



Windows VMs oVirt 4.4

What's new?

Gal Zaidman
Software Engineer

09/2020



About me

I'm a Software Engineer at Red Hat working in the integration team of the oVirt project. I've started working at Red Hat 2.5 years ago as an intern and since then I have worked on:

- Python 2 to 3 porting effort.
- Virtio-win drivers installer.
- Cockpit ovirt installation flow.
- OCP on oVirt.



[Blog](#)

[Linkedin](#)

[Github](#)

Why do we need an installer ?



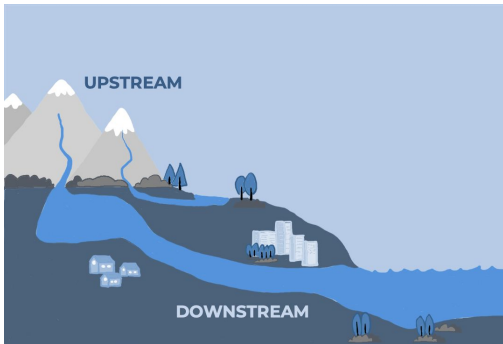
So what has changed?

Upstream first!

Same source code for upstream and downstream projects

And all layered products

[Source code github link](#)



So what has changed?

From an oVirt project to a RHEL/Fedora project

Installer has been contributed to the fedora virtio-win project and maintained by oVirt and virtio-win engine. That means that you will always get the latest drivers when they come out.



So what has changed?

Works on all microsoft supported windows versions
Added support for windows 10, 2k16 and 2k19

Officially included:

- Windows 8
- Windows 8.1
- Windows 10
- Windows 2K12
- Windows 2K12R2
- Windows 2K16
- Windows 2K19

So what has changed?

Docs changed



Installation Demo

This demo will show the process of creating a windows 10 VM, installing and uninstalling the drivers and agents.

We assume that the virtio-win iso and the windows iso is already available on the engine.

You can download the latest virtio-win iso [here](#) and use [this](#) ansible playbook to upload the iso

Installing in silent mode with the CLI

We can also perform the installation or uninstallation in silent mode with the CLI this mostly useful for **automation of the installation.**

1. Open cmd with **as admin.**
 2. Go to the virtio-win cd.
 3. installation: `virtio-win-guest-tools.exe /s /qn`
uninstallation: `virtio-win-guest-tools.exe /uninstall /s /qn`
- We can add the `/l*v "<path to log>"` to specify a path to write the log files.
 - We can add the `/norestart` to prevent a reboot after the installation.

Installing in silent mode with the CLI

We can override the default behavior and manually select which features would be installed with the ADDLOCAL property, for example installing just vioinput and vioscsi:

```
msiexec /i "<path to msi>" /qn /!*v "<path to log>" ADDLOCAL=FE_vioinput_driver,FE_vioscsi_driver.
```

List of the features we can override:

- FE_balloon_driver - Balloon Virtio-balloon is used to control the amount of memory a virtual machine actually accesses. It offers improved memory overcommitment.
- FE_network_driver - Network Paravirtualized network driver provides enhanced performance over emulated devices.
- FE_pvpanic_driver - Pvpanic QEMU pvpanic device driver.
- FE_qemufwcfg_driver - Qemufwcfg QEMU FWCfg device driver.
- FE_qemupcserial_driver - Qemupcserial QEMU PCI serial device driver.
- FE_vioinput_driver - Vioinput VirtIO Input Driver.
- FE_viorng_driver - Viorng VirtIO RNG device driver.

Installing in silent mode with the CLI

- FE_vioscsi_driver - Vioscsi VirtIO SCSI pass-through controller.
- FE_vioserial_driver - Vioserial VirtIO Serial device driver.
- FE_viostor_driver - Viostor Virtio Block driver.

Since the installer is an MSI we can use all MSI features (that are given to us by MS), we can refer to Microsoft docs:

[General](#)

[Command-Line Options](#)

[Property Reference](#)

The installation log

The best way to understand what when wrong on the installation is to view the log file.

The log file is saved by default in %TEMP% (path to a tmp file folder), with the name
Virtio-win-guest-tools_{Date}_{SEQ}_{Projects}.

A regular installation will generate 6 log files, 1 general and 5 which are project specific for example:

- Virtio-win-guest-tools_20200406072501 - General log file
- Virtio-win-guest-tools_20200406072501_000_virtio_win_gt_x64.msi
- Virtio-win-guest-tools_20200406072501_001_QxlWddmDod_0.19.2.0_x64.msi
- Virtio-win-guest-tools_20200406072501_002_spice_vdagent_x64_0.10.0_2.msi
- Virtio-win-guest-tools_20200406072501_003_ovirt_guest_agent_x64.msi
- Virtio-win-guest-tools_20200406072501_004_qemu_ga_x86_64.msi

The installation log

The General log file (Virtio-win-guest-tools XXXXX)

A short log file which summarizes the installation, it is written as the installation progress and contains information about the phases of the installation, bundle related variables which are evaluated very early in the installation process.

It should probably be the first log we will check to see where we failed and get a general understanding of the problem from the error message.

Tips, what should we look for:

- Error string of course :).
- Search for the string "Planned package: ", it should tell you what is the state and action set for each component.

The installation log

The Product specific log file (Virtio-win-guest-tools {Date} {SEQ} {Projects})

A verbose log file which list all the variables that were available during the installation and what happened on each phase.

In general once we know on which component the installation failed we should look at its log file to see what is the error.

There are many phases for the installation each phase begins with "Action start TIME PHASE" and ends with "Action ended TIME PHASE Return value ".

The phases which we will probably help us the most are:

- InstallValidate - will list for each component:
 - Installed: if the component is found on the system.
 - Request: the action we take on the component, for example NULL means nothing and Local mean install it.
- InstallServices/StopServices/StartServices - see failures when installing services, for example when there is a dependency which failed to be found or started.
- InstallDriverPackages - Drivers will not be installed if there is no matching devices for them, you can look for the string "No devices found that match driver", note that is not necessarily a bad thing, but can cause problems in case there is a dependency.

oVirt

Thank you!

<https://ovirt.org/>

users@ovirt.org

 @ovirt