

simplevm-v2v

Installation, Testing, and Usage Guide

Version 0.2.2 · Technology Preview

© Copyright 2026 Thinware, Inc. All Rights Reserved.

1. Overview

simplevm-v2v is a command-line utility that migrates a VMware virtual machine into a ready-to-run guest on a SimpleVM host. You point it at the VM's folder (the one holding its .vmx and .vmdk files); it converts each disk to the native qcow2 format, applies the deterministic fixes a converted guest needs to boot under KVM, and registers the machine with libvirt **defined but powered off** so you can review it before first boot. (Hyper-V and OVF/OVA sources are planned for a future release.)

simplevm-v2v is a deliberately simpler, more opinionated alternative to Red Hat's virt-v2v, built on the SimpleVM principle of simplicity over flexibility. Where virt-v2v modifies the guest to run optimally on virtio — injecting drivers, editing the registry, and rebuilding the initramfs — simplevm-v2v takes the opposite approach: it never touches the guest, and instead rebuilds a minimal, maximally compatible hardware set so the VM boots with no in-guest changes at all. virtio is to a SimpleVM guest what VMware Tools is to a VMware guest: not required to boot, but worth adding afterward for performance.

That philosophy produces the features that set it apart from virt-v2v:

- **Guaranteed boot, or a clean refusal** — its first priority is that every converted VM boots. When it cannot guarantee that, it refuses (or rolls back) rather than hand back a VM that will not start.
- **No guest surgery** — the source VM is never modified, every change lands in the output qcow2 images, and the guest's operating system is left exactly as it was.
- **Compatible hardware by default** — storage and network are forced to the most compatible bus the guest supports: SATA and e1000 for modern guests, IDE for pre-Windows-8 Windows, and virtio only when you ask for it with `--virtio`.
- **Deterministic boot fixes, applied and reported** — the two fixes a clean conversion needs — the Windows/Linux hardware-clock offset and the UEFI fallback bootloader — are applied automatically and spelled out in the summary, not left as homework.

- **Honest about what it drops** — devices it does not recreate that could affect boot or function (a virtual TPM with BitLocker, PCI passthrough, serial ports) are detected and called out before and after the run.
- **Atomic and predictable** — a normal failure rolls back the partial conversion; there is one opinionated path with no hardware knobs, and every run ends with a clear report of what changed and the steps to get the best performance.

Under the hood, `simplevm-v2v` drives standard host tools — `qemu-img` for the disk conversion, `virt-install` and `virsh` to define the guest, and `libguestfs` (`guestfish`) for the offline UEFI fix — rather than `virt-v2v`. Each run also produces a detailed log and a one-line summary suitable for monitoring.

This guide describes how to install the utility on a SimpleVM server, validate it, and run conversions.

Technology Preview Version 0.2.2 is a Technology Preview release provided for evaluation. It is feature-complete for VMware sources on Rocky Linux 10.x hosts but has not yet completed full production qualification. Before relying on it, confirm its behavior in your environment with a dry run (Section 4.1) followed by a conversion of a non-critical virtual machine. `simplevm-v2v` never modifies the source VM; every change it makes lands in the converted copy.

2. Prerequisites

Before you begin, ensure the following:

- **Location** — These steps are performed on the SimpleVM host itself (the hypervisor), not inside a guest virtual machine.
- **Privileges** — Administrative (root) access to the SimpleVM server. The utility writes virtual-machine storage, manages `libvirt`, and runs `libguestfs`, so all conversion operations must run as root (via `sudo`).
- **Required package: `libguestfs-tools`** — `simplevm-v2v` uses the `guestfish` program from the `libguestfs-tools` package to apply the offline UEFI boot fix, so this package must be installed alongside the utility (see Section 3, Step 1):

```
sudo dnf install libguestfs-tools -y
```

- **Platform tools** — The `virsh`, `qemu-img`, and `virt-install` commands — the libvirt/KVM tools `simplevm-v2v` drives. These are part of the SimpleVM virtualization stack on the host.
- **Source VM** — The VMware VM's folder, containing its `.vmx` and `.vmdk` files, reachable from the host as a local path or mount (for example, a backup, full copy of the VM located on a NAS or copied to local storage on the SimpleVM host). The VM must be consolidated: `simplevm-v2v` converts only standalone disks, so delete or consolidate any snapshots (and resolve linked clones) in vSphere first.

Note: VM backups created by Thinware vBackup are a compatible source.

- **Host networking** — Converted guests attach to libvirt's 'default' network, which must be defined and active on the host.
- **Destination storage** — A directory with enough free space for the converted qcow2 image(s). By default `simplevm-v2v` writes to the libvirt 'default' storage pool; use `--output-dir` to choose another directory.

NOTE To list the virtual machines already defined on the host and confirm that a name is not already in use, run:

```
sudo virsh list --all
```

TIP Conversions of large disks can run for a long time, and a dropped session will end any command running in it. Run conversions inside a `tmux` session so they survive a disconnect or browser refresh — see Section 7.

3. Installing `simplevm-v2v`

Procedure

1. Install the prerequisite package.

1. In the SimpleVM web console, open **Terminal**.
2. Install `libguestfs-tools` and confirm that `guestfish` is available:

```
sudo dnf install libguestfs-tools -y
```

```
guestfish --version
```

2. Create the application directory.

1. Log in to the SimpleVM web console and open **File browser**.

2. Browse to /opt.
3. Click the actions menu (... , top right) and select **Create directory**:
 - **Directory name:** simplevm
 - **Directory owner:** root:root
4. Click **Create**.

NOTE If you already installed the sister utility simplevm-backup, the /opt/simplevm directory already exists; place simplevm-v2v alongside it and continue with the next step.

3. Upload the executable.

1. In **File browser**, browse to /opt/simplevm.
2. Click **Upload**, select the simplevm-v2v file, and click **Open**.

4. Make the application executable.

1. In the SimpleVM web console, open **Terminal**.
2. Run the following commands:

```
cd /opt/simplevm  
sudo chmod +x simplevm-v2v
```

Verification

Confirm that the utility runs and displays its help text:

```
./simplevm-v2v --help
```

The command prints a usage summary, including all available options and exit codes. (Displaying help does not require administrative privileges; conversion operations do — see Section 4.)

TIP To run the utility from any directory without typing its full path, create a symbolic link on the system path:

```
sudo ln -s /opt/simplevm/simplevm-v2v /usr/local/bin/simplevm-v2v
```

4. Running a conversion

The utility takes one required option and several optional flags:

```
sudo ./simplevm-v2v -p <vm-folder> [options]
```

Short	Long	Description
-p	--path	Path to the source VMware VM folder (the folder containing the .vmx and .vmdk files). (Required)
-o	--output-dir	Directory for the converted qcow2 image(s). Default: the libvirt 'default' storage pool path.
-n	--vm-name	Name for the new guest. Default: the VM's name from its VMware configuration. Must be a valid libvirt domain name.
	--virtio	Use virtio storage and network instead of the compatible default. Requires the guest to already have virtio drivers (virtio-win on Windows, virtio modules in the initramfs on Linux).
-y	--yes	Skip the confirmation prompt (for automation).
-d	--dry-run	Validate and report the planned actions without making any changes.
-v	--verbose	Produce detailed diagnostic output.
-h	--help	Display the help text and exit.

IMPORTANT Run all conversion operations as root (using sudo), including dry runs. The utility writes storage, manages libvirt, and runs libguestfs; without administrative privileges it cannot reach these and will stop with an error.

The examples below are run from /opt/simplevm; alternatively, use the full path /opt/simplevm/simplevm-v2v (or the symbolic link from Section 3).

4.1. Validate with a dry run (recommended first)

A dry run performs all validation and reports exactly what **would** happen — the detected source, the guest's CPUs, memory, and firmware, the storage and network hardware that would be used, the disks that would be converted and their qcow2 targets, and (for UEFI guests) that the boot fix would run — without modifying anything. A dry run also collects every blocking problem it finds in a single pass, so you can resolve them all before a real run. Always start here, particularly on a production host.

```
sudo ./simplevm-v2v -p /mnt/vmware/web01 --dry-run
```

4.2. Perform a conversion

After reviewing the dry run, run the conversion:

```
sudo ./simplevm-v2v -p /mnt/vmware/web01
```

simplevm-v2v summarizes what it is about to do and prompts for confirmation (type yes to proceed). It then converts each disk to qcow2, applies the boot fixes the guest needs, and defines the guest with libvirt. The converted VM is left powered off so you can review it first; start it with `sudo virsh start <vm-name>`. To set the name and output location explicitly:

```
sudo ./simplevm-v2v -p /mnt/vmware/web01 -n web01-migrated -o /var/lib/libvirt/images
```

4.3. Opt into virtio, or skip the prompt (optional)

By default the converted guest uses compatible hardware (SATA and e1000, or IDE for pre-Windows-8 Windows) so it boots with no in-guest changes. If the guest is already prepared for virtio, add `--virtio` to use virtio storage and network for better I/O. For unattended use, add `--yes` to skip the confirmation prompt:

```
sudo ./simplevm-v2v -p /mnt/vmware/db01 --virtio --yes
```

NOTE `--virtio` requires the guest to already have virtio drivers — `virtio-win` on Windows, or virtio modules in the `initramfs` on Linux. Without them the converted VM will not boot. When in doubt, omit `--virtio` for a guaranteed boot, then migrate to virtio after first boot following the steps in the conversion summary.

Conversion output

Each successful run defines a ready-to-run guest in libvirt (powered off) and writes the converted disk image(s) to the output directory:

- `<output-dir>/<vm-name>.qcow2` — for a single-disk VM, or
- `<output-dir>/<vm-name>-sda.qcow2`, `<vm-name>-sdb.qcow2`, ... — one per disk for a multi-disk VM (`vda/vdb...` with `--virtio`; `hda/hdb...` for IDE-era guests)

It also writes two logs:

- `/var/log/simplevm/v2v/<vm-name>-<timestamp>.log` — a detailed transcript of the run
- `/var/log/simplevm/simplevm-v2v.log` — a one-line summary per run

At the end of a run, `simplevm-v2v` prints a summary of every change it made for boot compatibility (hardware, clock, firmware, and the UEFI fix) and the recommended next

steps — reviewing and starting the VM, reassigning networking after first boot, and, where applicable, migrating to virtio for better performance. After a successful run, confirm the guest with `sudo virsh list --all` and start it with `sudo virsh start <vm-name>`.

5. Verifying results

Exit codes

`simplevm-v2v` returns a standard exit code, suitable for scripting and scheduling:

Code Meaning

- | | |
|---|---|
| 0 | Success, or a completed dry run. |
| 1 | Validation or conversion failure. No guest was defined, and any partial output was rolled back. |
| 2 | Could not guarantee that the converted VM would boot. <code>simplevm-v2v</code> refused, or rolled back, rather than hand back a VM that may not start (see Section 6). |

Logs

- **Per-conversion log** — `/var/log/simplevm/v2v/<vm-name>-<timestamp>.log` records that run in detail.
 - **Application log** — `/var/log/simplevm/simplevm-v2v.log` records one summary line per run (result, source type, firmware, hardware profile, and duration), providing an at-a-glance history suitable for monitoring.
-

6. Troubleshooting

"Required command not found in PATH." A platform tool is missing. `simplevm-v2v` needs `virsh`, `qemu-img`, and `virt-install`; UEFI guests also need `guestfish`. `libguestfs-tools` provides `guestfish`:

```
sudo dnf install libguestfs-tools -y
```

"libvirt's 'default' network is not defined" or "not active." Converted guests attach to the default network. Define and start it, then retry:

```
sudo virsh net-start default && sudo virsh net-autostart default
```

"A VM named ... already exists" or "Target image already exists." Choose a different name with `--vm-name`, or a different directory with `--output-dir` (or remove the conflicting VM or image), then re-run.

"This VM is running on a snapshot (or is a linked clone)." `simplevm-v2v` converts only consolidated disks. In vSphere, delete or consolidate the VM's snapshots, then re-run.

"Source NIC MAC ... is already in use on this host." `simplevm-v2v` preserves the source NIC's MAC address. Remove or change the MAC on the conflicting VM, then re-run.

Could not guarantee boot (exit 2). For a UEFI guest, no EFI System Partition was found, or its bootloader was missing or ambiguous, so the conversion was refused rather than produce a VM that may not start. Inspect the guest's ESP and convert manually, or contact support.

The converted VM has no network after first boot. This is expected: the new NIC is a different model than VMware's, so the guest sees a new interface. Reassign networking once after first boot (the conversion summary notes this).

The utility reports that it has expired. This Technology Preview build is time-limited. Contact Thinware for an updated release.

7. Running long conversions with `tmux` (recommended)

Conversions of large disks can take a long time, and a dropped SSH or web-console session ends any command running in that session. We recommend running conversions inside a `tmux` session, which keeps the conversion running even if you disconnect, refresh the browser, or close the terminal. You can detach from a running conversion and reattach later to check its progress.

Installing `tmux`

Install `tmux`, then enable mouse support (so you can scroll with the mouse wheel) for all users. In the SimpleVM web console, open **Terminal** and run:

```
sudo dnf install tmux -y
```

```
sudo tee /etc/tmux.conf > /dev/null << 'EOF'
```

```
# Enable mouse support for all users
```

```
set -g mouse on
```

```
EOF
```

Using tmux (persistent sessions)

1. Create and attach to a new named session:

```
tmux new -s <session-name>
```

2. Run your long-running conversion inside the session (for example, the simplevm-v2v command from Section 4).

3. Detach from the session to leave it running in the background: press **Ctrl + b**, then **d**.

4. Reattach later (after a disconnect, browser refresh, or closed terminal):

```
tmux attach -t <session-name>
```

TIP For example, to run a conversion in a session named **convert**, start the session, launch the conversion, then press **Ctrl + b, d** to detach and reattach later with `tmux attach -t convert`:

```
tmux new -s convert
```

```
sudo /opt/simplevm/simplevm-v2v -p /mnt/vmware/web01
```

8. Support and feedback

This is a Technology Preview release, and your testing helps shape the final product. Please report issues, unexpected behavior, or suggestions to Thinware support at **support@thinware.net**.