Red Hat Enterprise Linux 6 V2V Guide

Importing Virtual Machines with virt-v2v

Dayle Parker
Tim Hildred
Laura Novich
David Jorm
Laura Bailey

Red Hat Enterprise Linux 6 V2V Guide

Importing Virtual Machines with virt-v2v

Dayle Parker
Red Hat Engineering Content Services
dayleparker@redhat.com

Laura Novich
Red Hat Engineering Content Services
lnovich@redhat.com

Laura Bailey
Red Hat Engineering Content Services
lbailey@redhat.com

Tim Hildred
Red Hat Engineering Content Services
thildred@redhat.com

David Jorm
Red Hat Engineering Content Services
djorm@redhat.com

Abstract

This book is a guide to importing virtual machines from foreign hypervisors to Red Hat Enterprise Virtualization and KVM managed by libvirt.

Table of Contents

Preface

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the Liberation Fonts set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later include the Liberation Fonts set by default.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

`Mono-spaced Bold`

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keys and key combinations. For example:

> To see the contents of the file **`my_next_bestselling_novel`** in your current working directory, enter the **`cat my_next_bestselling_novel`** command at the shell prompt and press **`Enter`** to execute the command.

The above includes a file name, a shell command and a key, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from an individual key by the plus sign that connects each part of a key combination. For example:

> Press **`Enter`** to execute the command.
>
> Press **`Ctrl`**+**`Alt`**+**`F2`** to switch to a virtual terminal.

The first example highlights a particular key to press. The second example highlights a key combination: a set of three keys pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **`mono-spaced bold`**. For example:

> File-related classes include **`filesystem`** for file systems, **`file`** for files, and **`dir`** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog-box text; labeled buttons; check-box and radio-button labels; menu titles and submenu titles. For example:

> Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **`Buttons`** tab, select the **`Left-handed mouse`** check box and click **`Close`** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).
>
> To insert a special character into a **gedit** file, choose **Applications** → **Accessories** →

Character Map from the main menu bar. Next, choose **Search** → **Find...** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

To connect to a remote machine using ssh, type **ssh *username*@*domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.

The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.

To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above: username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

Publican is a *DocBook* publishing system.

1.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
static int kvm_vm_ioctl_deassign_device(struct kvm *kvm,
                  struct kvm_assigned_pci_dev *assigned_dev)
{
          int r = 0;
          struct kvm_assigned_dev_kernel *match;

          mutex_lock(&kvm->lock);

          match = kvm_find_assigned_dev(&kvm->arch.assigned_dev_head,
                                        assigned_dev->assigned_dev_id);
          if (!match) {
                  printk(KERN_INFO "%s: device hasn't been assigned before, "
                    "so cannot be deassigned\n", __func__);
                  r = -EINVAL;
                  goto out;
          }

          kvm_deassign_device(kvm, match);

          kvm_free_assigned_device(kvm, match);

out:
          mutex_unlock(&kvm->lock);
          return r;
}
```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled “Important” will not cause data loss but may cause irritation and frustration.

Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

2. We Need Feedback!

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in Bugzilla: http://bugzilla.redhat.com/ against the product **Red Hat Enterprise Linux 6**.

When submitting a bug report, be sure to mention the manual's identifier: *doc-V2V_Guide*.

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, include the section number and some of the surrounding text so we can find it easily.

Chapter 1. Introducing V2V

V2V is an acronym for virtual to virtual, referring to the process of importing virtual machines from one virtualization platform to another. Red Hat Enterprise Virtualization and Red Hat Enterprise Linux are capable of performing V2V operations using the **virt-v2v** command.

1.1. What is virt-v2v?

The **virt-v2v** command converts virtual machines from a foreign hypervisor to run on KVM, managed by Red Hat Enterprise Virtualization or libvirt. **virt-v2v** can currently convert virtual machines running Red Hat Enterprise Linux and Windows on Xen, KVM and VMware ESX / ESX(i) hypervisors. **virt-v2v** enables para-virtualized (**virtio**) drivers in the converted virtual machine if possible.

The following guest operating systems are supported by **virt-v2v**:

Supported guest operating systems:

- Red Hat Enterprise Linux 3.9
- Red Hat Enterprise Linux 4
- Red Hat Enterprise Linux 5
- Red Hat Enterprise Linux 6
- Windows XP
- Windows Vista
- Windows 7
- Windows Server 2003
- Windows Server 2008

All minor releases of the above guest operating systems are supported by **virt-v2v**.

The following source hypervisors are supported by **virt-v2v**:

Supported source hypervisors:

Unless otherwise specified, all minor releases of the following source hypervisors are supported by **virt-v2v**:

- Xen — all versions released by Red Hat
- KVM — all versions released by Red Hat
- VMware ESX / ESX(i) — versions 3.5, 4.0, 4.1, 5.0, 5.1

1.2. About this guide

This guide describes how to import virtual machines from foreign hypervisors to Red Hat Enterprise Virtualization and KVM managed by libvirt.

1.2.1. Audience

This guide is intended for system administrators who manage a virtualized environment using Red Hat Enterprise Virtualization or Red Hat Enterprise Linux. An advanced level of system administration, preferably including familiarity with virtual machine data center operations, is assumed. This document is not intended for beginners.

1.2.2. Red Hat Enterprise Virtualization

The Red Hat Enterprise Virtualization platform is a richly featured virtualization management solution providing fully integrated management across virtual machines. It is based on the leading open source virtualization platform and provides superior technical capabilities. The platform offers scalability in the management of large numbers of virtual machines.

1.3. Further reading

Red Hat offers a wealth of documentation solutions across its various virtualization products. Coverage of Red Hat Enterprise Linux and its inbuilt virtualization products includes:

- *Red Hat Enterprise Linux — Virtualization Getting Started Guide*: This guide provides an introduction to virtualization concepts, advantages, and tools, and an overview of Red Hat virtualization documentation and products.
- *Red Hat Enterprise Linux — Virtualization Host Configuration and Guest Installation Guide*: This guide covers the installation of virtualization software and configuration of guest machines on a virtualization host.
- *Red Hat Enterprise Linux — Virtualization Administration Guide*: This guide covers administration of hosts, networking, storage, and device and guest management using either virt-manager or virsh as primary configuration tools. This guide also includes a libvirt and QEMU reference, as well as troubleshooting information.
- *Red Hat Enterprise Linux — Virtualization Security Guide*: This guide provides an overview of virtualization security technologies provided by Red Hat. Also included are recommendations for securing hosts, guests, and shared infrastructure and resources in virtualized environments.
- *Red Hat Enterprise Linux — Virtualization Tuning and Optimization Guide*: This guide provides tips, tricks and suggestions for making full use of virtualization performance features and options for your systems and guest virtual machines.
- *Red Hat Enterprise Linux — V2V Guide*: This guide describes importing virtual machines from KVM, Xen and VMware ESX / ESX(i) hypervisors to Red Hat Enterprise Virtualization and KVM managed by libvirt.

The Red Hat Enterprise Virtualization documentation suite provides information on installation, development of applications, configuration and usage of the Red Hat Enterprise Virtualization platform and its related products.

- *Red Hat Enterprise Virtualization — Administration Guide* describes how to set up, configure and manage Red Hat Enterprise Virtualization. It assumes that you have successfully installed the Red Hat Enterprise Virtualization Manager and hosts.
- *Red Hat Enterprise Virtualization — Command Line Shell Guide* contains information for installing and using the Red Hat Enterprise Virtualization Manager command line shell.
- *Red Hat Enterprise Virtualization — Developer Guide* explains how to use the REST API. It covers the fundamentals of the REST architectural concepts in the context of a virtualization environment and provides examples of the API in operation. It also documents the installation and use of the Python Software Development Kit.
- *Red Hat Enterprise Virtualization — Evaluation Guide* enables prospective customers to evaluate the features of Red Hat Enterprise Virtualization. Use this guide if you have an evaluation license.
- *Red Hat Enterprise Virtualization — Installation Guide* describes the installation prerequisites and procedures. Read this if you need to install Red Hat Enterprise Virtualization. The installation of hosts, Manager and storage are covered in this guide. You will need to refer to the *Red Hat Enterprise Virtualization Administration Guide* to configure the system before you can start using the platform.

- *Red Hat Enterprise Virtualization — Manager Release Notes* contain release specific information for Red Hat Enterprise Virtualization Managers.
- *Red Hat Enterprise Virtualization — Power User Portal Guide* describes how power users can create and manage virtual machines from the Red Hat Enterprise Virtualization User Portal.
- *Red Hat Enterprise Virtualization — Quick Start Guide* provides quick and simple instructions for first time users to set up a basic Red Hat Enterprise Virtualization environment.
- *Red Hat Enterprise Virtualization — Technical Notes* describe the changes made between the current release and the previous one.
- *Red Hat Enterprise Virtualization — Technical Reference Guide* describes the technical architecture of Red Hat Enterprise Virtualization and its interactions with existing infrastructure.
- *Red Hat Enterprise Virtualization — User Portal Guide* describes how users of the Red Hat Enterprise Virtualization system can access and use virtual desktops from the User Portal.

Note

All of the guides for these products are available at the Red Hat Customer Portal: https://access.redhat.com/site/documentation/

Chapter 2. Installing virt-v2v

virt-v2v is run from a Red Hat Enterprise Linux host. It must be installed on the host.

Procedure 2.1. Installing virt-v2v

1. **Subscribe to the virt-v2v RHN channel**

   virt-v2v is available on Red Hat Network (RHN) in the **Red Hat Enterprise Linux Server (v.6 for 64-bit x86_64)** or **Red Hat Enterprise Linux Workstation (v.6 for x86_64)** channel. Ensure the system is subscribed to the appropriate channel before installing **virt-v2v**.

2. **Install the pre-requisites**

   - If you are converting Windows virtual machines, you must install the *libguestfs-winsupport* and *virtio-win* packages. These packages provide support for NTFS and Windows para-virtualized block and network drivers.
   - If you attempt to convert a virtual machine using NTFS without the *libguestfs-winsupport* package installed, the conversion will fail.
   - If you attempt to convert a virtual machine running Windows without the *virtio-win* package installed, the conversion will fail giving an error message concerning missing files.

3. **Log on to Red Hat Network**

   The *libguestfs-winsupport* is available for Red Hat Enterprise Linux Server 6 in the **Red Hat Enterprise Linux Server V2V Tools for Windows (v. 6)** channel, while the *virtio-win* package is available in the **Red Hat Enterprise Linux Server Supplementary (v. 6)** channel. To install these packages, ensure that your system has the required permissions to subscribe to both channels and run the following command as root:

   ```
   rhn-channel -a -c rhel-x86_64-server-supplementary-6 --user USERNAME --password PASSWORD
   ```

   ```
   rhn-channel -a -c rhel-x86_64-server-v2vwin-6 --user USERNAME --password PASSWORD
   ```

   ```
   yum install libguestfs-winsupport virtio-win
   ```

4. **Install virt-v2v package**

   As root, run the command:

   ```
   yum install virt-v2v
   ```

5. Run **virt-v2v** as the root user from a Linux shell.

Chapter 3. Converting virtual machines to run on KVM managed by libvirt

virt-v2v can convert virtual machines to run on Red Hat Enterprise Linux, using KVM managed by libvirt. Virtual machines can be converted from Xen, KVM, VMware ESX / ESX(i), and Hyper-V environments. The **libvirt** and **virsh** commands are documented in the *Red Hat Enterprise Linux Virtualization Administration Guide*. You may also use the command **man virsh** for more information.

3.1. Introduction

The **virt-v2v** command converts virtual machines from a foreign hypervisor to run on KVM, managed by libvirt. The following guest operating systems are supported by **virt-v2v**:

- Red Hat Enterprise Linux 3.9
- Red Hat Enterprise Linux 4
- Red Hat Enterprise Linux 5
- Red Hat Enterprise Linux 6
- Windows XP
- Windows Vista
- Windows 7
- Windows Server 2003
- Windows Server 2008

The following hypervisors are supported:

- KVM
- libvirt-managed Xen
- VMware ESX / ESX(i) - versions 3.5, 4.0, 4.1, 5.0, 5.1
- Hyper-V

The **virt-v2v** command enables para-virtualized (**virtio**) drivers in the converted guest, if possible.

virt-v2v is available on Red Hat Network (RHN) in the **Red Hat Enterprise Linux Server (v.6 for 64-bit x86_64)** or **Red Hat Enterprise Linux Workstation (v.6 for x86_64)** channel.

The **virt-v2v** tool requires root access to the host system.

Some of the new features for **virt-v2v** starting with Red Hat Enterprise Linux 6 are:

- The **-op** and **-osd** command line options continue to be supported, but are deprecated in favour of **-os**. There is no deprecation warning when they are used.
- The **-of** command line option allows specification of the file format to be used for target storage: *raw* or *qcow2*. This feature allows for the conversion of a virtual machine with raw storage to qcow2 and vice versa.
- The **-oa** command line option allows the allocation policy of the target storage to be specified: *sparse* or *preallocated*. This can be used to convert between sparse and preallocated. Underlying this change, sparse volumes are now supported.
- The configuration file can now contain target profiles, which specify the storage location, output format and allocation policy for a target. This allows the user to specify **--profile<foo>** rather

than **-os<a> -op<b> -oa<oc>**.

- The conversion of Windows virtual machines to libvirt targets is supported.

Refer to the **virt-v2v** man page for further details on these and other features.

To install **virt-v2v** from RHN, ensure the system is subscribed to the appropriate channel, then run:

```
# yum install virt-v2v
```

3.2. Preparing to convert a virtual machine

virt-v2v converts virtual machines from a foreign hypervisor to run on Red Hat Enterprise Linux, using KVM managed by libvirt. It automatically creates a libvirt domain for the converted virtual machines.

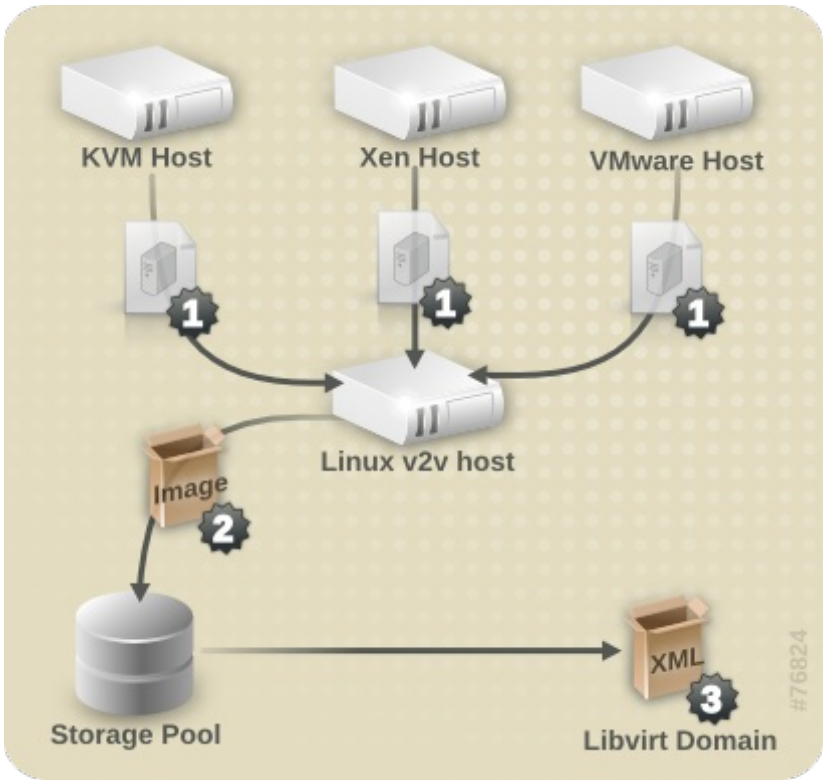

Figure 3.1. Converting a virtual machine

Before a virtual machine can be converted, ensure that the following steps are completed.

Procedure 3.1. Preparing a virtual machine for conversion

1. Create a local storage domain for transferred storage.

   virt-v2v copies the guest virtual machine storage to a locally defined libvirt storage pool during import. This pool can be defined using any libvirt tool, and can be of any type. The simplest way to create a new pool is with **virt-manager**. Refer to the *Red Hat Enterprise Linux Virtualization Administration Guide* for complete instructions on creating storage pools with either **virt-manager** or **virsh**.

2. Create local network interfaces.

   The local machine must have an appropriate network to which the converted virtual machine can connect. This is likely to be a bridge interface. A bridge interface can be created using standard

tools on the host. From libvirt version 0.8.3 and onward, **virt-manager** can also create and manage bridges. For more information on bridged networking with libvirt, see the *Red Hat Enterprise Linux Virtualization Host Configuration and Guest Installation Guide* or the *Red Hat Enterprise Linux Virtualization Administration Guide*.

3. Specify network mappings in **virt-v2v.conf**. This step is *optional*, and is not required for most use cases.

   If your virtual machine has multiple network interfaces, **/etc/virt-v2v.conf** must be edited to specify the network mapping for all interfaces. You can specify an alternative **virt-v2v.conf** file with the ***-f*** parameter.

   If your virtual machine only has a single network interface, it is simpler to use the ***--network*** or ***--bridge*** parameters, rather than modifying **virt-v2v.conf**.

4. Create a profile for the conversion in **virt-v2v.conf**.

   This step is *optional*. Profiles specify a conversion method, storage location, output format and allocation policy. When a profile is defined, it can be called using ***--profile*** rather than individually providing the ***-o***, ***-os***, ***-of*** and ***-oa*** parameters. See virt-v2v.conf(5) for details.

3.2.1. Preparing to convert a virtual machine running Linux

Before a virtual machine running Linux can be converted, ensure that the following steps are completed.

Procedure 3.2. Preparing to convert a virtual machine running Linux

1. Obtain the software.

   As part of the conversion process, **virt-v2v** may install a new kernel and drivers on the virtual machine. If the virtual machine being converted is registered to Red Hat Network (RHN), the required packages will be automatically downloaded. For environments where RHN is not available, the **virt-v2v.conf** file references a list of RPMs used for this purpose. The RPMs relevant to your virtual machine must be downloaded manually from RHN and made available in the directory specified by the ***path-root*** configuration element, which by default is **/var/lib/virt-v2v/software/**. **virt-v2v** will display an error similar to Example 3.1, “Missing Package error” if the software it depends upon for a particular conversion is not available.

   Example 3.1. Missing Package error

   ```
   virt-v2v: Installation failed because the following files referenced in the
   configuration file are required, but missing:
   rhel/6/kernel-2.6.32-128.el6.x86_64.rpm
   rhel/6/ecryptfs-utils-82-6.el6.x86_64.rpm
   rhel/6/ecryptfs-utils-82-6.el6.i686.rpm
   ```

2. To obtain the relevant RPMs for your environment, repeat these steps for each missing package:
   a. Log in to the Red Hat Customer Portal: https://access.redhat.com/.
   b. In the Red Hat Customer Portal, select the **Downloads** tab to open the **Software & Download Center** page.
   c. Select **Packages** to enter the **Package Search** page (https://rhn.redhat.com/rhn/channels/software/Search.do).
   d. In the **Search For** field, type the package name exactly matching the one shown in the error message. For the example shown in Example 3.1, “Missing Package error”, the first package is **kernel-2.6.32-128.el6.x86_64**
   e. In the **Where to search** field, select **In the following architectures** and select

the **x86_64** checkbox. Click **Search**.

f. A list of packages displays. Select the package name identical to the one in the error message. This opens the details page, which contains a detailed description of the package.

g. Select **Download Package** at the bottom of the page.

h. Save the downloaded package to the appropriate directory in **/var/lib/virt-v2v/software**. For Red Hat Enterprise Linux 6, the directory is **/var/lib/virt-v2v/software/rhel/6**.

3.2.2. Preparing to convert a local Xen virtual machine

The following is required when converting virtual machines on a host which used to run Xen, but has been updated to run KVM. It is not required when converting a Xen virtual machine imported directly from a running libvirt/Xen instance.

Procedure 3.3. Preparing to convert a local Xen virtual machine

- Obtain the XML for the virtual machine.

  virt-v2v uses a libvirt domain description to determine the current configuration of the virtual machine, including the location of its storage. Before starting the conversion, obtain the XML from the host running the virtual machine with the following command:

  ```
  virsh dumpxml guest_name > guest_name.xml
  ```

  This will require booting into a Xen kernel to obtain the XML, as libvirt needs to connect to a running Xen hypervisor to obtain its metadata. The conversion process is optimized for KVM, so obtaining domain data while running a Xen kernel, then performing the conversion using a KVM kernel will be more efficient than running the conversion on a Xen kernel.

3.2.3. Preparing to convert a virtual machine running Windows

Important

virt-v2v does not support conversion of the Windows Recovery Console. If a virtual machine has a recovery console installed and VirtIO was enabled during conversion, attempting to boot the recovery console will result in a stop error.
Windows XP x86 does not support the Windows Recovery Console on VirtIO systems, so there is no resolution to this. However, on Windows XP AMD64 and Windows 2003 (x86 and AMD64), the recovery console can be re-installed after conversion. The re-installation procedure is the same as the initial installation procedure. It is not necessary to remove the recovery console first. Following re-installation, the recovery console will work as intended.

Before a virtual machine running Windows can be converted, ensure that the following steps are completed.

1. Install the *libguestfs-winsupport* package on the host running **virt-v2v**. This package provides support for NTFS, which is used by many Windows systems. The *libguestfs-winsupport* package is provided by the **RHEL V2VWIN (v. 6 for 64-bit x86_64)** channel. Ensure your system is subscribed to this channel, then run the following command as root:

   ```
   yum install libguestfs-winsupport
   ```

If you attempt to convert a virtual machine using NTFS without the *libguestfs-winsupport* package installed, the conversion will fail. An error message similar to Example 3.2, "Error message when converting a Windows virtual machine without libguestfs-winsupport installed" will be shown.

Example 3.2. Error message when converting a Windows virtual machine without libguestfs-winsupport installed

```
No operating system could be detected inside this disk image.

This may be because the file is not a disk image, or is not a virtual
machine
image, or because the OS type is not understood by virt-inspector.

If you feel this is an error, please file a bug report including as much
information about the disk image as possible.
```

2. Install the *virtio-win* package on the host running **virt-v2v**. This package provides para-virtualized block and network drivers for Windows guests. The *virtio-win* package is provided by the **RHEL Server Supplementary (v. 6 64-bit x86_64)** channel. Ensure your system is subscribed to this channel, then run the following command as root:

```
yum install virtio-win
```

If you attempt to convert a virtual machine running Windows without the *virtio-win* package installed, the conversion will fail. An error message similar to Example 3.3, "Error message when converting a Windows virtual machine without virtio-win installed" will be shown.

Example 3.3. Error message when converting a Windows virtual machine without virtio-win installed

```
virt-v2v: Installation failed because the following files referenced in the
configuration file are required, but missing: /usr/share/virtio-
win/drivers/i386/Win2008
```

Note

When virtual machines running Windows are converted for output to Red Hat Enterprise Virtualization, post-processing of the virtual machine image will be performed by the Red Hat Enterprise Virtualization Manager to install updated drivers. See Section 7.2.2, "Configuration changes for Windows virtual machines" for details of the process. This step will be omitted when virtual machines running Windows are converted for output to libvirt.

3.3. Converting a virtual machine

Once you have prepared to convert the virtual machines, use **virt-v2v** to perform the actual conversions. This section provides the steps to convert the virtual machines, and the command syntax for **virt-v2v**.

Note that conversions are resource intensive processes that require copying the whole disk image for a virtual machine. In typical environments, converting a single virtual machine takes approximately 5-10 minutes. In Example 3.4, "Typical virt-v2v conversion time" a virtual machine with a single 8GB disk is

copied over SSH on a 1GigE network on three-year-old consumer hardware:

Example 3.4. Typical virt-v2v conversion time

```
win2k3r2-pv-32.img: 100% [=========================================]D
0h02m57s
virt-v2v: win2k3r2-pv-32 configured with virtio drivers.
```

The size of the disk to be copied is the major factor in determining conversion time. For a virtual machine on average hardware with a single disk of 20GB or less, a conversion usually takes less than 10 minutes.

3.3.1. Converting a local virtual machine using virt-v2v

virt-v2v converts virtual machines from a foreign hypervisor to run on KVM, managed by libvirt. The general command syntax for converting machines to run on KVM, managed by libvirt is:

```
virt-v2v -i libvirtxml -op pool --bridge bridge_name guest_name.xml
virt-v2v -op pool --network netname guest_name
virt-v2v -ic esx://esx.example.com/?no_verify=1 -op pool --bridge bridge_name
guest_name
```

For a list of **virt-v2v** parameters, refer to Chapter 7, *References*.

3.3.2. Converting a remote KVM virtual machine

KVM virtual machines can be converted remotely via SSH. Ensure that the host running the virtual machine is accessible via SSH.

To convert the virtual machine, run:

```
virt-v2v -ic qemu+ssh://root@vmhost.example.com/system -op pool --bridge
bridge_name guest_name
```

Where **vmhost.example.com** is the host running the virtual machine, **pool** is the local storage pool to hold the image, ***bridge_name*** is the name of a local network bridge to connect the converted virtual machine's network to, and ***guest_name*** is the name of the Xen virtual machine.

You may also use the ***--network*** parameter to connect to a locally managed network if your virtual machine only has a single network interface. If your virtual machine has multiple network interfaces, edit **/etc/virt-v2v.conf** to specify the network mapping for all interfaces.

If your virtual machine is Red Hat Enterprise Linux 4 or Red Hat Enterprise Linux 5 and uses a kernel which doesn't support the KVM VirtIO drivers, *virt-v2v* will attempt to install a new kernel during the conversion process. You can avoid this requirement by updating the kernel to a recent version of Red Hat Enterprise Linux 6 which supports VirtIO prior to conversion.

Note

When converting from KVM, **virt-v2v** requires that the image of the source virtual machine exists within a storage pool. If the image is not currently in a storage pool, you must create one.

3.3.3. Converting a local Xen virtual machine

Ensure that the guest virtual machine's XML is available locally, and that the storage referred to in the XML is available locally at the same paths.

To convert the virtual machine from an XML file, run:

```
virt-v2v -i libvirtxml -op pool --bridge bridge_name guest_name.xml
```

Where **pool** is the local storage pool to hold the image, **bridge_name** is the name of a local network bridge to connect the converted virtual machine's network to, and **guest_name.xml** is the path to the virtual machine's exported XML.

You may also use the **--network** parameter to connect to a locally managed network if your virtual machine only has a single network interface. If your virtual machine has multiple network interfaces, edit **/etc/virt-v2v.conf** to specify the network mapping for all interfaces.

If your virtual machine uses a Xen para-virtualized kernel (it would be called something like **kernel-xen** or **kernel-xenU**), **virt-v2v** will attempt to install a new kernel during the conversion process. You can avoid this requirement by installing a regular kernel, which will not reference a hypervisor in its name, alongside the Xen kernel prior to conversion. You should not make this newly installed kernel your default kernel, because Xen will not boot it. **virt-v2v** will make it the default during conversion.

Note

When converting from Xen, **virt-v2v** requires that the image of the source virtual machine exists in a storage pool. If the image is not currently in a storage pool, you must create one. Contact Red Hat Support for assistance creating an appropriate storage pool.

3.3.4. Converting a remote Xen virtual machine

Xen virtual machines can be converted remotely via SSH. Ensure that the host running the virtual machine is accessible via SSH.

To convert the virtual machine, run:

```
virt-v2v -ic qemu+ssh://root@vmhost.example.com/system -op pool --bridge
bridge_name  guest_name
```

Where **vmhost.example.com** is the host running the virtual machine, **pool** is the local storage pool to hold the image, ***bridge_name*** is the name of a local network bridge to connect the converted virtual machine's network to, and ***guest_name*** is the name of the Xen virtual machine.

You may also use the **--network** parameter to connect to a locally managed network if your virtual machine only has a single network interface. If your virtual machine has multiple network interfaces, edit **/etc/virt-v2v.conf** to specify the network mapping for all interfaces.

If your virtual machine uses a Xen para-virtualized kernel (it would be called something like **kernel-xen** or **kernel-xenU**) **virt-v2v** will attempt to install a new kernel during the conversion process. You can avoid this requirement by installing a regular kernel, which will not reference a hypervisor in its name, alongside the Xen kernel prior to conversion. You should not make this newly installed kernel your default kernel, because Xen will not boot it. **virt-v2v** will make it the default during conversion.

Note

When converting from Xen, **virt-v2v** requires that the image of the source virtual machine exists in a storage pool. If the image is not currently in a storage pool, you must create one. Contact Red Hat Support for assistance creating an appropriate storage pool.

3.3.5. Converting a VMware ESX / ESX(i) virtual machine

Important

When converting a Windows virtual machine from VMware ESX / ESX(i), ensure that VMware Tools is not installed on the virtual machine. The VMware Tools must be uninstalled prior to the conversion process. If a virtual machine is converted with the VMware Tools installed, it will not function correctly.

Ensure that the virtual machine is stopped prior to running the v2v process.

To convert the virtual machine, run:

```
virt-v2v -ic esx://esx.example.com/ -op pool --bridge bridge_name guest_name
```

Where **esx.example.com** is the VMware ESX / ESX(i) server, **pool** is the local storage pool to hold the image, **bridge_name** is the name of a local network bridge to connect the converted virtual machine's network to, and **guest_name** is the name of the virtual machine.

You may also use the ***--network*** parameter to connect to a locally managed network if your virtual machine only has a single network interface. If your virtual machine has multiple network interfaces, edit **/etc/virt-v2v.conf** to specify the network mapping for all interfaces.

3.3.5.1. Authenticating to the ESX / ESX(i) server

Connecting to the ESX / ESX(i) server will require authentication. **virt-v2v** supports password authentication when connecting to ESX / ESX(i). It reads passwords from **$HOME/.netrc**. The format of this file is described in the **netrc(5)** man page. An example entry is:

```
machine esx.example.com login root password s3cr3t
```

Note

The **.netrc** file must have a permission mask of 0600 to be read correctly by **virt-v2v**.

3.3.5.2. Connecting to an ESX / ESX(i) server with an invalid certificate

In non-production environments, the ESX / ESX(i) server may have a non-valid certificate, for example a self-signed certificate. In this case, certificate checking can be explicitly disabled by adding ***'?no_verify=1'*** to the connection URI as shown below:

```
... -ic esx://esx.example.com/?no_verify=1 ...
```

3.3.6. Converting a virtual machine running Windows

This example demonstrates converting a local (libvirt-managed) Xen virtual machine running Windows for output to Red Hat Enterprise Virtualization. Ensure that the virtual machine's XML is available locally, and that the storage referred to in the XML is available locally at the same paths.

To convert the guest virtual machine from an XML file, run:

```
virt-v2v -i libvirtxml -o rhev -osd storage.example.com:/exportdomain --network
rhevm guest_name.xml
```

Where ***guest_name.xml*** is the path to the virtual machine's exported XML, and **storage.example.com:/exportdomain** is the export storage domain.

You may also use the ***--network*** parameter to connect to a locally managed network if your virtual machine only has a single network interface. If your virtual machine has multiple network interfaces, edit **/etc/virt-v2v.conf** to specify the network mapping for all interfaces.

If your virtual machine uses a Xen para-virtualized kernel (it would be called something like **kernel-xen** or **kernel-xenU**), **virt-v2v** will attempt to install a new kernel during the conversion process. You can avoid this requirement by installing a regular kernel, which won't reference a hypervisor in its name, alongside the Xen kernel prior to conversion. You should not make this newly installed kernel your default kernel, because Xen will not boot it. **virt-v2v** will make it the default during conversion.

3.4. Running converted virtual machines

On successful completion, **virt-v2v** will create a new libvirt domain for the converted virtual machine with the same name as the original virtual machine. It can be started as usual using libvirt tools, for example **virt-manager**.

Note

virt-v2v cannot currently reconfigure a virtual machine's network configuration. If the converted virtual machine is not connected to the same subnet as the source, the virtual machine's network configuration may have to be updated manually.

Chapter 4. Converting virtual machines to run on Red Hat Enterprise Virtualization

virt-v2v can convert virtual machines to run on Red Hat Enterprise Virtualization. Virtual machines can be converted from Xen, KVM and VMware ESX / ESX(i) environments. Before converting virtual machines to run on Red Hat Enterprise Virtualization, you must attach an export storage domain to the Red Hat Enterprise Virtualization data center being used. Section 4.2, “Attaching an export storage domain” explains the process of attaching an export storage domain. For more information on export storage domains, see the *Red Hat Enterprise Virtualization Administration Guide*.

4.1. Acceptable converted storage output formats

It is important to note that when converting a guest virtual machine to run on Red Hat Enterprise Virtualization, not all combinations of storage format and allocation policy are supported. The supported combinations differ according to whether the Red Hat Enterprise Virtualization data center the guest will be imported into uses block (FC or iSCSI) or file (NFS) for its data storage domain. Note that **virt-v2v** writes to an *export* storage domain, and this is always required to be NFS.

Note

The important element for a successful virtual machine import into Red Hat Enterprise Virtualization is the type of the data domain. **virt-v2v** is unable to detect the data center type, so this check must be applied manually by the user.

Table 4.1. Allocation Policy: Preallocated

Data Domain Type	Storage Format	Supported
NFS	raw	Yes
	qcow2	No
FC/iSCSI	raw	Yes
	qcow2	No

Table 4.2. Allocation Policy: Sparse

Data Domain Type	Storage Format	Supported
NFS	raw	Yes
	qcow2	Yes
FC/iSCSI	raw	No
	qcow2	Yes

Data format and allocation policy of the virtual machine being converted by **virt-v2v** will be preserved unless the output data format and allocation policy are specified using the **-of** and **-oa** parameters respectively. To import virtual machines using sparse allocation into an FC or iSCSI data center, the storage format must be converted to qcow2. This is achieved by passing the parameters **-of qcow2 -oa sparse** to **virt-v2v**. Note that converting between raw and qcow2 formats is a resource intensive

operation, and roughly doubles the length of time taken for the conversion process.

Important

Preallocated qcow2 storage is never supported in Red Hat Enterprise Virtualization, although **`virt-v2v`** is able to write it. Import to Red Hat Enterprise Virtualization will fail.

4.2. Attaching an export storage domain

Before converting virtual machines to run on Red Hat Enterprise Virtualization, you must attach an export storage domain to the Red Hat Enterprise Virtualization data center being used.

An export storage domain can be attached to a data center to enable the import or export of virtual machines from one data center to another. An export storage domain can also be used to backup virtual machines and templates.

Note

At a given time, an export domain can only be attached to a single data center.

Procedure 4.1. Attaching an export storage domain

1. Log in to the Red Hat Enterprise Virtualization Administration Portal. Click the **`Data Centers`** tab. Select the data center to which the export storage domain is to be attached.
2. The details pane displays. Select the **`Storage`** tab.
3. Click **`Attach Export`** to add the storage location where the images are stored.

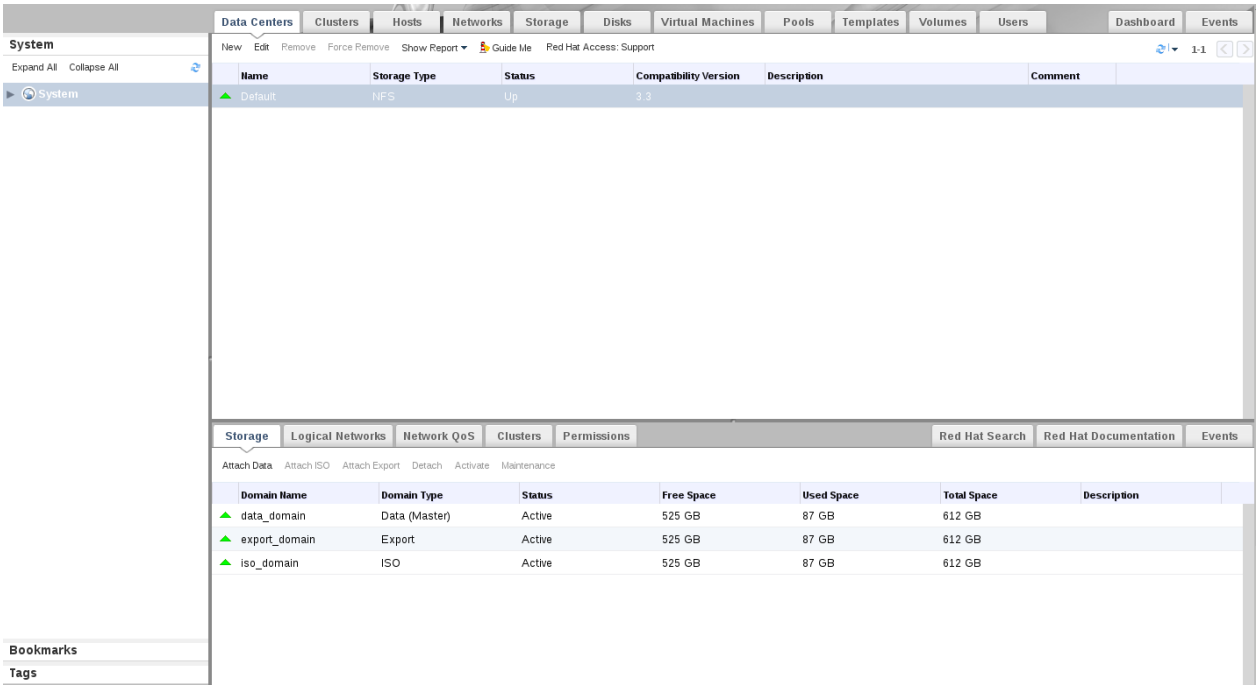

Figure 4.1. Attaching an Export Domain

4. The **`Attach Export Domain`** dialog box displays if there are export domains available. Select the export domain from the list.

.

5. Click the **OK** button. The new export storage domain displays on the **Storage** tab of the details pane with a status of *Locked*, followed by *Inactive*.
6. Select the new export storage domain on the **Storage** tab of the details pane, and click the **Activate** button.
7. The export domain will be activated in a few moments and display an *Active* status.

4.3. Converting a virtual machine

virt-v2v converts virtual machines from a foreign hypervisor to run on Red Hat Enterprise Virtualization. It automatically packages the virtual machine images and metadata, then uploads them to a Red Hat Enterprise Virtualization export storage domain. For more information on export storage domains, see Section 4.2, “Attaching an export storage domain”. **virt-v2v** always makes a copy of storage before conversion.

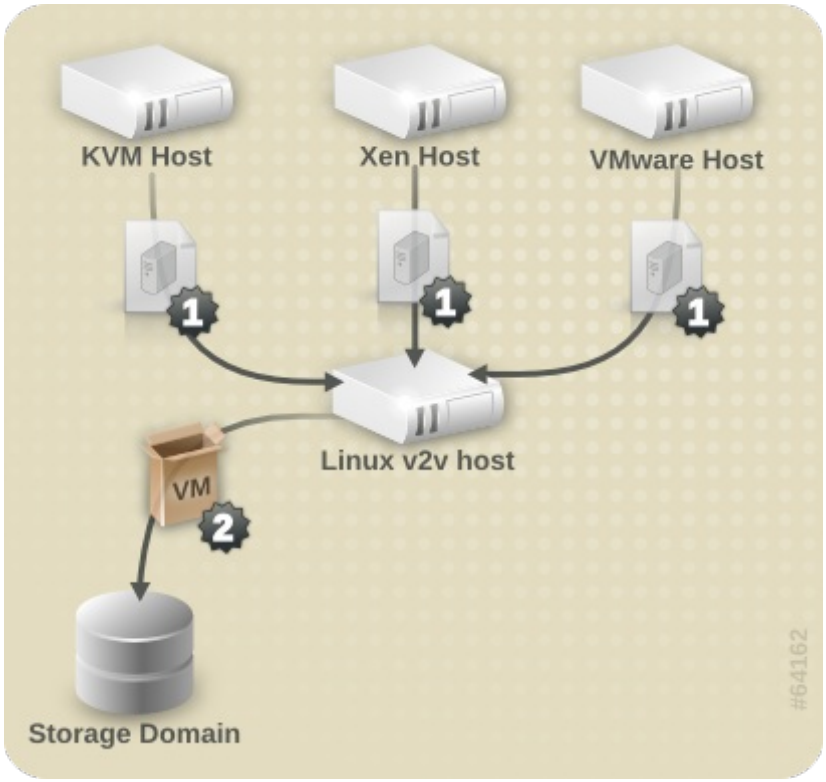

Figure 4.2. Converting a virtual machine

From the export storage domain, the virtual machine images can be imported into Red Hat Enterprise Virtualization using the Administration Portal.

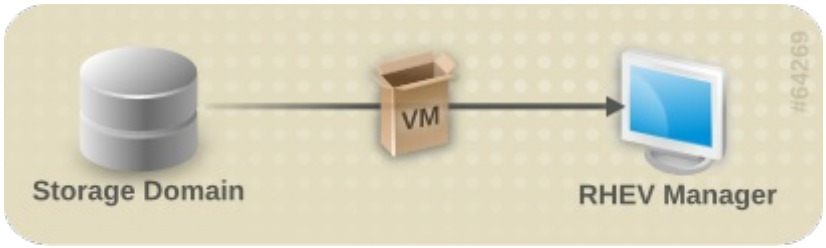

Figure 4.3. Importing a virtual machine

4.3.1. Preparing to convert a virtual machine

Before a virtual machine can be converted, ensure that the following steps are completed:

Procedure 4.2. Preparing to convert a virtual machine

1. Create an NFS export domain. **virt-v2v** can transfer the converted virtual machine directly to an NFS export storage domain. From the export storage domain, the virtual machine can be imported into a Red Hat Enterprise Virtualization data center. The storage domain must be mountable by the machine running **virt-v2v**. When exporting to a Red Hat Enterprise Virtualization export domain, **virt-v2v** must run as root.

   

   Note

   The export storage domain is accessed as an NFS share. By default, Red Hat Enterprise Linux 6 uses NFSv4, which does not require further configuration. However, for NFSv2 and NFSv3 clients, the **rpcbind** and **nfslock** services must be running on the host used to run **virt-v2v**. The network must also be configured to allow NFS access to the storage server. For more details refer to the *Red Hat Enterprise Linux Storage Administration Guide*.

2. Specify network mappings in **virt-v2v.conf**. This step is *optional*, and is not required for most use cases.

   If your virtual machine has multiple network interfaces, **/etc/virt-v2v.conf** must be edited to specify the network mapping for all interfaces. You can specify an alternative **virt-v2v.conf** file with the ***-f*** parameter. If you are converting to a virtual machine for output to both libvirt and Red Hat Enterprise Virtualization, separate **virt-v2v.conf** files should be used for each conversion. This is because a converted bridge will require different configuration depending on the output type (libvirt or Red Hat Enterprise Virtualization).

   If your virtual machine only has a single network interface, it is simpler to use the ***--network*** or ***--bridge*** parameters, rather than modifying **virt-v2v.conf**.

3. Create a profile for the conversion in **virt-v2v.conf**. This step is *optional*. Profiles specify a conversion method, storage location, output format and allocation policy. When a profile is defined, it can be called using ***--profile*** rather than individually providing the ***-o***, ***-os***, ***-of*** and ***-oa*** parameters. See virt-v2v.conf(5) for details.

4.3.1.1. Preparing to convert a virtual machine running Linux

The following is required when converting virtual machines which run Linux, regardless of which hypervisor they are being converted from.

Procedure 4.3. Preparing to convert a virtual machine running Linux

1. Obtain the software.

   As part of the conversion process, **virt-v2v** may install a new kernel and drivers on the virtual machine. If the virtual machine being converted is registered to Red Hat Network (RHN), the required packages will be automatically downloaded. For environments where RHN is not available, the **virt-v2v.conf** file references a list of RPMs used for this purpose. The RPMs relevant to your virtual machine must be downloaded manually from RHN and made available in the directory specified by the ***path-root*** configuration element, which by default is **/var/lib/virt-v2v/software/**. **virt-v2v** will display an error similar to Example 3.1,

"Missing Package error" if the software it depends upon for a particular conversion is not available.

2. To obtain the relevant RPMs for your environment, repeat these steps for each missing package:
   a. Log in to the Red Hat Customer Portal: https://access.redhat.com/.
   b. In the Red Hat Customer Portal, select the **Downloads** tab to open the **Software & Download Center** page.
   c. Select **Packages** to enter the **Package Search** page (https://rhn.redhat.com/rhn/channels/software/Search.do).
   d. In the **Search For** field, type the package name exactly matching the one shown in the error message. For the example shown in Example 3.1, "Missing Package error", the first package is **kernel-2.6.32-128.el6.x86_64**
   e. In the **Where to search** field, select **In the following architectures** and select the **x86_64** checkbox. Click **Search**.
   f. A list of packages displays. Select the package name identical to the one in the error message. This opens the details page, which contains a detailed description of the package.
   g. Select **Download Package** at the bottom of the page.
   h. Save the downloaded package to the appropriate directory in **/var/lib/virt-v2v/software**. For Red Hat Enterprise Linux 6, the directory is **/var/lib/virt-v2v/software/rhel/6**.

4.3.1.2. Preparing to convert a virtual machine running Windows

Important

virt-v2v does not support conversion of the Windows Recovery Console. If a virtual machine has a recovery console installed and VirtIO was enabled during conversion, attempting to boot the recovery console will result in a stop error.
Windows XP x86 does not support the Windows Recovery Console on VirtIO systems, so there is no resolution to this. However, on Windows XP AMD64 and Windows 2003 (x86 and AMD64), the recovery console can be re-installed after conversion. The re-installation procedure is the same as the initial installation procedure. It is not necessary to remove the recovery console first. Following re-installation, the recovery console will work as intended.

Important

When converting a virtual machine running Windows with multiple drives, for output to Red Hat Enterprise Virtualization, it is possible in certain circumstances that additional drives will not be displayed by default. Red Hat Enterprise Virtualization will always add a CD-ROM device to a converted virtual machine. If the virtual machine did not have a CD-ROM device before conversion, the new CD-ROM device may be assigned a drive letter which clashes with an existing drive on the virtual machine. This will render the existing device inaccessible. The occluded disk device can still be accessed by manually assigning it a new drive letter. It is also possible to maintain previous drive letter assignment by manually changing the drive letter assigned to the new CD-ROM device, then rebooting the virtual machine.

The following is required when converting virtual machines running Windows, regardless of which hypervisor they are being converted from. The conversion procedure depends on post-processing by

the Red Hat Enterprise Virtualization Manager for completion. See Section 7.2.2, “Configuration changes for Windows virtual machines” for details of the process.

Procedure 4.4. Preparing to convert a virtual machine running Windows

Before a virtual machine running Windows can be converted, ensure that the following steps are completed.

1. Install the *libguestfs-winsupport* package on the host running **virt-v2v**.

   This package provides support for NTFS, which is used by many Windows systems. The *libguestfs-winsupport* package is provided by the **RHEL V2VWIN (v. 6 for 64-bit x86_64)** channel. Ensure your system is subscribed to this channel, then run the following command as root:

   ```
   yum install libguestfs-winsupport
   ```

   If you attempt to convert a virtual machine using NTFS without the *libguestfs-winsupport* package installed, the conversion will fail. An error message similar to Example 4.1, “Error message when converting a Windows virtual machine without libguestfs-winsupport installed” will be shown:

   Example 4.1. Error message when converting a Windows virtual machine without libguestfs-winsupport installed

   ```
   No operating system could be detected inside this disk image.

   This may be because the file is not a disk image, or is not a virtual
   machine image, or because the OS type is not understood by virt-inspector.

   If you feel this is an error, please file a bug report including as much
   information about the disk image as possible.
   ```

2. Install the *virtio-win* package on the host running **virt-v2v**.

   This package provides para-virtualized block and network drivers for Windows guests. The *virtio-win* package is provided by the **RHEL Server Supplementary (v. 6 64-bit x86_64)** channel. Ensure your system is subscribed to this channel, then run the following command as root:

   ```
   yum install virtio-win
   ```

   If you attempt to convert a virtual machine running Windows without the *virtio-win* package installed, the conversion will fail. An error message similar to Example 3.3, “Error message when converting a Windows virtual machine without virtio-win installed” will be shown.

3. Upload the guest tools ISO to the ISO Storage Domain.

   Note that the guest tools ISO is not required for the conversion process to succeed. However, it is recommended for all Windows virtual machines running on Red Hat Enterprise Virtualization. The Red Hat Enterprise Virtualization Manager installs Red Hat's Windows drivers on the guest virtual machine using the guest tools ISO after the virtual machines have been converted. See Section 7.2.2, “Configuration changes for Windows virtual machines” for details.

   Locate and upload the guest tools ISO as follows:

   a. Locate the guest tools ISO.

      The guest tools ISO is distributed using the Red Hat Network as *rhev-guest-tools-iso.rpm*, an RPM file installed on the Red Hat Enterprise Virtualization Manager. After installing the Red Hat Enterprise Virtualization Manager, the guest tools ISO can be found at

/usr/share/rhev-guest-tools-iso/rhev-tools-setup.iso.

b. Upload the guest tools ISO.

Upload the guest tools ISO to the ISO Storage Domain using the ISO uploader.

Refer to the *Red Hat Enterprise Virtualization Administration Guide* for more information on uploading ISO files, and installing guest agents and drivers.

4.3.1.3. Preparing to convert a local Xen virtual machine

The following is required when converting virtual machines on a host which used to run Xen, but has been updated to run KVM. It is not required when converting a Xen virtual machine imported directly from a running libvirt/Xen instance.

Procedure 4.5. Preparing to convert a local Xen virtual machine

- Obtain the XML for the virtual machine.

  virt-v2v uses a libvirt domain description to determine the current configuration of the virtual machine, including the location of its storage. Before starting the conversion, obtain this from the host running the virtual machine with the following command:

  ```
  virsh dumpxml guest_name > guest_name.xml
  ```

  This will require booting into a Xen kernel to obtain the XML, as libvirt needs to connect to a running Xen hypervisor to obtain its metadata. The conversion process is optimized for KVM, so obtaining domain data while running a Xen kernel, then performing the conversion using a KVM kernel will be more efficient than running the conversion on a Xen kernel.

4.3.2. Converting a virtual machine

Once you have prepared to convert the virtual machines, use **virt-v2v** to perform the actual conversions. This section provides the steps to convert the virtual machines, and the command syntax for **virt-v2v**.

Note that conversions are resource intensive processes that require copying the whole disk image for a virtual machine. In typical environments, converting a single virtual machine takes approximately 5-10 minutes. In Example 4.2, “Typical virt-v2v conversion time” a virtual machine with a single 8GB disk is copied over SSH on a 1GigE network on three-year-old consumer hardware:

> **Example 4.2. Typical virt-v2v conversion time**
>
> ```
> win2k3r2-pv-32.img: 100% [=========================================]D
> 0h02m57s
> virt-v2v: win2k3r2-pv-32 configured with virtio drivers.
> ```

The size of the disk to be copied is the major factor in determining conversion time. For a virtual machine on average hardware with a single disk of 20GB or less, a conversion usually takes less than 10 minutes.

4.3.2.1. virt-v2v

virt-v2v converts virtual machines from a foreign hypervisor to run on Red Hat Enterprise Virtualization. The general command syntax for converting machines to run on Red Hat Enterprise Virtualization is:

```
virt-v2v -i libvirtxml -o rhev -os storage.example.com:/exportdomain --network
rhevm guest_name.xml
virt-v2v -o rhev -os storage.example.com:/exportdomain --network rhevm guest_name
virt-v2v -ic esx://esx.example.com/?no_verify=1 -o rhev -os
storage.example.com:/exportdomain --network rhevm guest_name
```

A full specification of the parameters which can be used with **virt-v2v** is available in Section 7.1, “virt-v2v Parameters”.

Important

Presently, **virt-v2v** does not support converting a virtual machine with more than 4 IDE disks to Red Hat Enterprise Virtualization. Attempting to convert a virtual machine with 5 or more disks will fail.

4.3.2.2. Converting a local Xen virtual machine

Ensure that the virtual machine's XML is available locally, and that the storage referred to in the XML is available locally at the same paths.

To convert the virtual machine from an XML file, run:

```
virt-v2v -i libvirtxml -o rhev -os storage.example.com:/exportdomain --network
rhevm guest_name.xml
```

Where **storage.example.com:/exportdomain** is the export storage domain, **rhevm** is the locally managed network to connect the converted virtual machine's network to, and **guest_name.xml** is the path to the virtual machine's exported xml.

You may also use the ***--bridge*** parameter to connect to a locally managed network if your virtual machine only has a single network interface. If your virtual machine has multiple network interfaces, edit **/etc/virt-v2v.conf** to specify the network mapping for all interfaces.

To convert the virtual machine from a running Xen hypervisor, run:

```
virt-v2v -ic xen:/// -o rhev -os storage.example.com:/exportdomain --network
rhevm guest_name
```

Where **storage.example.com:/exportdomain** is the export storage domain, **rhevm** is the locally managed network to connect the converted virtual machine's network to, and ***guest_name*** is the name of the Xen virtual machine.

You may also use the ***--bridge*** parameter to connect to a locally managed network if your virtual machine only has a single network interface. If your virtual machine has multiple network interfaces, edit **/etc/virt-v2v.conf** to specify the network mapping for all interfaces.

If your guest uses a Xen para-virtualized kernel (it would be called something like **kernel-xen** or **kernel-xenU**), **virt-v2v** will attempt to install a new kernel during the conversion process. You can avoid this requirement by installing a regular kernel, which will not reference a hypervisor in its name, alongside the Xen kernel prior to conversion. You should not make this newly installed kernel your default kernel, because Xen will not boot it. **virt-v2v** will make it the default during conversion.

4.3.2.3. Converting a remote Xen virtual machine

Xen virtual machines can be converted remotely using SSH. Ensure that the host running the virtual machine is accessible via SSH. Even on a guest with multiple disks, each virtual disk transfer requires a separate SSH session.

Important

It is recommended to set up SSH keys for authentication prior to the remote virtual machine conversion. Otherwise, a user will be required to manually enter SSH credentials for each guest disk being transferred. Failure to enter a password manually in the time after the transfer completes but before the SSH negotiation times out will cause **virt-v2v** to fail. This is especially important for large disks, as the disk transfer can take an unspecified length of time.

To convert the virtual machine, run:

```
virt-v2v -o rhev -ic xen+ssh://root@vmhost.example.com -os
storage.example.com:/exportdomain --network rhevm guest_name
```

Where **vmhost.example.com** is the host running the virtual machine, **storage.example.com:/exportdomain** is the export storage domain, **rhevm** is the locally managed network to connect the converted virtual machine's network to, and ***guest_name*** is the name of the Xen virtual machine.

You may also use the ***--bridge*** parameter to connect to a locally managed network if your virtual machine only has a single network interface. If your virtual machine has multiple network interfaces, edit **/etc/virt-v2v.conf** to specify the network mapping for all interfaces.

If your guest uses a Xen para-virtualized kernel (it would be called something like **kernel-xen** or **kernel-xenU**), **virt-v2v** will attempt to install a new kernel during the conversion process. You can avoid this requirement by installing a regular kernel, which will not reference a hypervisor in its name, alongside the Xen kernel prior to conversion. You should not make this newly installed kernel your default kernel, because Xen will not boot it. **virt-v2v** will make it the default during conversion.

4.3.2.4. Converting a local KVM virtual machine

Use the following procedure to convert a local KVM virtual machine:

Procedure 4.6. Converting a local KVM virtual machine

1. **Stop the virtual machine.**
   a. Ensure that the virtual machine is stopped prior to running the v2v process. If the virtual machine is in a clustered Red Hat Enterprise Linux HA virtual machine environment, stop and disable the virtual machine resource in cluster node using the command:

      ```
      clusvcadm -d vm:<guest>
      ```

   b. After stopping the virtual machine, run **virsh define <path-to-guest.xml>** to place the stopped virtual machine under the control of libvirt. This command allows **virt-v2v** to recognize the virtual machine and enable it to be converted.
2. **Convert the virtual machine.**
   To convert the virtual machine, run:

```
virt-v2v -o rhev -os storage.example.com:/exportdomain --network rhevm
guest_name
```

Where **storage.example.com:/exportdomain** is the export storage domain, **rhevm** is the locally managed network to connect the converted virtual machine's network to, and ***guest_name*** is the name of the KVM virtual machine.

You may also use the ***--bridge*** parameter to connect to a locally managed network if your virtual machine only has a single network interface. If your virtual machine has multiple network interfaces, edit **/etc/virt-v2v.conf** to specify the network mapping for all interfaces.

4.3.2.5. Converting a remote KVM virtual machine

KVM virtual machines can be converted remotely via SSH. Ensure that the host running the virtual machine is accessible via SSH, and that the virtual machine is stopped prior to running the v2v process. Even on a guest with multiple disks, each virtual disk transfer requires a separate SSH session.

Important

It is recommended to set up SSH keys for authentication prior to the remote virtual machine conversion. Otherwise, a user will be required to manually enter SSH credentials for each guest disk being transferred. Failure to enter a password manually in the time after the transfer completes but before the SSH negotiation times out will cause **virt-v2v** to fail. This is especially important for large disks, as the disk transfer can take an unspecified length of time.

To convert the virtual machine, run:

```
virt-v2v -ic qemu+ssh://root@kvmhost.example.com/system -o rhev -os
storage.example.com:/exportdomain --network rhevm guest_name
```

Where **kvmhost.example.com** is the host running the virtual machine, **storage.example.com:/exportdomain** is the export storage domain, **rhevm** is the locally managed network to connect the converted virtual machine's network to, and ***guest_name*** is the name of the KVM virtual machine.

You may also use the ***--bridge*** parameter to connect to a locally managed network if your virtual machine only has a single network interface. If your virtual machine has multiple network interfaces, edit **/etc/virt-v2v.conf** to specify the network mapping for all interfaces.

4.3.2.6. Converting a VMware ESX / ESX(i) virtual machine

Important

When converting a Windows virtual machine from VMware ESX / ESX(i), ensure that VMware Tools is not installed on the virtual machine. The VMware Tools must be uninstalled prior to the conversion process. If a virtual machine is converted with the VMware Tools installed, it will not function correctly.

Ensure that the virtual machine is stopped prior to running the v2v process.

To convert the virtual machine, run:

```
virt-v2v -ic esx://esx.example.com/ -o rhev -os storage.example.com:/exportdomain
--network rhevm guest_name
```

Where **storage.example.com:/exportdomain** is the export storage domain, **rhevm** is the locally managed network to connect the converted virtual machine's network to, and ***guest_name*** is the name of the virtual machine.

You may also use the ***--bridge*** parameter to connect to a locally managed network if your virtual machine only has a single network interface. If your virtual machine has multiple network interfaces, edit **/etc/virt-v2v.conf** to specify the network mapping for all interfaces.

Authenticating to the ESX / ESX(i) server

Connecting to the ESX / ESX(i) server will require authentication. **virt-v2v** supports password authentication when connecting to ESX / ESX(i). It reads passwords from **$HOME/.netrc**. The format of this file is described in **netrc(5)**. An example entry is:

```
machine esx.example.com login root password s3cr3t
```

Note

The **.netrc** file must have a permission mask of 0600 to be read correctly by **virt-v2v**.

Connecting to an ESX / ESX(i) server with an invalid certificate

In non-production environments, the ESX / ESX(i) server may have a non-valid certificate, for example a self-signed certificate. In this case, certificate checking can be explicitly disabled by adding ***?no_verify=1*** to the connection URI as shown below:

```
... -ic esx://esx.example.com/?no_verify=1 ...
```

4.3.2.7. Converting a VMware VMDK virtual machine

Important

Before a VMware VMDK virtual machine can be converted to run on Red Hat Enterprise Virtualization, the virtual machine must be exported to OVA format using the VMware OVF Tool. For details on exporting a VMDK virtual machine to OVA format, refer to the following documentation: http://www.vmware.com/support/developer/ovf/.

When converting an OVA virtual machine, it is not necessary to stop the virtual machine to perform the conversion. However, if the original guest continues to run, the states of the original guest and the converted guest will diverge.

To convert the virtual machine to Red Hat Enterprise Virtualization using the OVA image file, run:

```
virt-v2v -i ova --profile myrhevserver esx_guest.ova
```

Where **myrhevserver** is the Red Hat Enterprise Virtualization server, and **esx_guest** is the name of the ESX / ESX (i) virtual machine.

4.3.3. Importing and running the converted virtual machine

On successful completion, **virt-v2v** will upload the exported virtual machine to the specified export domain. To import and run the converted virtual machine:

Procedure 4.7. Importing and running the converted virtual machine

1. In the Red Hat Enterprise Virtualization Administration Portal, select the export domain from the **Storage** tab. The export domain must have a status of **Active**.
2. Select the **VM Import** tab in the details pane to list the available virtual machines to import.
3. Select one or more virtual machines and click **Import**. The **Import Virtual Machine(s)** window will open.
4. In the drop-down menus, select the select the **Default Storage Domain**, **Cluster**, and **Cluster Quota** in the data center.
5. Select the **Collapse Snapshots** check box to remove snapshot restore points and include templates in template-based virtual machines. Click **OK** to import the virtual machines.

For more information on importing virtual machines, refer to the *Red Hat Enterprise Virtualization Administration Guide*.

Network configuration

virt-v2v cannot currently reconfigure a guest's network configuration. If the converted guest is not connected to the same subnet as the source, the guest's network configuration may have to be updated manually.

4.3.4. Scripting the v2v process

The entire v2v process can be scripted, enabling the automated batch processing of a large number of virtual machines. The process is broken up into two steps, which must be run on separate hosts.

Procedure 4.8. Scripting the v2v process

1. Use **virt-v2v** to convert the virtual machines and copy them to the export storage domain. This step must be run on a Linux host. The process is detailed in Section 4.3.2, “Converting a virtual machine”.
2. Once the conversion is complete, use the Red Hat Enterprise Virtualization Administration Portal to import the virtual machines from the export storage domain. This step must be run on the Red Hat Enterprise Virtualization Manager server.

   For more information on importing a virtual machine with the Red Hat Enterprise Virtualization Administration Portal, see the *Red Hat Enterprise Virtualization Administration Guide*.

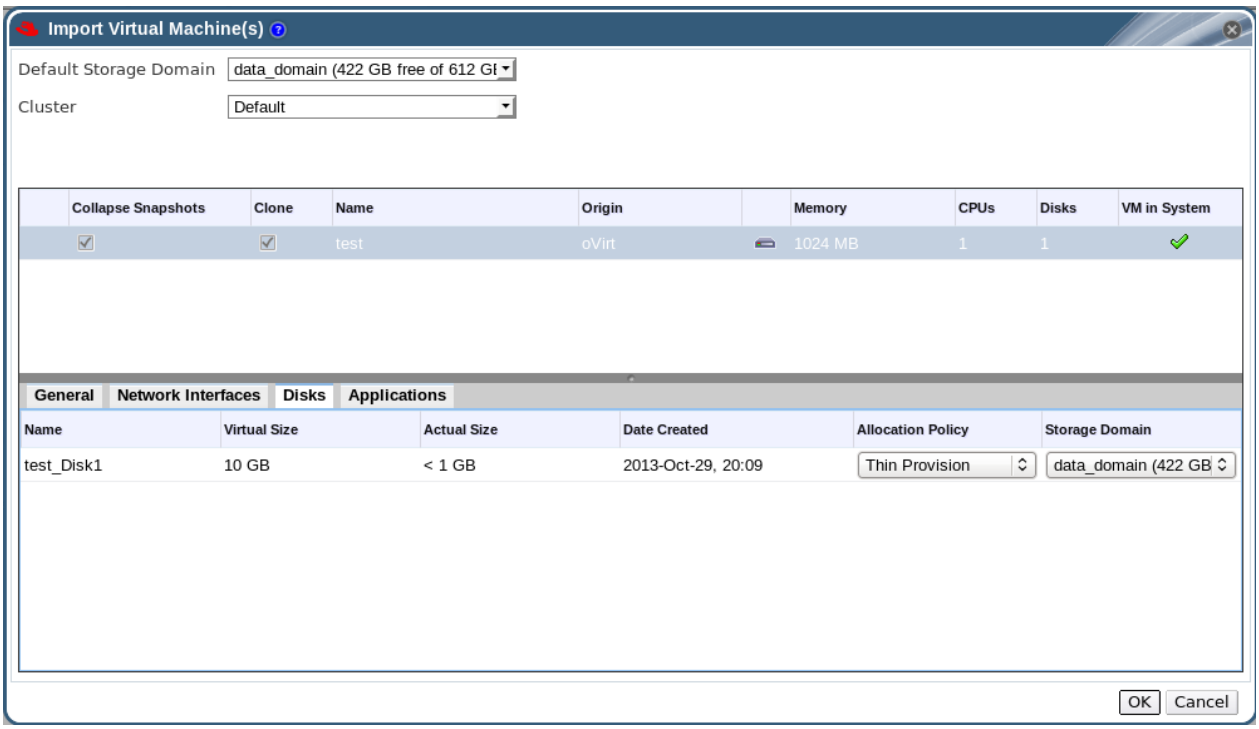

Figure 4.4. Importing a virtual machine with the Red Hat Enterprise Virtualization Administration Portal

Alternatively, the Python SDK or the command line can also be used to import the virtual machines from the export storage domain:

To import the virtual machines using the SDK, use the following:

Example 4.3. Importing virtual machines from the export storage domain using the SDK

```
api = API(url="http(s)://...:.../api",
          username="...",
          password="...",
          filter=False,
          debug=True)


sd = api.storagedomains.get(id="from-sd-id")
import_candidate = sd.vms.get(id="vm-to-import")
import_candidate.import_vm(action=params.Action(
cluster=api.clusters.get(id="to-cluster-id"),
storage_domain=api.storagedomains.get(id="to-sd-id")))
```

Note

When using the SDK method, entities can also be fetched and passed using ***name=***.

To import the virtual machines using the command line, connect to the Red Hat Enterprise Virtualization Manager shell and use the following command:

Example 4.4. Importing virtual machines from the export storage domain using the command line

```
action vm "vm-to-import" import_vm --storagedomain-identifier "from-sd-
id" --cluster-id "to-cluster-id" --storage_domain-id "to-sd-id"
```

Note

When using the command line method, entities can also be fetched and passed using ***-name***.

4.3.5. Scripted bulk v2v process

For bulk import scenarios, it is advantageous to be able to perform the scripted v2v process from a single host. Remote procedure calls to the Red Hat Enterprise Virtualization Manager can be made using the REST API. This enables a single script running on a single Linux host to perform both steps of the v2v process. Figure 4.5, “Scripted bulk v2v process” illustrates the steps performed by the script.

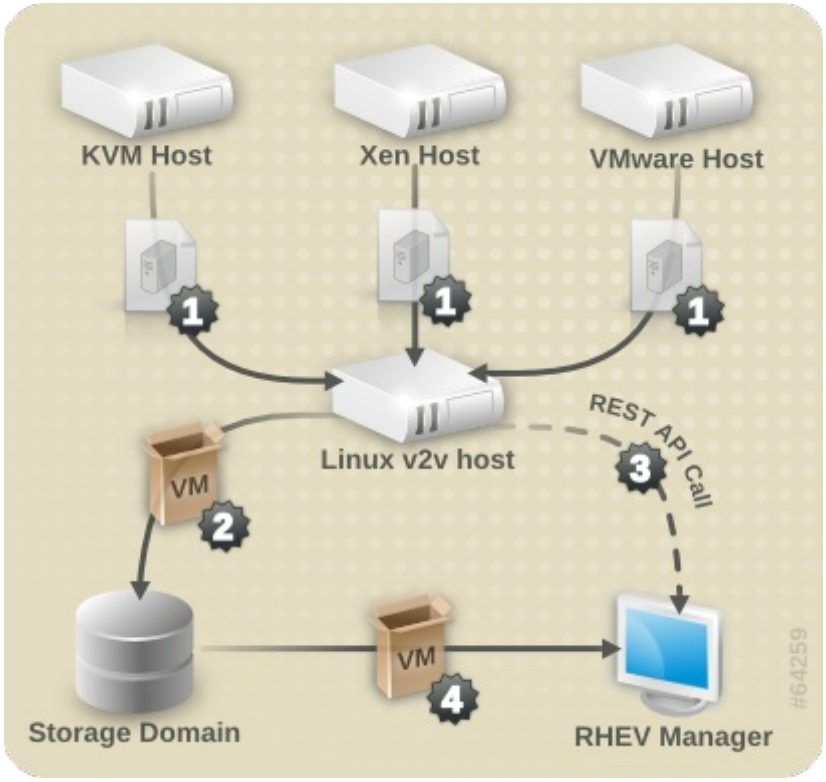

Figure 4.5. Scripted bulk v2v process

The scripted bulk v2v process involves the following steps, as shown in Figure 4.5, “Scripted bulk v2v process”:

1. The virtual machine image is retrieved from the source hypervisor.
2. The virtual machine image is packaged and copied to the export storage domain.
3. A remote procedure call is made to the Red Hat Enterprise Virtualization Manager, telling it to import the virtual machine.

4. The Manager imports the virtual machine from the export storage domain.

To configure and run the scripted bulk v2v process:

Procedure 4.9. Configuring and running the scripted bulk v2v process

1. Ensure the REST API is enabled on the Red Hat Enterprise Virtualization Manager, and it is accessible from the Linux host running the v2v process. For more information about the REST API, see the *Red Hat Enterprise Virtualization Developer Guide*.
2. On the Linux host, create the file **v2v.sh** with the following contents. Ensure you edit the script to contain appropriate values for your environment.

Example 4.5. Single host v2v script

```
#!/bin/sh
# Declare all VMs to import
XENDOMAINS=("rhelxen" "rhel5")
KVMDOMAINS=("rhelkvm")
VMWAREVMS=("rhel54vmware")

# Iterate through each Xen domain, performing the conversion
for domain in ${XENDOMAINS[@]}
do
        virt-v2v -ic xen:///localhost -o rhev -os
storage.example.com:/exportdomain --network rhevm $domain
done

# Iterate through each KVM domain, performing the conversion
for domain in ${KVMDOMAINS[@]}
do
        virt-v2v -o rhev -os storage.example.com:/exportdomain --network
rhevm $domain
done

# Iterate through each VMware VM, performing the conversion
for vm in ${VMWAREVMS[@]}
do
        virt-v2v -ic esx://esx.example.com/?no_verify=1 -o rhev -os
storage.example.com:/exportdomain --network rhevm $vm
done


# Call the import VM procedure remotely on the RHEV Manager

export BASE_URL='https://[rhevm-host]'
export HTTP_USER='user@internal'
export HTTP_PASSWORD='password'

curl -o rhevm.cer http://[rhevm-host]/ca.crt

# Get the export storage domains

curl -X GET -H "Accept: application/xml" -u
"${HTTP_USER}:${HTTP_PASSWORD}" --cacert rhevm.cer
${BASE_URL}/api/storagedomains?search=nfs_export -o exportdomain
EXPORT_DOMAIN=`xpath exportdomain '/storage_domains/storage_domain/@id' |
sed -e 's/ id=//' | sed -e 's/"//g'`

# Get the datacenter

curl -X GET -H "Accept: application/xml" -u
"${HTTP_USER}:${HTTP_PASSWORD}" --cacert rhevm.cer
${BASE_URL}/api/datacenters?search=NFS -o dc
DC=`xpath dc '/data_centers/data_center/@id' | sed -e 's/ id=//' | sed -e
's/"//g'`

# Get the cluster

curl -X GET -H "Accept: application/xml" -u
"${HTTP_USER}:${HTTP_PASSWORD}" --cacert rhevm.cer
${BASE_URL}/api/clusters?search=NFS -o cluster
CLUSTER_ELEMENT=`xpath cluster '/clusters/cluster/@id' | sed -e 's/ id=//'
| sed -e 's/"//g'`
```

```
# List contents of export storage domain

curl -X GET -H "Accept: application/xml" -u
"${HTTP_USER}:${HTTP_PASSWORD}" --cacert rhevm.cer
${BASE_URL}/api/storagedomains/${EXPORT_DOMAIN}/vms -o vms

# For each vm, export
VMS=`xpath vms '/vms/vm/actions/link[@rel="import"]/@href' | sed -e 's/
href="//g' | sed -e 's/"/ /g'`

for vms in $VMS
do
        curl -v -u "${HTTP_USER}:${HTTP_PASSWORD}" -H "Content-type:
application/xml" -d
'<action><cluster><name>cluster_name</name></cluster><storage_domain><name>
data_domain</name></storage_domain><overwrite>true</overwrite><discard_sna
pshots>true</discard_snapshots></action>' --cacert rhevm.cer
${BASE_URL}$vms
done
```

Note

Use the **POST** method to export virtual machines with the REST API. For more information about using the REST API, see the *Red Hat Enterprise Virtualization Developer Guide*.

3. Run the **v2v.sh** script. It can take several hours to convert and import a large number of virtual machines.

Chapter 5. Converting physical machines to virtual machines

Read this chapter for information about converting physical machines to virtual machines with the Red Hat Physical-to-Virtual (P2V) solution, Virt P2V.

Virt P2V is comprised of both **virt-p2v-server**, included in the *virt-v2v* package, and the P2V client, available from the Red Hat Customer Portal as **rhel-6.x-p2v.iso**. **rhel-6.x-p2v.iso** is a bootable disk image based on a customized Red Hat Enterprise Linux 6 image. Booting a machine from **rhel-6.x-p2v.iso** and connecting to a V2V conversion server that has **virt-v2v** installed allows data from the physical machine to be uploaded to the conversion server and converted for use with either Red Hat Enterprise Virtualization, or KVM managed by **libvirt**.

Note that the host **must** be running Red Hat Enterprise Linux 6. Other host configurations will not work.

Important

Adhere to the following rules. Failure to do so may cause the loss of data and disk malfunction:

- The Physical to Virtual (P2V) feature requires a Red Hat Enterprise Linux 6 virtualization host with *virt-v2v-0.8.7* or later. You can check your version of virt-v2v by running **$ rpm -q virt-v2v**.
- Note that you **cannot** convert to a Red Hat Enterprise Linux 5 conversion server, or with a *virt-v2v* package previous to version 0.8.7-6.el6.
- A number of operating systems can be converted from physical machines to virtual machines, but be aware that there are known issues converting physical machines using software RAID. Red Hat Enterprise Linux 6 machines with a filesystem root on a software RAID md device may be converted to guest virtual machines. Red Hat Enterprise Linux 4 and Red Hat Enterprise Linux 5 physical machines with their filesystem root on a software RAID md device **cannot** be converted to virtual machines. There is currently no workaround available.

5.1. Prerequisites

For a physical machine to be converted using the P2V client, it must meet basic hardware requirements in order to successfully boot the P2V client:

- Must be bootable from PXE, Optical Media (CD, DVD), or USB.
- At least 512 MB of RAM.
- An ethernet connection.
- Console access (keyboard, video, mouse).
- An operating system supported by **virt-v2v**:
  - Red Hat Enterprise Linux 3.9
  - Red Hat Enterprise Linux 4
  - Red Hat Enterprise Linux 5
  - Red Hat Enterprise Linux 6
  - Windows XP
  - Windows Vista
  - Windows 7
  - Windows Server 2003

- Windows Server 2008

5.2. Preparing to convert a physical machine

Before you use P2V, you must first prepare your conversion server and download and prepare the **rhel-6.x-p2v.iso** boot media. For full instructions see the *Red Hat Enterprise Linux Installation Guide*. Note that there is one ISO image for both i386 and x86_64 architectures.

5.2.1. Install virt-v2v on a conversion server

A conversion server is any physical server running Red Hat Enterprise Linux 6 or higher with the **virt-v2v** package installed on it. To install **virt-v2v** follow the instructions in Chapter 2, *Installing virt-v2v*. **virt-v2v** version 0.8.7-6 or higher is required.

5.2.2. Enable root login over SSH

Now that **virt-v2v** is installed, the conversion server must be prepared to accept P2V client connections. The P2V client connects to the conversion server as root using SSH, so root login over SSH must be allowed on the conversion server.

Enable root login over SSH:

1. As root, edit the sshd_config file in **/etc/ssh/sshd_config**:

```
nano /etc/ssh/sshd_config
```

2. Add a line in the Authentication section of the file that says ***PermitRootLogin yes***. This line may already exist and be commented out with a "#". In this case, remove the "#".

```
# Authentication:
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

3. Save the updated **/etc/ssh/sshd_config** file.
4. Restart the SSH server:

```
service sshd restart
```

You can now connect to the conversion server as root over SSH.

5.2.3. Define a target profile in `virt-v2v.conf`

Now that you are able to connect to the conversion server as root, it must be pre-configured with details about what to do with the virtual machine it creates. These details are given as a target profile in the **/etc/virt-v2v.conf** file on the conversion server.

Define a target profile in `virt-v2v.conf`:

1. As root, edit **/etc/virt-v2v.conf**:

```
nano /etc/virt-v2v.conf
```

2. Scroll to the end of the file. Before the final **</virt-v2v>**, add the following:

```
<profile name="myrhev">
<method>rhev</method>
<storage format="raw" allocation="preallocated">
nfs.share.com:/export1
</storage>
<network type="default">
<network type="network" name="rhevm"/>
</network>
</profile>
```

Where:

- **Profile Name** is an arbitrary, descriptive target profile name.
- **Method** is the destination hypervisor type (rhev or libvirt).
- **Storage Format** is the output storage format, either raw or qcow2.
- **Allocation** is the output allocation policy, either preallocated or sparse.
- **Network type** specifies the network to which a network interface should be connected when imported into Red Hat Enterprise Virtualization. The first network type entry contains details about network configuration before conversion, the second network type entry maps to an after conversion configuration. In the given example, any detected network card is to be mapped to the managed network called rhevm.

Important

The value associated with the **<storage format>** tag (in the above example "nfs.share.com:/export1") must match the value associated with the **<method>** tag. In this example, since the output method is "rhev", the value associated with storage must be an initialized NFS share. For the libvirt method, the storage format value must be an initialized storage domain that exists locally on the conversion server, for example "default".

You have created a target profile that defines what will happen to the virtual machine that results from this P2V conversion.

5.2.4. Create the bootable media

P2V uses bootable media to create a bootable image of the hard drive of a physical machine and send it to the conversion server for import into a hypervisor. You will need a Red Hat Network account to download the **rhel-6.x-p2v.iso** ISO. Follow the instructions on preparing bootable media from the *Red Hat Enterprise Linux Installation Guide*. Note that there is only one ISO image for both i386 and x86_64 architectures.

The latest release of **rhel-6.x-p2v.iso** can be found at https://rhn.redhat.com/rhn/software/channel/downloads/Download.do?cid=10358

Create the appropriate bootable media:

The **rhel-6.x-p2v.iso** file can be used three ways: as a bootable disk, a PXE boot image, and as a bootable USB device.

- Burn the ISO to a blank CD-ROM or DVD-ROM, and insert it into the disk drive of the physical machine that is to be converted. To boot properly from this boot media, some changes to BIOS settings may be required to ensure that the optical disk drive is first in the boot order.

- Use the ISO to create a bootable USB media. To boot properly from this boot media, some changes to BIOS settings may be required to ensure that the USB device is first in the boot order. In addition, some BIOSes do not support booting from USB media. The P2V client disk image is approximately 100 MB, so the USB device must be large enough to hold the disk image.
- Prepare a PXE Boot image for your existing PXE server. To boot from PXE, some changes to BIOS settings may be required to ensure that the PXE boot is first in the boot order.

More information about creating boot media can be found in Appendix A, *Additional procedures*.

You have finished preparing and are now ready to move on to converting a physical machine to a virtual machine.

5.3. Converting physical machines to virtual machines

By now you have prepared a conversion server and your chosen bootable media. Next you will boot the P2V client, connect to your conversion server, and convert your physical machine. The conversion process can take a long time, especially when large physical disks are being converted.

Converting a physical machine to a virtual machine:

1. Boot the P2V client boot media that you created in preparation for your conversion. Because the P2V client is built upon a Red Hat Enterprise Linux 6 live image, the Red Hat Enterprise Linux 6 splash image is displayed while the tool is booting.
2. Configure networking. Generally the P2V client configures networking automatically using DHCP. If it is unable to configure networking automatically, you will need to configure it manually. **`IP Address`**, **`Gateway`**, and **`Prefix`** are required fields. Enter values that are appropriate for your network, and click **`Use these network settings`**.

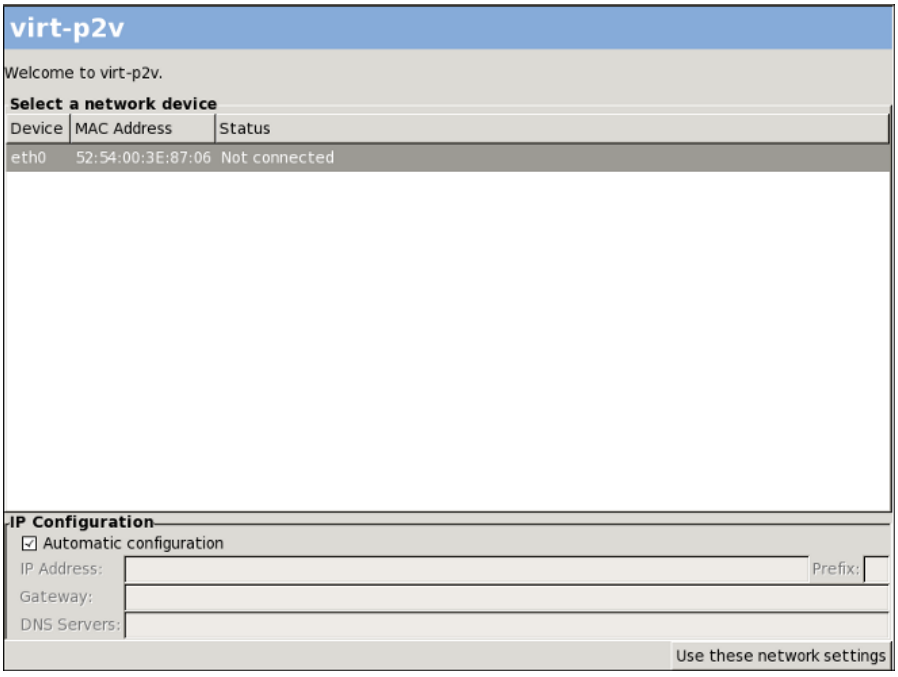

Figure 5.1. Network configuration on the P2V client

Important

If for some reason manual configuration is preferred, disconnect the physical machine from the network entirely before booting. Then you will be able to enter a networking configuration directly and reconnect the physical machine to the network.

3. When networking is configured, you are prompted for connection details for the conversion server. These include **`Hostname`** (either hostname or IP address), **`Username`** (this must be root), and **`Password`**. Enter these details and click **`Connect`**.

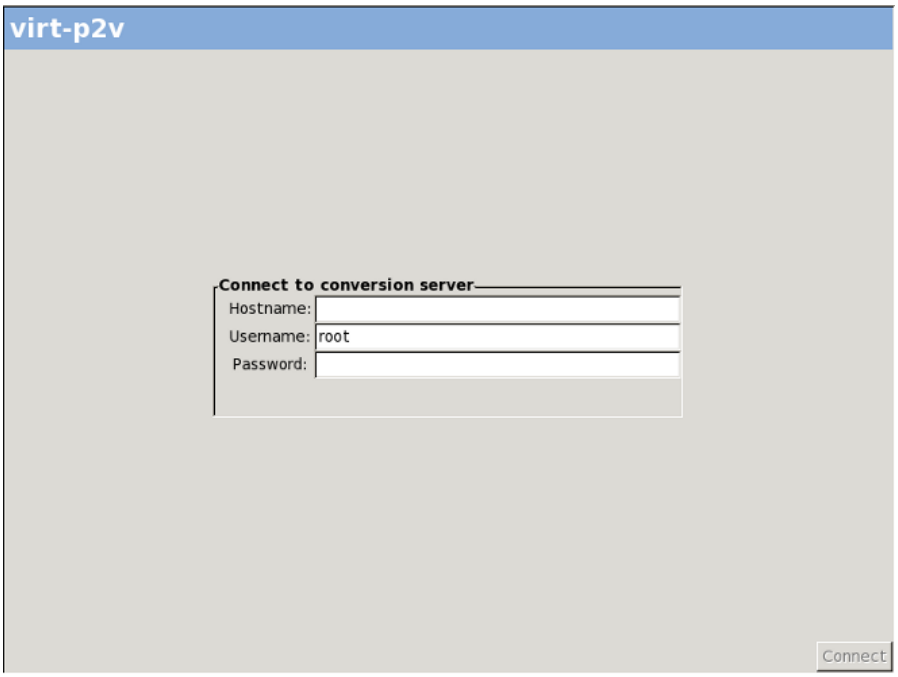

Figure 5.2. Connecting to the conversion server

4. After connecting to your conversion server, configure the virtual hardware that will be attached to the converted physical machine, and select which physical hardware should be converted. At least one Fixed Storage device must be converted, including the device containing the operating system installation.

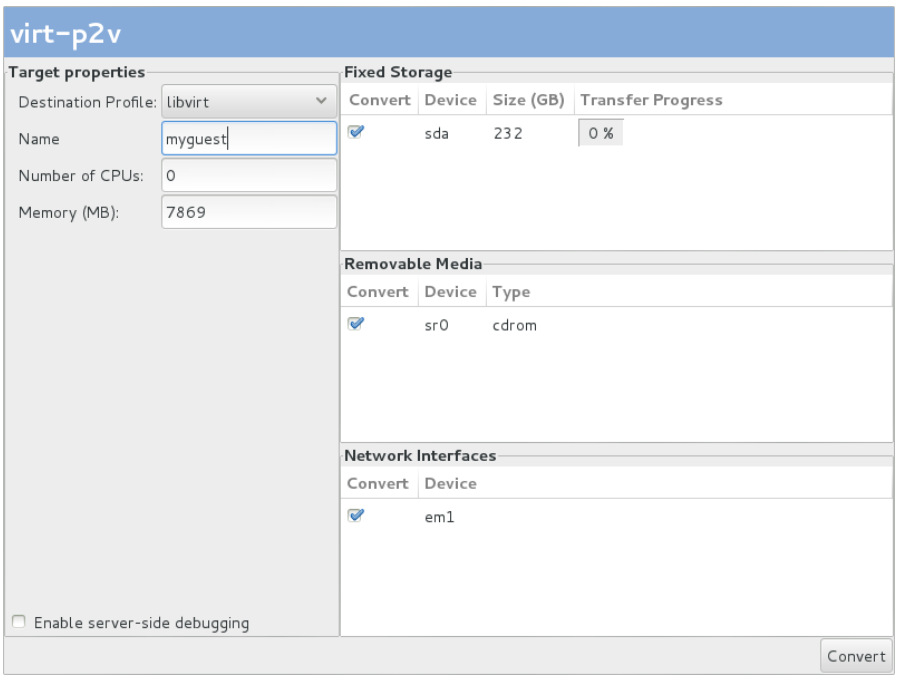

Figure 5.3. Configuring the virtual machine

5. Select a **Destination Profile** from the drop down menu. These reflect the target profiles included in the **/etc/virt-v2v.conf** file on the conversion server.
6. Enter a **Name** for the Virtual Machine that will result from the conversion.
7. The **Number of CPUs** and **Memory(MB)** are automatically detected and completed. Change them if more CPUs and/or Memory are desired on the outputted virtual machine.
8. If you wish to print debugging messages to a file, select the **Enable server-side debugging** checkbox. This instructs the server to write **LIBGUESTFS_TRACE** and **LIBGUESTFS_DEBUG** output during the conversion process. This step is *optional*.
9. When **Target Properties**, **Fixed Storage**, **Removable Media**, and **Network Interfaces** have all been configured as desired, click **Convert**.

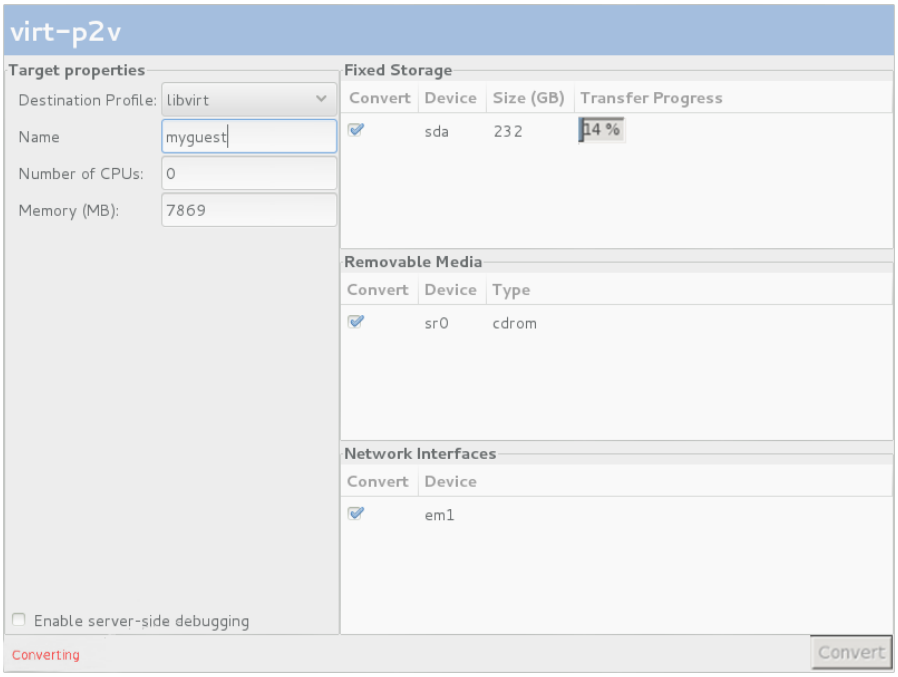

Figure 5.4. Conversion in process

10. When the conversion is complete, a success message is displayed. You can shut down the physical machine.

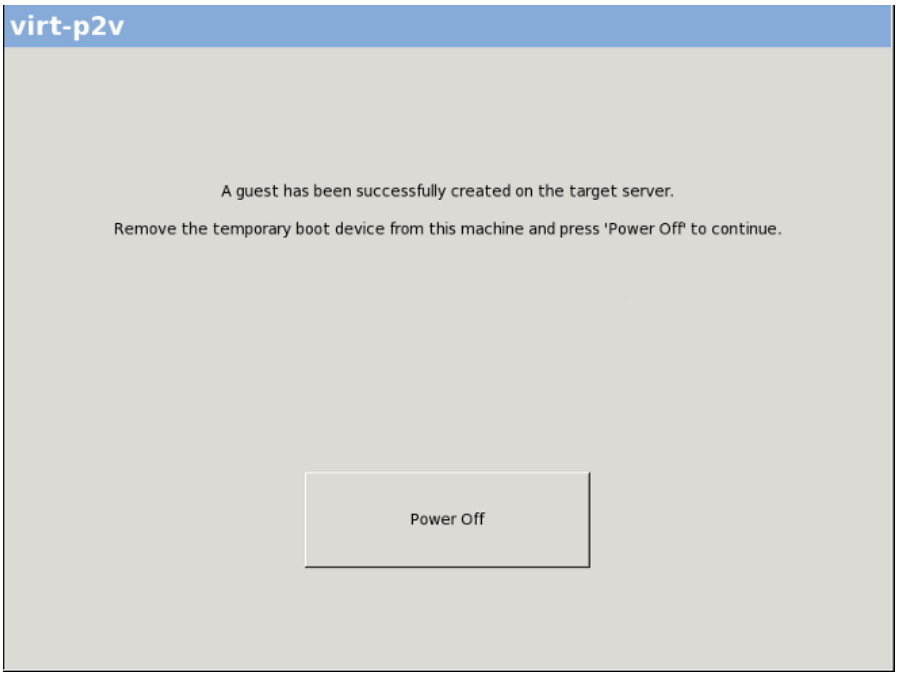

Figure 5.5. Conversion successfully completed

You have converted your physical machine into a virtual machine. You can now import and run it on a hypervisor.

5.4. Importing and running converted virtual machines on target hypervisors

Now that you have converted your physical machine to a virtual machine, it can be run on a hypervisor.

For information on running the converted physical machine as a virtual machine on Red Hat Enterprise Linux using virt-manager, see:

Section 3.4, “Running converted virtual machines”

For information on importing and running the converted physical machine as a virtual machine in Red Hat Enterprise Virtualization using the Red Hat Enterprise Virtualization Manager, see:

Section 4.3.3, “Importing and running the converted virtual machine”

Chapter 6. Debugging and troubleshooting

6.1. Debugging V2V conversions

Problems encountered when attempting a V2V conversion can be more easily explained to engineers or support services if debugging messages are enabled when V2V is run.

Exporting the debugging messages increases the verbosity of the V2V process, causing **virt-v2v** to print out messages as it runs. These messages will be displayed in the terminal from which **virt-v2v** is run.

Simple redirection can be used to print **virt-v2v** debug messages to a file. The conversion runs normally like this:

```
virt-v2v -i libvirtxml -os pool --bridge bridge_name guest_name.xml
```

Instead, edit this command to export the debug messages to the **virt-v2v.log** file. To do this, prefix the above command with the environment variables **LIBGUESTFS_TRACE=1 LIBGUESTFS_DEBUG=1**, and redirect the output to the **virt-v2v.log** file by adding **... 2>&1 | tee virt-v2v.log** to the end of the command.

```
LIBGUESTFS_TRACE=1 LIBGUESTFS_DEBUG=1 virt-v2v -i libvirtxml -os pool --bridge


 bridge_name guest_name.xml ... 2>&1 | tee virt-v2v.log
```

6.2. Debugging P2V conversions

Problems encountered when attempting a P2V conversion can be more easily explained to engineers or support services if debugging messages are enabled when P2V is run.

Exporting the debugging messages increases the verbosity of the P2V process, causing **virt-p2v** to print out messages as it runs. These messages will be displayed in the terminal from which **virt-p2v** is run.

Simple redirection can be used to print **virt-p2v** debug messages to a file.

To enable P2V debugging, select the **Enable server-side debugging** checkbox on the convert screen in the **virt-p2v** client before clicking the **Convert** button.

This instructs the server to write **LIBGUESTFS_TRACE** and **LIBGUESTFS_DEBUG** output during the conversion process.

Refer to Chapter 5, *Converting physical machines to virtual machines* for instructions on using **virt-p2v**.

6.3. Known issue with iscsi/mpath/scsi storage volumes

It is not possible at the moment with **virt-v2v** to convert a guest with a storage volume in a pool of any of the following types:

- iscsi

- mpath
- scsi

Converting such a guest results in a failed conversion. There is no workaround for this issue.

Chapter 7. References

This chapter contains reference information for **virt-v2v**.

7.1. virt-v2v Parameters

The following parameters can be used with **virt-v2v**:

-i *input*	Specifies the input method to obtain the guest for conversion. The default is libvirt. Supported options are: **libvirt** Guest argument is the name of a libvirt domain. **libvirtxml** Guest argument is the path to an XML file containing a libvirt domain. **ova** Guest argument is the path to a VMware-exported OVA file.
-ic *URI*	Specifies the connection to use with the libvirt input method. If omitted, this defaults to **qemu:///system**. Note, this only works when **virt-v2v** is run as root. **virt-v2v** can currently automatically obtain guest storage from local libvirt connections, ESX / ESX(i) connections, and connections over SSH. Other types of connection are not supported.
-o *method*	Specifies the output method. If no output method is specified, the default is libvirt. Supported output methods are: **libvirt** Create a libvirt guest. See the **-oc** and **-os** options. **-os** must be specified for the libvirt output method. **rhev** Create a guest on a Red Hat Enterprise Virtualization export storage domain, which can later be imported using the Manager. The export storage domain must be specified using **-os** for the **rhev** output method.
-oc *URI*	Specifies the libvirt connection to use to create the converted guest. If omitted, this defaults to **qemu:///system** if **virt-v2v** is run as root. Note that **virt-v2v** must be able to write directly to storage described by this libvirt connection. This makes writing to a remote connection impractical at present.
-os *storage*	Specifies the location where new storage will be created for the converted guest. This is dependent on the output method, specified by the **-o** parameter. For the **libvirt** output method, this must be the name of a storage pool. For the **rhev** output method, this specifies the NFS path to a Red Hat Enterprise Virtualization export storage domain. Note that the storage domain must have been previously initialized by the Red Hat

	Enterprise Virtualization Manager. The domain must be in the format <***host***>:<***path***>, for example, **rhev-storage.example.com:/rhev/export**. The NFS export must be mountable and writable by the host running **virt-v2v**.
-op *pool* (deprecated)	This parameter is still supported, but is deprecated in favor of ***-os***.
-osd domain (deprecated)	This parameter is still supported, but is deprecated in favor of ***-os***.
-of *format*	Specifies the on-disk format which will be used for the converted guest. Currently supported options are **raw** and **qcow2**. The output format does not need to be the same as the source format - **virt-v2v** can convert from **raw** to **qcow2** and vice versa. If not specified, the converted guest will use the same format as the source guest.
-oa *allocation*	Specifies whether the converted guest should use **sparse** or **preallocated** storage. The allocation scheme does not need to be the same as the source scheme: **virt-v2v** can convert from sparse to preallocated and vice versa. If not specified, the converted guest will use the same allocation scheme as the source.
-on *outputname*	Renames the guest. If this option is not used, then the output name is the same as the input name.
-f *file* | --config *file*	Load a **virt-v2v** configuration from file. Multiple configuration files can be specified; these will be searched in the order in which they are specified. If no configuration is specified, the defaults are **/etc/virt-v2v.conf** and **/var/lib/virt-v2v/virt-v2v.db** in that order. **Important** When overriding the default configuration details, we recommend also specifying **/var/lib/virt-v2v/virt-v2v.db**, as it contains default configuration data required for conversions.
-n *network* | --network *network*	Map all guest bridges or networks which don't have a mapping in the configuration file to the specified network. This option cannot be used in conjunction with ***--bridge***.
-b *bridge* | --bridge *bridge*	Map all guest bridges or networks which don't have a mapping in the configuration file to the specified bridge. This option cannot be used in conjunction with ***--network***.
-p *profile* | --profile *profile*	Use the default values for output method, output storage and network mappings from ***profile*** in the configuration file.
--root=*filesystem*	In a multi-boot virtual machine, select the root file system to be converted. The default value for this option is **--root=ask**. When this option is selected, **virt-v2v** lists the possible root file systems and asks the user which file system should be used.

Warning

In versions of Red Hat Enterprise Linux earlier than version 6.3, the default value was **--root=single**, which could cause **virt-v2v** to fail when a multi-boot virtual machine was detected.

Other available options include:

first
Selects the first root device if multiple devices are detected. Since this is a heuristic, the choice may not always be correct.

single
Specifies that there is only one root device available to use. **virt-v2v** will fail if more than one device is detected.

<path>
Specifies a particular root device to use, for example, **--root=/dev/sda2** would specify the second partition on the first hard drive. If the specified device does not exist or was not detected as a root device, **virt-v2v** will fail.

--list-profiles	Display a list of target profile names specified in the configuration file.
--help	Display brief help.
--version	Display version number and exit.

7.2. Configuration changes

As well as configuring libvirt appropriately, **virt-v2v** will make certain changes to a virtual machine to enable it to run on a KVM hypervisor either with or without virtIO drivers. These changes are specific to the guest operating system. The details specified here apply to supported Red Hat Enterprise Linux versions and Windows.

7.2.1. Configuration changes for Linux virtual machines

Table 7.1. virt-v2v changes to Linux virtual machines

Change	Description
Kernel	Unbootable kernels (i.e. Xen para-virtualized kernels) will be uninstalled. No new kernel will be installed if there is a remaining kernel which supports VirtIO. If no remaining kernel supports VirtIO and the configuration file specifies a new kernel it will be installed and configured as the default.
X reconfiguration	If the guest has X configured, its display driver will be updated. See Table 7.2, "Configured drivers in a Linux guest" for which driver will be used.
Rename block devices	If reconfiguration has caused block devices to change name, these changes will be reflected in **/etc/fstab**.
Configure device drivers	Whether VirtIO or non-VirtIO drivers are configured, **virt-v2v** will ensure that the correct network and block drivers are specified in the modprobe configuration.
initrd	**virt-v2v** will ensure that the initrd for the default kernel supports booting the root device, whether it is using VirtIO or not.
SELinux	**virt-v2v** will initiate a relabel of the guest on the next boot. This ensures that any changes it has made are correctly labeled according to the guest's local policy.

virt-v2v will configure the following drivers in a Linux guest:

Table 7.2. Configured drivers in a Linux guest

Para-virtualized driver type	Driver module
Display	cirrus
Storage	virtio_blk
Network	virtio_net
In addition, initrd will preload the virtio_pci driver	
Other drivers	
Display	cirrus
Block	Virtualized IDE
Network	Virtualized e1000

7.2.2. Configuration changes for Windows virtual machines

Warning

Before converting Windows virtual machines, ensure that the *libguestfs-winsupport* and *virtio-win* packages are installed on the host running **virt-v2v**. These packages provide support for NTFS and Windows para-virtualized block and network drivers. If you attempt to convert a virtual machine using NTFS without the *libguestfs-winsupport* package installed, the conversion will fail. If you attempt to convert a virtual machine running Windows without the *virtio-win* package installed, the conversion will fail giving an error message concerning missing files. See Section 4.3.1.2, "Preparing to convert a virtual machine running Windows" for details.

virt-v2v can convert virtual machines running Windows XP, Windows Vista, Windows 7, Windows Server 2003 and Windows Server 2008. The conversion process for virtual machines running Windows is slightly to different to the process for virtual machines running Linux. Windows virtual machine images are converted as follows:

1. **virt-v2v** installs VirtIO block drivers.
2. **virt-v2v** installs the CDUpgrader utility.
3. **virt-v2v** copies VirtIO block and network drivers to **%SystemRoot%\Drivers\VirtIO**. The *virtio-win* package does not include network drivers for Windows 7 and Windows XP. For those operating systems, the rtl8139 network drivers are used. rtl8139 support must be already available in the guest virtual machine.
4. **virt-v2v** adds **%SystemRoot%\Drivers\VirtIO** to **DevicePath**, meaning this directory is automatically searched for drivers when a new device is detected.
5. **virt-v2v** makes registry changes to include the VirtIO block drivers in the **CriticalDeviceDatabase** section of the registry, and ensure the CDUpgrader service is started at the next boot.

At this point, **virt-v2v** has completed the conversion. The converted virtual machine is now fully functional, and the conversion is complete for output to KVM managed by libvirt. If the virtual machine is being converted for output to Red Hat Enterprise Virtualization, the Red Hat Enterprise Virtualization Manager will perform additional steps to complete the conversion:

1. The virtual machine is imported and run on the Manager. See the *Red Hat Enterprise Virtualization Administration Guide* for details.

   

   Important

   The first boot stage can take several minutes to run, and must not be interrupted. It will run automatically without any administrator intervention other than starting the virtual machine. To ensure the process is not interrupted, no user should log in to the virtual machine until it has quiesced. You can check for this in the Manager GUI.

2. If the guest tools ISO has been uploaded to the Manager, as detailed in Section 4.3.1.2, "Preparing to convert a virtual machine running Windows", the Manager attaches the guest tools CD to the virtual machine.
3. CDUpgrader detects the guest tools ISO and installs all the VirtIO drivers from it, including additional tools that are not included in *virtio-win*. The VirtIO drivers are re-installed if the drivers in the guest tools ISO are newer than the ones previously installed from *virtio-win*. This ensures that the tools are kept up to date.

Additional procedures

A.1. Creating bootable media

The P2V Client can be booted from PXE boot, a bootable USB device, or optical media. Scripts for preparing boot options are included with the **rhel-6.x-p2v.iso** ISO in the LiveOS directory.

A.1.1. Create a P2V client boot CD

The exact series of steps that produces a CD from an image file varies greatly from computer to computer, depending on the operating system and disc burning software installed. This procedure describes burning an ISO image to disk using **Brasero** which is included in Red Hat Enterprise Linux 6.

Make sure that your disc burning software is capable of burning discs from image files. Although this is true of most disc burning software, exceptions exist.

1. Insert a blank, writable CD into your computer's CD or DVD burner.
2. Open the **Applications** menu, choose the **Sound and Video** sub-menu, and click **Brasero Disk Burner**.
3. Click the **Burn Image** button.
4. Click the **Click here to select a disc image** button.
5. Browse to the **rhel-6.x-p2v.iso** and select it for burning.
6. Click **Burn**.

Your BIOS may need to be changed to allow booting from your DVD/CD-ROM drive.

A.1.2. Create a bootable P2V USB media

1. As root, mount the **rhel-6.x-p2v.iso**:

```
mkdir /mnt/p2vmount
```

```
mount -o loop rhel-6.x-p2v.iso /mnt/p2vmount
```

2. Attach your USB device to the computer. For the **livecd-iso-to-disk** script to function, the USB filesystem *must* be formatted vfat, ext[234] or btrfs.
3. From a terminal as root run the **livecd-iso-to-disk** script:

```
bash /mnt/p2vmount/LiveOS/livecd-iso-to-disk /PATH/TO/rhel-6.x-p2v.iso
/dev/YOURUSBDEVICE
```

4. When the script finishes successfully, eject the USB device.

A.1.3. Create a PXE boot image

1. As root, mount the **rhel-6.x-p2v.iso**

```
mkdir /mnt/p2vmount
```

```
mount -o loop rhel-6.x-p2v.iso /mnt/p2vmount
```

2. From a terminal as root run the **livecd-iso-to-pxeboot** script:

```
bash /mnt/p2vboot/LiveOS/livecd-iso-to-pxeboot /PATH/TO/rhel-6.x-p2v.iso
```

When the command successfully completes, there is a tftpboot directory in the directory from which the command was run.

3. Rename the newly created tftpboot directory to a more descriptive name:

```
mv tftpboot/ p2vboot/
```

4. Copy the p2vboot/ sub-directory to the /tftpboot directory:

```
cp -R p2vboot/ /tftpboot/
```

Set up your DHCP, TFTP and PXE server to serve **/tftpboot/p2vboot/pxeboot.0** .

Note

The initrd image contains the whole CD ISO. You will notice when pxebooting that initrd can take a long time to download. This is normal behavior.

Revision History

Revision 11-60.404 **Mon Nov 25 2013** **Rüdiger Landmann**

Rebuild with Publican 4.0.0

Revision 11-60 **Mon Nov 18 2013** **Dayle Parker**

Version for 6.5 GA release.

Revision 11-59 **Fri Nov 15 2013** **Dayle Parker**

Updated screenshots in "Attaching an export storage domain" and "Scripting the v2v process" procedures.

Updated Red Hat Customer Portal procedures ("Preparing to convert a virtual machine running Linux") in two chapters.

Added reference to Red Hat Enterprise Virtualization Administration Guide.

Revision 11-57 **Thurs Nov 7 2013** **Dayle Parker**

Minor edits and markup for BZ#994812.

Revision 11-56 **Fri Sept 27 2013** **Dayle Parker**

Minor edits for beta release.

Included new P2V screenshots with debugging checkbox.

Revision 11-54 **Thurs Sept 26 2013** **Dayle Parker**

Edited wording issues found in localization.

Reorganized and edited Windows content from Chapter 3 covered in other sections.

Revision 11-53 **Mon Sept 16 2013** **Dayle Parker**

Edited list of supported operating systems and hypervisors based on QE feedback for BZ#972987.

Revision 11-51 **Fri Sept 13 2013** **Dayle Parker**

Edited converting VMDK virtual machines section for BZ#982844.

Edited list of supported operating systems and hypervisors for BZ#972987.

Revision 11-50 **Mon Sept 9 2013** **Dayle Parker**

Added section on debugging P2V for BZ#986133.

Added conversion limitation note for BZ#964092.

Revision 11-49 **Thurs Sept 5 2013** **Dayle Parker**

Added VMDK conversion example for BZ#982844.

Added "ova" to input options in the virt-v2v Parameters table.

Revision 11-48 **Fri Aug 16 2013** **Dayle Parker**

Added conversion time example in chapters 2 and 3 for BZ#975320.

Revision 11-47 **Fri Aug 9 2013** **Dayle Parker**

Combined duplicate content from chapters 3 and 5 to create a draft chapter.

Revision 11-46 **Fri Apr 26 2013** **Dayle Parker**

Added revised documentation suite list to first chapter; restructured Preface and first chapter.

Revision 11-45 **Mon March 4 2013** **Dayle Parker**
Took topics out of topic tool and placed text into book.

Revision 11-44 **Tue Feb 19 2013** **Dayle Parker**
Version for 6.4 GA release.

Revision 11-42 **Tues Feb 5 2013** **Dayle Parker**
Minor corrections throughout book.

Revision 11-41 **Thurs Jan 31 2013** **Dayle Parker**
Updated procedures relating to the guest tools ISO in Chapter 3 and 4 for BZ#881469.
Updated "Importing and running the converted virtual machine" in Chapter 3 for BZ#881469.

Revision 11-41 **Wed Jan 30 2013** **Dayle Parker**
Updated procedures relating to the guest tools ISO in Chapter 3 and 4 for BZ#881469.
Updated "Importing and running the converted virtual machine" in Chapter 3 for BZ#881469.

Revision 11-39 **Thurs Jan 24 2013** **Dayle Parker**
Changed subsection title 2.1. Converting a virtual machine to 2.1. Preparing to convert a virtual machine; minor reorganization of section.
Minor grammatical edits throughout book.

Revision 11-38 **Wed Jan 16 2013** **Dayle Parker**
Made corrections for QE review in BZ#881469.

Revision 11-37 **Thurs Dec 20 2012** **Dayle Parker**
Made corrections for QE review in BZ#881469.

Revision 11-36 **Thurs Dec 13 2012** **Dayle Parker**
Corrected ISO filenames for BZ#713997.

Revision 11-35 **Wed Nov 28 2012** **Dayle Parker**
Applied QE feedback to Chapter 3 - new RHEV screenshots and typo fixes.

Revision 11-34 **Tue Nov 27 2012** **Dayle Parker**
Applied QE feedback to 'Scripting the v2v Process' section for BZ#754884.

Revision 11-33 **Thu Nov 22 2012** **Laura Novich**
fixed script in the Single host v2v script section for https://bugzilla.redhat.com/show_bug.cgi?id=832361

Revision 11-32 **Thurs Nov 22 2012** **Dayle Parker**
Applied SME feedback to 'Scripting the v2v Process' section for BZ#754884.

Revision 11-31 **Wed Nov 21 2012** **Dayle Parker**
Edited and republished 'Scripting the v2v Process' section for BZ#754884.

Revision 11-30 **Fri Nov 16 2012** **Dayle Parker**
Clarified Guest Tools ISO paragraph for BZ#754887.

Revision 11-29 **Mon Nov 12 2012** **Dayle Parker**
Added note about REST API in 3.3.5. Scripted bulk v2v process for BZ#832361.

Revision 11-28 **Thurs Nov 8 2012** **Dayle Parker**
Corrected typos for BZ#713997.

Revision 11-27 **Wed Nov 7 2012** **Dayle Parker**
Edited minor typos in topics.

Revision 11-26 **Wed Nov 7 2012** **Dayle Parker**
Updated script in 3.3.5. Scripted bulk v2v process for BZ#832361.

Revision 11-24 **Mon Nov 5 2012** **Dayle Parker**
Updated ISO filename and link for BZ#713997 in 5.2.4. Create the bootable media and A.1. Creating bootable media.

Revision 11-23 **Fri Nov 2 2012** **Dayle Parker**
Corrections for BZ#832019 in Chapter 4 and 5.

Revision 11-22 **Thu Nov 1 2012** **Laura Novich**
added section to troubleshooting BZ841542

Revision 11-21 **Thurs Nov 1 2012** **Dayle Parker**
Corrections for BZ#754888 in Chapter 6 and 7.

Revision 11-20 **Thurs Nov 1 2012** **Dayle Parker**
Additions for BZ#754886 in Chapter 2.
Corrections for BZ#754887 in Chapter 3.

Revision 11-19 **Wed Oct 31 2012** **Laura Novich**
added .iso download information

Revision 11-18 **Wed Oct 31 2012** **Dayle Parker**
Fixes for BZ#754886 in Chapter 2.

Revision 11-18 **Wed Oct 31 2012** **Laura Novich**
added .iso download information

Revision 11-17 **Wed Oct 31 2012** **Laura Novich**
changed revision history page and chapter 1 BZ754885

Revision 11-16 **Tue Oct 30 2012** **Laura Novich**
fixed typo in chapter 3

Revision 11-15 **Mon Oct 29 2012** **Laura Novich**
fixed typo in chapter 3

Revision 11-14 **Mon Oct 29 2012** **Dayle Parker**
Made virtualization terms consistent throughout book for BZ#825894

Revision 11-13 **Thu Oct 25 2012** **Laura Novich**
finished v2v migration chapter - added ESX(i) as supported

Revision 11-12 **Wed Oct 24 2012** **Laura Novich**
finished v2v migration chapter

Revision 11-11 **Wed Oct 24 2012** **Laura Novich**
fixed P2V chapter and reference libvirt chapter - https://bugzilla.redhat.com/show_bug.cgi?id=816930

Revision 11-10 **Wed Oct 24 2012** **Laura Novich**
fixed P2V chapter - https://bugzilla.redhat.com/show_bug.cgi?id=816930

Revision 11-9 **Mon Oct 15 2012** **Laura Novich**
fixed BZ754885 - changed Revision History and Preface.

Revision 11-8 **Sun Oct 14 2012** **Laura Novich**
fixed ch5

Revision 11-7 **Sun Oct 14 2012** **Laura Novich**
fixed ch5

Revision 11-6 **Monday June 18 2012** **Laura Novich**
Preparing for GA

Revision 11-5 **Monday June 04 2012** **Laura Bailey**
Corrected note regarding storage pool requirement (BZ#709279).

Revision 11-4 **Monday May 14 2012** **Laura Bailey**
Added details about Virt P2V requirements.

Revision 11-3 **Monday March 26 2012** **Laura Bailey**
Corrected a number of minor typographical errors and improved clarity (BZ#713997).

Revision 11-2 **Monday March 5 2012** **Dayle Parker**
Made admonition title style consistent throughout book.(BZ#732866).
Labelled figures, edited screenshot images in Section 5.3. (BZ#732865).

Revision 11-1 **Friday February 24 2012** **Laura Bailey**
Included the V2V migration chapter (BZ#709279).

Revision 10-0 **Monday January 23 2012** **Laura Bailey**
Corrected word usage according to style guide (BZ#733005).
Updated use of the ***--file*** parameter (BZ#749766).
Corrected channel command usage (BZ#765869).

Revision 9-0 **Thursday December 08 2011** **Laura Bailey**
Reinstated P2V-related content previously included by Tim Hildred.
Adjusted formatting to make replaceable content more readily identifiable.
Updated parameters available for use with virt-v2v.

Added admonition about attempting to convert the Windows Recovery Console.

Revision 7-0 **Friday December 02 2011** **Laura Bailey**

Release for GA of Red Hat Enterprise Linux 6.2.

Revision 6-0 **Friday July 22 2011** **Tim Hildred**

Promoted and emphasised section detailing acceptable storage format/allocation policy combinations
Changed warnings to importants

Revision 5-0 **Friday June 17 2011** **Tim Hildred**

Fixed the following bugs:
BZ#712320 - Added warnings about transferring multiple HDD images over ssh.
BZ#696050 - Cannot import qcow2/preallocated guest to NFS data domain when convert a qcow2/sparse guest with -oa preallocated parameter. .
BZ#710161 - Section detailing the installation of the pre-reqs for virt-v2v is not detailed enough.

Revision 4-0 **Monday April 11 2011** **Cheryn Tan**

Fixed the following bugs:
BZ#694773 - It is not required for upload Guest Tools ISO for guest to install driver
BZ#694778 - It's better to add converting a remote KVM virtual machine to RHEV to guide
BZ#694775 - The virt-v2v convert command syntax "virt-v2v -os pool --network netname vm-name" need update

Revision 3-0 **Friday April 8 2011** **Cheryn Tan**

Fixed the following bugs:
BZ#694437 - The time stamp of copyright should be updated
BZ#694441 - Can't obtain virt-v2v relevant software with the V2V Guide steps
BZ#694442 - The doc should update about portmap service, as portmap is replaced by rpcbind in RHEL6

Revision 2-0 **Monday November 29 2010** **David Jorm**

Complete draft except for example 3.3

Revision 1-0 **Monday October 25 2010** **David Jorm**

Initial build