
'C:\\Scripts\\veeam_credential_key.xml' -hostName 'esx1.mycompany.com' -vmName 'VM001'
-vmRemoteId 'vm-11538' -jobName 'Weekly Backup' }"
```

## Parameters

Parameter	Mandatory?	Description	Applicable vCommander Variables
-debug	No	Enables the debug log	N/A
-jobName <job-name>	Yes	Name of the backup job for the VM. If a name is not specified, or if the specified job does not exist in Veeam, an error is thrown	N/A
-hostName <vm-host>	Yes	Host name of the ESX. Used to speed up the query for the VM	#{target.host.name}
-remoteComputerName <veeam-host>	Yes	NETBIOS name, IP address, or fully qualified domain name of the Veeam host	N/A
-veeamCredential <path>	Yes	File path to an encrypted credential key to access the Veeam host	N/A
-vmName <vm-name>	Yes	Name of the VM. Used to speed up the query for the VM	#{target.deployedName}
-vmRemoteId <unique-id>	Yes	The unique ID for the VM, assigned by VMWare. This ID is unique across many applications including VMware, vCommander, and Veeam. The unique ID allows the Veeam Integration to handle duplicate VM names on the same host or in the same resource pool, as well as duplicate folder names (as long as they do not have identical parent folder names)	#{target.remoteId}

## Backup-VM

### Synopsis

Creates a backup job in Veeam and submits the specified VM into this job. A request is then made for this job to begin immediately.

If a job name is not specified, a job name will be created in the format `{vm_name}-{vm_remote_id}-mm/dd/yyyy hh:mm:ss {AM|PM}`. If there is another backup job with the same name, an error is thrown. Unique job names are recommended.

You can specify a number of vCommander custom attribute parameters to retrieve post-backup information such as the last successful backup timestamp, the number of restore points and the job name that was created or used. You must create these custom attributes in vCommander before running this command.

This command will perform an actual backup, which could take anywhere from minutes to potentially hours. vCommander custom attributes are updated when there is a successful backup.

### Syntax

```
Backup-VM -computerName <vCommander-host> -vCommanderCredential <path> -
remoteComputerName <veeam-host> -veeamCredential <path> -hostName <vm-host> -vmRemoteId
<unique-id> -vmName <vm-name> -jobName <job-name> [other_options]
```

### Example command line

This example shows how you can use vCommander variables in the workflow script. See [Using Variables to Access vCommander Metadata for Workflows](#) in the *vCommander User Guide* for more information.

```
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy
RemoteSigned -Command "& { import-module veeamadapterclient; import-module vcommander;
```

```
#Add a -Debug switch to output debug logs
```

```
Backup-VM -computerName 'prod.mycompany.com' -vCommanderCredential
'C:\Scripts\vcmdr_credential_key.xml' -remoteComputerName 'veeam.mycompany.com' -
veeamCredential 'C:\Scripts\veeam_credential_key.xml' -hostName 'esx1.mycompany.com'
-vmName 'VM001' -vmRemoteId 'vm-11538' -jobName 'VM001-vm-11538-backup' -
attribNameLastBackup 'Last Backup Time' -attribNameJobName 'Backup Job Name' -
attribNameNumRestorePoints 'Number of Restore Points' }
```

## Parameters

Parameter	Mandatory?	Description	Applicable vCommander Variables
-attribNameJobName <string>	No	Name of a predefined vCommander custom attribute used to track the last backup job name. The custom attribute must be of type Text. Custom attribute value will only be populated or updated for the VM when there is a successful backup. If you specify a nonexistent attribute, it is ignored	N/A
-attribNameLastBackup <string>	No	Name of a predefined vCommander custom attribute used to track the last backup time. The custom attribute must be of type Text. The format of the timestamp is set by Veeam.  Custom attribute value will only be populated or updated for the VM when there is a successful backup. If you specify a nonexistent attribute, it is ignored	N/A
-attribNameNumRestorePoints <string>	No	Name of a predefined vCommander custom attribute used to track the number of restore points for the specified VM. The custom attribute must be of type Text. Custom attribute value will only be populated or updated for the VM when there is a successful backup. If you specify a nonexistent attribute, it is ignored	N/A
-computerName <vCommander host>	Yes	NETBIOS name, IP address, or fully qualified domain name of the vCommander host	#{system.address}
-debug	No	Enables the debug logging level; log messages appear in workflow comments	N/A
-hostName <vm-host>	Yes	Host name of the ESX. Used to speed up the query for the VM	#{target.host.name}

Parameter	Mandatory?	Description	Applicable vCommander Variables
-jobName <job-name>	No	Job name for the request. If not specified, a name with the following format will be assigned: <vmName>-<vm remote id>-dd/mm/yyyy hh:mm:ss PM AM	{target.deployedName} {target.remoteld}
-remoteComputerName <veeam-host>	Yes	NETBIOS name, IP address, or fully qualified domain name of the Veeam host	N/A
-veeamCredential <path>	Yes	File path to an encrypted credential key to access the Veeam host	N/A
-vCommanderCredential <path>	Yes	File path to an encrypted credential key to access vCommander. This account must have permission to set custom attributes within vCommander, and the account must have visibility of the VM being backed up	N/A
-vmName <vm-name>	Yes	Name of the VM. Used to speed up the query for the VM	{target.deployedName}
-vmRemoteld <unique-id>	Yes	The unique ID for the VM, assigned by VMWare. This ID is unique across many applications including VMware, vCommander, and Veeam. The unique ID allows the Veeam Integration to handle duplicate VM names on the same host or in the same resource pool, as well as duplicate folder names (as long as they do not have identical parent folder names)	{target.remoteld}

## Remove-VMFromJob

### Synopsis

Adds a VM to a Veeam exclusion list so that it is no longer backed up. If the specified VM is already in the exclusion list, the command is ignored and the log indicates success. The actual removal process depends on the retention policy configured for this job. The backup job must exist or an error is thrown.

### Syntax

```
Remove-VMFromJob -remoteComputerName <veeam-host> -veeamCredential <path> -vmName <vm-name> -vmRemoteId <unique-id> -jobName <job-name> [other_options]
```

### Example command line

```
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy RemoteSigned -Command "& { import-module veeamadapterclient; import-module vcommander;
```

```
#Add a -Debug switch to output debug logs
```

```
Remove-VMFromJob -remoteComputerName 'veeam.mycompany.com' -veeamCredential 'C:\\Scripts\\veeam_credential_key.xml' -vmName 'VM001' -vmRemoteId 'vm-11538' -jobName 'Weekly Backup' }"
```

### Parameters

Parameter	Mandatory?	Description	Applicable vCommander Variables
-debug	No	Enables the debug log	N/A
-jobName <job-name>	Yes	Name of the backup job for the VM. If a name is not specified, or if the specified job does not exist in Veeam, an error is thrown	N/A
-remoteComputerName <veeam-host>	Yes	NETBIOS name, IP address, or fully qualified domain name of the Veeam host	N/A
-veeamCredential <path>	Yes	File path to an encrypted credential key to access the Veeam host	N/A
-vmName <vm-name>	Yes	Name of the VM. Used to speed up the query for the VM	#{target.deployedName}

Parameter	Mandatory?	Description	Applicable vCommander Variables
-vmRemoteld <unique-id>	Yes	The unique ID for the VM, assigned by VMWare. This ID is unique across many applications including VMware, vCommander, and Veeam. The unique ID allows the Veeam Integration to handle duplicate VM names on the same host or in the same resource pool, as well as duplicate folder names (as long as they do not have identical parent folder names)	#{target.remoteld}

## Start-QuickBackup

### Synopsis

Performs an on-demand incremental backup for a specific VM. The VM must be added to a backup job and must have at least one full backup.

**Note:** This function is supported only with Veeam 8.0 and higher.

### Syntax

```
Start-QuickBackup -remoteComputerName <veeam-host> -veeamCredential <path> -vmName <vm-name> -vmRemoteId <unique-id> -hostName <vm-host-name> -jobName <job-name>
```

### Example command line

```
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy RemoteSigned -Command "& { import-module veeamadapterclient; import-module vcommander;
```

```
#Add a -Debug switch to output debug logs
```

```
Start-QuickBackup -remoteComputerName 'veeam.mycompany.com' -veeamCredential 'C:\\Scripts\veeam_encrypted_key.xml' -vmName 'VM001' -vmRemoteId 'vm-11538' -hostName 'esxl.embotics.com' -jobName 'Monthly Backup' }"
```

### Parameters

Parameter	Mandatory?	Description	Applicable vCommander Variables
-debug	No	Enables the debug logging level; log messages appear in workflow comments	N/A

Parameter	Mandatory?	Description	Applicable vCommander Variables
-hostName <vm-host>	Yes	Host name of the ESX. Used to speed up the query for the VM	#{target.host.name}
-jobName <job-name>	No	Job name for the request. If not specified, a name with the following format will be assigned: <vmName>-<vm remote id>-dd/mm/yyyy hh:mm:ss PM AM	#{target.deployedName} #{target.remoteld}
-remoteComputerName <veeam-host>	Yes	NETBIOS name, IP address, or fully qualified domain name of the Veeam host	N/A
-veeamCredential <path>	Yes	File path to an encrypted credential key to access the Veeam host	N/A
-vmName <vm-name>	Yes	Name of the VM. Used to speed up the query for the VM	#{target.deployedName}
-vmRemoteld <unique-id>	Yes	The unique ID for the VM, assigned by VMWare. This ID is unique across many applications including VMware, vCommander, and Veeam. The unique ID allows the Veeam Integration to handle duplicate VM names on the same host or in the same resource pool, as well as duplicate folder names (as long as they do not have identical parent folder names)	#{target.remoteld}

## Troubleshooting

Script output is captured as workflow comment (you must specify that the workflow step output is captured as comment). To view script output, click **Tools > Workflow Status**, select a workflow, click **Workflow Management**, and click the **Comments** tab.

**Note:** A maximum of 8000 characters of script output can be captured as workflow comment. If the script output is longer than 8000 characters, it will be truncated. If the log level is set to Debug, some log entries will not be displayed. To see debug log statements, add the *-Debug* switch to your command line.

Exceptions are also printed to the vCommander log. More information is generally printed in the log file than in the workflow comments.

**Table: Veeam Integration Issues**

Issue	Solution or Comment
1. Can't run PowerShell on 64-bit systems	PowerShell may look at the 32-bit version path regardless of which version of PowerShell you are running. You may need to install the modules in the 32-bit location. See <a href="#">Install the Veeam PowerShell Module</a> .
2. PowerShell message: "<Script-file> cannot be loaded because the execution of scripts is disabled on this system"	Set the correct execution policy with the following command:  Set-ExecutionPolicy RemoteSigned  If you still get the error and your machine is 64-bit, ensure that the execution policy is set for the correct instance of PowerShell and that the correct instance of PowerShell is used in the workflow script step.
3. PowerShell message: "Enable-PSRemoting: Access is denied. You have to run this cmdlet from an elevated process"	Run PowerShell as administrator.
4. PowerShell message: "Set-ExecutionPolicy : Access to the registry key 'HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\PowerShell\1\ShellIds\Microsoft.PowerShell' is denied"	Run PowerShell as administrator.
5. PowerShell message: "The Windows PowerShell snap-in 'Veeam PsSnapIn' is not installed on this machine"	Install the PowerShell Snap-in feature of Veeam. See <a href="#">Verify Installation of the Veeam PowerShell Snap-in</a> .

Issue	Solution or Comment
6. PowerShell warning when attempting to load the VCommander module: "Import-Module : Could not load file or assembly '<file>' or one of its dependencies. Operation is not supported."	Because you downloaded the .zip file from the Internet, Windows is treating the contents as "untrusted". Right-click the downloaded .zip file and select <b>Properties</b> . On the General tab, click <b>Unblock</b> . Now, extract the .zip file again. <b>Note:</b> If you do not have full permissions on the folder where you saved the .zip file, the file will remain blocked. If this happens, move the .zip file to a location where you have full permissions (for example, your desktop), and unblock the .zip file there.
7. vCommander message: "Duplicate job name; unable to decide which job to run"	Specify a different job name.
8. vCommander message: "Unable to establish connection to vCommander webservices on host '{0}'. Cause {1}"	<p>First, make sure the account that the encrypted credential key represents exists on the vCommander Users and Roles page.</p> <p>Second, make sure you enter the correct credentials when generating the encryption key. See <a href="#">Encrypt Your Authentication Credentials</a>.</p>
9. vCommander message: "Unable to login to vCommander webservices as '{0}'. Cause {1}"	<p>First, make sure the account that the encrypted credential key represents exists on the vCommander Users and Roles page.</p> <p>Second, make sure you enter the correct credentials when generating the encryption key. See <a href="#">Encrypt Your Authentication Credentials</a>.</p>
10. No script output in workflow comments, but script step executed successfully	Check the vCommander log for script errors. While certain parts of the script step command line (such as variables) can be validated immediately, the entire script step cannot be validated until it is run.
11. vCommander message: "Failed to locate VM with remote id '{0}'. Does the user have visibility of this VM?"	In vCommander, click <b>Configuration &gt; Users and Roles</b> . Select the user and ensure that under <b>Access Rights in</b> the right pane, the user can see the VMs.

# vCommander Veeam Integration Guide

vCommander version 5.7 and higher  
Veeam version 1.1

embotics



© 2018, Embotics Corporation, all rights reserved. Information described herein is furnished for informational use only, is subject to change without notice, and should not be taken as a guarantee, commitment, condition or offer by Embotics. Embotics, the Embotics logo, and all other Embotics product names and logos are trademarks or registered trademarks of Embotics and/or its subsidiaries in the United States and other countries. All other product or company names are property of their respective owners.