www.qubes-os.org /doc/templates/windows/qubes-windows-tools-4-1/

Qubes Windows Tools (QWT)

27-35 minutes ⋮ 10/17/2024

Qubes Windows Tools (QWT) are a set of programs and drivers that provide integration of Windows 7, 8.1, 10 and 11 Standalone, TemplateVMs and AppVMs with the rest of the Qubes system. They contain several components than can be enabled or disabled during installation, and rely on specific functions of Qubes which support this integration:

- **Shared components (required)** - common libraries used by QWT components
- **Qubes GUI Agent** - video driver and GUI agent that enable the seamless GUI mode that integrates windows apps onto the common Qubes trusted desktop (currently only for Windows 7)
- **Disable UAC** - User Account Control may interfere with QWT and doesn’t really provide any additional benefits in Qubes environment
- **Clipboard sender/receiver** - Support for secure clipboard copy/paste between the Windows VM and other AppVMs
- **File sender/receiver** - Support for secure file exchange between the Windows VM and other AppVMs
- **Xen PV drivers** - drivers for the virtual hardware exposed by Xen for Windows that increase performance compared to QEMU emulated devices and are required for attaching USB devices
    - Base Xen PV Drivers (required): paravirtual bus and interface drivers
    - Xen PV Disk Drivers: paravirtual storage drivers
    - Xen PV Network Drivers: paravirtual network drivers
    - Move user profiles: user profile directory (`C:\users`) is moved to VM’s private disk backed by `private.img file` in `dom0` (useful mainly for HVM templates).
- **Qubes Core Agent** (part of Qubes) - Needed for proper integration with Qubes as well as for `qvm-run` and generic `qrexec` for the Windows VM (e.g. ability to run custom service within/from the Windows VM)
- **Copy/Edit in Disposable VM** (part of Qubes) - Support for editing files in DisposableVMs
- **Audio** - Audio support is available even without QWT installation if `qvm-features audio-model` is set as `ich6`

Note: Due to the security problems described in QSB-091, installation of Qubes Windows Tools is currently blocked. Instead, a text file containing a warning is displayed. Currently, it is difficult to estimate the severity of the risks posed by the sources of the Xen drivers used in QWT possibly being compromised, so it was decided not to offer direct QWT installation until this problem could be treated properly. While Windows qubes are, in Qubes, generally not regarded as being very trustworthy, a possible compromise of the Xen drivers used in Qubes Windows Tools might create a risk for Xen or `dom0` and thus be dangerous for Qubes itself. This risk may be small or even non-existent, as stated in QSB-091. If you **understand** this risk and are **willing to take it**, you can still install the previous versions of Qubes Windows Tools, using the command

```
sudo qubes-dom0-update qubes-windows-tools-4.1.68
```

for Qubes R4.1.2, or

```
sudo qubes-dom0-update qubes-windows-tools-4.1.69
```

for Qubes R4.2.0, respectively, instead of the command listed in step 1 of the installation described below. This will provide the .iso file to be presented as an installation drive to the Windows qube in step 3 of the QWT installation.

If you prefer to download the corresponding .rpm files for manual QWT installation, these are still available from the repositories (version 4.1.68-1 for Qubes R4.1.2 and version 4.1.69-1 for Qubes R4.2.0). After downloading, copy the file to `dom0` as described in How to copy from dom0 and install it via `sudo dnf install /path/to/rpmfile`. Now you can proceed according to step 3 of the description below.

Warning: These older versions of Qubes Windows Tools will be replaced during the next dom0 update by the current dummy version 4.1.70-1. This can be inhibited by appending the line `exclude=qubes-windows-tools` to the file `/etc/dnf/dnf.conf` in dom0. But this will also stop any further QWT updates - so be sure to remove this line when - hopefully - a new functional version 4.1.71-1 of Qubes Windows Tools will be made available!!!

Note: If you choose to move profiles, drive letter `Q:` must be assigned to the secondary (private) disk.

Note: Xen PV disk drivers are not installed by default. This is because they seem to cause problems (BSOD = Blue Screen Of Death). We're working with upstream devs to fix this. *However*, the BSOD seems to only occur after the first boot and everything works fine after that. **Enable the drivers at your own risk** of course, but we welcome reports of success/failure in any case (backup your VM first!). With disk PV drivers absent `qvm-block` will not work for the VM, but you can still use standard Qubes inter-VM file copying mechanisms. On the other hand, the Xen PV drivers allow USB device access even without QWT installation if `qvm-features stubdom-qrexec` is set as `1`

Below is a breakdown of the feature availability depending on the windows version:

Feature	Windows 7 x64	Windows 8.1/10/11 x64
Qubes Video Driver	+	-
Qubes Network Setup	+	+
Private Volume Setup (move profiles)	+	+
File sender/receiver	+	+
Clipboard Copy/Paste	+	+
Application shortcuts	+	+
Copy/Edit in Disposable VM	+	+
Block device	+	+
USB device	+	+
Audio	+	+

Qubes Windows Tools are open source and are distributed under a GPL license.

Notes:

- Currently only 64-bit versions of Windows 7, 8.1, 10 and 11 are supported by Qubes Windows Tools. Only emulated SVGA GPU is supported (although there has been reports on working GPU passthrough).
- This page documents the process of installing Qubes Windows Tools in version **R4.1**.
- *In testing VMs only* it's probably a good idea to install a VNC server before installing QWT. If something goes very wrong with the Qubes gui agent, a VNC server should still allow access to the OS.

Preparation

Windows 7 only: Before proceeding with the installation we need to disable the Windows mechanism that allows only signed drivers to be installed, because currently the Qubes video driver, available for Windows 7, provided as part of the Windows Tools are not digitally signed with a publicly recognizable certificate. To do that:

1. Start the command prompt as Administrator, i.e. right click on the Command Prompt icon (All Programs -> Accessories) and choose "Run as administrator"
2. In the command prompt type `bcdedit /set testsigning on`
3. Reboot your Windows VM

In the future this step will not be necessary anymore, because we will sign our drivers with a publicly verifiable certificate. However, it should be noted that even now, the fact that those drivers are not digitally signed, this doesn't affect security of the Windows VM in 'any' way. This is because the actual installation `iso` file can be verified as described in step 3 below. The only downside of those drivers not being signed is the inconvenience to the user that he or she must disable the signature enforcement policy before installing the tools.

The Xen PV Drivers bundled with QWT are signed by a Linux Foundation certificate. Thus Windows 10 and 11 do not require this security mitigation.

Warning: it is recommended to increase the default value of Windows VM's `qrexec_timeout` property from 60 (seconds) to, for example, 300. During one of the first reboots after Windows Tools installation Windows user profiles are moved onto the private VM's virtual disk (private.img) and this operation can take some time. Moving profiles and, later on, updating a Windows installation, is performed in an early boot phase when `qrexec` is not yet running, so timeout may occur with the default value. To change the property use this command in `dom0`: *(where <VMname> is the name of your Windows VM)*

```
[user@dom0 ~] $ qvm-prefs <VMname> qrexec_timeout 7200
```

Installing Windows OS as a Qubes VM

Please refer to this page for instructions on how to install Windows in a Qubes VM.

Warning: It is strongly suggested to enable autologon for any Windows HVMs that will have Qubes Tools installed. To do so, run `netplwiz` command from the `Win+R`/Start menu and uncheck the *Users must* *enter a user name and password to use this computer* option.

Installing the Qubes Windows Tools on Windows 7, 8.1, 10 and 11 both as a StandaloneVM as well as a Template VM and a corresponding AppVM is described in the following sections.

Note: Seamless mode is currently not available for windows 10 and 11. Please check the top of this document for the full feature availability breakdown.

1. First, make sure that `qubes-windows-tools` is installed in your system:

   ```
   sudo qubes-dom0-update qubes-windows-tools
   ```

   (If the above command does not work, it could be that the Qubes Tools are not in the stable repo yet. Try installing from the testing repo instead.)

   You can also install the package from testing repositories, where we usually publish new versions first:

   ```
   sudo qubes-dom0-update --enablerepo=qubes-dom0-current-testing qubes-windows-tools
   ```

   If an earlier version of Qubes Windows Tools is already installed, with enabled current-testing repo you need to specify as action to `upgrade` the existing package, because the default action is `install`, which will fail if it detects that QWT is already present in Dom0:

   ```
   sudo qubes-dom0-update --enablerepo=qubes-dom0-current-testing --action=upgrade qubes-windows-tools
   ```

   This package brings the ISO with Qubes Windows Tools that is passed to the VM when `--install-windows-tools` is specified for the `qvm-start` command. Please note that none of this software ever runs in Dom0 or any other part of the system except for the Windows AppVM in which it is to be installed.

2. **For Windows 8.1, 10 and 11:** From the Windows command line, disable hibernation in order to avoid incomplete Windows shutdown, which may lead to corruption of the VM’s disk.

   Also, these versions of Windows won’t show the CD-ROM drive after starting the qube with `qvm-start vm --cdrom ...` or `qvm-start ... --install-windows-tools`. The solution is to disable hibernation in Windows with this command. (That command is included in QWT’s setup but it’s necessary to run it manually in order to be able to open QWT’s setup ISO/CD-ROM in Windows).

3. To install the Qubes Windows Tools in a Windows VM one should start the VM passing the additional option `--install-windows-tools`:

   ```
   qvm-start <VMname> --install-windows-tools
   ```

   Once the Windows VM boots, a CDROM should appear in the ‘My Computer’ menu (typically as `D:`

or `E:`) with the setup program `qubes-tools-x64.msi` in its main directory.

4. Install Qubes Windows Tools by starting `qubes-tools-x64.msi` (logged in as administrator), optionally selecting the `Xen PV disk drivers`. For installation in a template, you should select `Move user profiles`.

   Several times, Windows security may ask for confirmation of driver installation. Driver installation has to be allowed; otherwise the installation of Qubes Windows Tools will abort.

   If during installation, the Xen driver requests a reboot, select “No” and let the installation continue - the system will be rebooted later.

5. After successful installation, the Windows VM must be shut down and started again, possibly a couple of times. On each shutdown, wait until the VM is really stopped, i.e. Qubes shows no more activity.

6. Qubes will automatically detect that the tools have been installed in the VM and will set appropriate properties for the VM, such as `qrexec_installed`, `guiagent_installed`, and `default_user`. This can be verified (but is not required) using the `qvm-prefs` command *(where <VMname> is the name of your Windows VM)*:

   ```
       [user@dom0 ~] $ qvm-prefs <VMname>
   ```

It is advisable to set some other parameters in order to enable audio and USB block device access, synchronize the Windows clock with the Qubes clock, and so on:

```
        [user@dom0 ~] $ qvm-features <VMname> audio-model ich9
        [user@dom0 ~] $ qvm-features <VMname> stubdom-qrexec 1
        [user@dom0 ~] $ qvm-features <VMname> timezone localtime

For audio, the parameter `audio-model`can be selected as `ich6` or
`ich9`; select the value that gives the best audio quality. Audio quality
may also be improved by setting the following parameters, but this can
depend on the Windows version and on your hardware:

        [user@dom0 ~] $ qvm-features <VMname> timer-period 1000
        [user@dom0 ~] $ qvm-features <VMname> out.latency 10000
        [user@dom0 ~] $ qvm-features <VMname> out.buffer-length 4000

With the value `localtime` the dom0 `timezone` will be provided to
virtual hardware, effectively setting the Windows clock to that of Qubes.
With a digit value (negative or positive) the guest clock will have an
offset (in seconds) applied relative to UTC.
```

1. Reboot Windows. If the VM starts, but does not show any window then shutdown Windows from the Qube manager, wait until it has really stopped, and reboot Windows once more.

2. Now the system should be up, with QWT running correctly.

3. **Windows 7 only:** Optionally enable seamless mode on VM startup. This can be done by setting appropriate values in the Windows registry:

    - Start the command prompt as administrator, i.e. right click on the Command Prompt icon (All Programs -> Accessories) and choose “Run as administrator”
    - In the command prompt type `regedit`
    - In the registry editor, position to the key `\HKEY_LOCAL_MACHINE\Software\Invisible Things Lab\Qubes Tools\`
    - Change the value `SeamlessMode` from 0 to 1
    - Position to the key `\HKEY_LOCAL_MACHINE\Software\Invisible Things Lab\Qubes Tools\qga\`
    - Change the value `SeamlessMode` from 0 to 1
    - Terminate the registry editor.

After the next boot, the VM will start in seamless mode.

```
 If Windows is used in a TemplateVM / AppVM combination, this registry
fix has to be applied to the TemplateVM, as the `HKLM` registry key
belongs to the template-based part of the registry.
```

1. Lastly to enable file copy operations to a Windows VM, the `default_user` property of this VM should be set to the `<username>` that you use to login to the Windows VM. This can be done via the following command on a `dom0` terminal: *(where <VMname> is the name of your Windows VM)*

    ```
    [user@dom0 ~] $ qvm-prefs <VMname> default_user <username>
    ```

Warning: If this property is not set or set to a wrong value, files copied to this VM are stored in the folder

C:\Windows\System32\config\systemprofile\Documents\QubesIncoming<source_VM>

If the target VM is an AppVM, this has the consequence that the files are stored in the corresponding TemplateVM and so are lost on AppVM shutdown.

Installing Xen’s PV drivers in the VM will lower its resources usage when using network and/or I/O intensive applications, but *may* come at the price of system stability (although Xen’s PV drivers on a Windows VM are usually very stable). They can be installed as an optional part of Qubes Windows Tools (QWT), which bundles Xen’s PV drivers.

Notes about using Xen’s VBD (storage) PV driver:

- **Windows 7:** Installing the driver requires a fully updated VM or else you’ll likely get a BSOD (“Blue Screen Of Death”) and a VM in a difficult to fix state. Updating Windows takes *hours* and for casual usage there isn’t much of a performance between the disk PV driver and the default one; so there is likely no need to go through the lengthy Windows Update process if your VM doesn’t have access to untrusted networks and if you don’t use I/O intensive apps or attach block devices. If you plan to update your newly installed Windows VM it is recommended that you do so *before* installing Qubes Windows Tools. Installing the driver will probably cause Windows 7 activation to become invalid, but the activation can be restored using the Microsoft telephone activation method.
- The option to install the storage PV driver is disabled by default in Qubes Windows Tools
- In case you already had QWT installed without the storage PV driver and you then updated the VM,

you may then install the driver by again starting the QWT installer and selecting the change option.

Using Windows AppVMs in seamless mode

Note: This feature is only available for Windows 7

Once you start a Windows-based AppVM with Qubes Tools installed, you can easily start individual applications from the VM (note the `-a` switch used here, which will auto-start the VM if it is not running):

```
[user@dom0 ~] $ qvm-run -a my-win-appvm explorer.exe
```

Also, the inter-VM services work as usual – e.g. to request opening a document or URL in the Windows AppVM from another VM:

```
[user@dom0 ~] $ qvm-open-in-vm my-win-appvm roadmap.pptx

[user@dom0 ~]$ qvm-open-in-vm my-win-appvm https://invisiblethingslab.com
```

… just like in the case of Linux AppVMs. Of course all those operations are governed by central policy engine running in Dom0 – if the policy doesn't contain explicit rules for the source and/or target AppVM, the user will be asked whether to allow or deny the operation.

Inter-VM file copy and clipboard works for Windows AppVMs the same way as for Linux AppVM (except that we don't provide a command line wrapper, `qvm-copy-to-vm` in Windows VMs) – to copy files from Windows AppVMs just right-click on the file in Explorer, and choose: Send To-> Other AppVM.

To simulate Ctrl-Alt-Delete in the HVM (SAS, Secure Attention Sequence), press Ctrl-Alt-Home while having any window of this VM in the foreground.

Changing between seamless and full desktop mode

You can switch between seamless and "full desktop" mode for Windows HVMs in their settings in Qubes Manager. The latter is the default.

Using template-based Windows AppVMs

Qubes allows HVM VMs to share a common root filesystem from a select Template VM, just as for Linux AppVMs. This mode is not limited to Windows AppVMs, and can be used for any HVM (e.g. FreeBSD running in a HVM).

In order to create an HVM TemplateVM, the type "TemplateVM" has to be selected on creating the VM. Then set memory as appropriate, and install the Windows OS (or any other OS) into this template the same way as you would install it into a normal HVM – please see instructions on this page.

If you use this Template as it is, then any HVMs that use it will effectively be DisposableVMs - the User directory will be wiped when the HVM is closed down.

If you want to retain the User directory between reboots, then it would make sense to store the `C:\Users` directory on the 2nd disk which is automatically exposed by Qubes to all HVMs. This 2nd disk is backed by the `private.img` file in the AppVMs' and is not reset upon AppVMs reboot, so the user's directories and profiles would survive the AppVMs reboot, unlike the "root" filesystem which will be reverted to the "golden image" from the Template VM automatically. To facilitate such separation of user profiles, Qubes Windows Tools provide an option to automatically move `C:\Users` directory to the 2nd disk backed by `private.img`. It's a selectable feature of the installer. For Windows 7, it requires the private disk to be renamed to `Q:` before QWT installation (see above); for Windows 8.1, 10 and 11, this renaming occurs during QWT installation automatically. If that feature is selected during installation, completion of the process requires two reboots:

- The private disk is initialized and formatted on the first reboot after tools installation. It can't be done **during** the installation because Xen mass storage drivers are not yet active.
- User profiles are moved to the private disk on the next reboot after the private disk is initialized. Reboot is required because the "mover utility" runs very early in the boot process so OS can't yet lock any files in there. This can take some time depending on the profiles' size and because the GUI agent is not yet active dom0/Qubes Manager may complain that the AppVM failed to boot. That's a false alarm (you can increase the AppVM's default boot timeout using `qvm-prefs`), the VM should appear "green" in Qubes Manager shortly after.

It also makes sense to disable Automatic Updates for all the template-based AppVMs – of course this should be done in the Template VM, not in individual AppVMs, because the system-wide settings are stored in the root filesystem (which holds the system-wide registry hives). Then, periodically check for updates in the Template VM and the changes will be carried over to any child AppVMs.

Once the template has been created and installed it is easy to create AppVMs based on it, by selecting the type "AppVM" and a suitable template.

Using Windows disposables

Windows qubes can be used as disposables, like any other Linux-based qubes. On creating a template for Windows disposables, certain preparations have to be executed:

- Create an AppVM based on a Windows TemplateVM.
- Start this AppVM and insert a link to the command prompt executable in the `Autostart` directory of the Windows menu tree:
    - **For Windows 7:** - If the Windows qube started in seamless mode, hit the Windows keyboard key while the cursor is positioned in a window of this VM. In non-seamless mode, klick on the Start button. In both cases, the Windows menu will be displayed.
        - Position into the `Autostart` submenu.
    - **For Windows 8.1, 10 or 11:** - Type Win+R to open the execution Prompt.
        - Type `shell:startup`.
        - An explorer window will open, which is positioned to the `Autostart` folder.
    - Right-click and select the option "New -> Link".
    - Select `C:\Windows\System32\CMD.exe` as executable.
    - Name the link, e.g. as `Command Prompt`.
    - Close the Window with `OK`.
    - Shut down this AppVM.
- In the Qube Manager, refresh the applications of the newly created AppVM and select those applications that you want to make available from the disposable. Alternatively, in dom0 execute the

command `qvm-sync-appmenus <VMname>`, *where <VMname> is the name of your windows qube*.
- In the Qube Manager, go to the “Advanced” tab and enable the option `Disposable template` for your Windows qube. Alternatively, in dom0 execute the commands `qvm-prefs <VMname> template_for_dispvms True` and `qvm-features <VMname> appmenus-dispvm 1`.
- Click `Apply`.
- Still in the Advanced tab, select your Windows qube as its own `Default disposable template`. Alternatively, in dom0 execute the command `qvm-prefs <VMname> default_dispvm <VMname>`.
- Close the Qube Manager by clicking `OK`.

Now you should have a menu `Disposable: <VMname>` containing the applications that can be started in a disposable Windows VM. If you set the newly created and configured Windows VM as `Default disposable template` for any other Windows- (or Linux-) based qube, this qube can use the Windows-based dispvm like any other disposable.

For further information on usage of disposables, see How to use disposables.

Caution: *If a Windows-based disposable is used from another qube via the `Open/Edit in DisposableVM` command, this disposable may not close automatically, due to the command prompt window still running in this dispvm. In this case, the disposable has to be shut down manually.*

Installation logs

If the install process fails or something goes wrong during it, include the installation logs in your bug report. They are created in the `%TEMP%` directory, by default `<user profile>\AppData\Local\Temp`. There are two text files, one small and one big, with names starting with `Qubes_Windows_Tools`.

Uninstalling QWT is supported. After uninstalling you need to manually enable the DHCP Client Windows service, or set IP settings yourself to restore network access.

Configuration

Various aspects of Qubes Windows Tools (QWT) can be configured through the registry. The main configuration key is located in `HKEY_LOCAL_MACHINE\SOFTWARE\Invisible Things Lab\Qubes Tools`. Configuration values set on this level are global to all QWT components. It’s possible to override global values with component-specific keys, this is useful mainly for setting log verbosity for troubleshooting. Possible configuration values are:

Name	Type	Description	Default value
LogDir	String	Directory where logs are created	c:\Program Files\Invisible Things Lab\Qubes Tools\log
LogLevel	DWORD	Log verbosity (see below)	2 (INFO)
LogRetention	DWORD	Maximum age of log files (in seconds), older logs are automatically deleted	604800 (7 days)

Possible log levels:

Level	Title	Description
1	Error	Serious errors that most likely cause irrecoverable failures
2	Warning	Unexpected but non-fatal events
3	Info	Useful information (default)
4	Debug	Internal state dumps for troubleshooting
5	Verbose	Trace most function calls

Debug and Verbose levels can generate large volume of logs and are intended for development/ troubleshooting only.

To override global settings for a specific component, create a new key under the root key mentioned above and name it as the executable name, without `.exe` extension.

Component-specific settings currently available:

Component	Setting	Type	Description	Default value
qga	DisableCursor	DWORD	Disable cursor in the VM. Useful for integration with Qubes desktop so you don’t see two cursors. Can be disabled if you plan to use the VM through a remote desktop connection of some sort. Needs gui agent restart to apply change (locking OS/logoff should be enough since qga is restarted on desktop change).	1

Troubleshooting

If the VM is inaccessible (doesn’t respond to qrexec commands, gui is not functioning), try to boot it in safe mode:

- `[user@dom0 ~] $ qvm-start --debug <VMname>`
- Enable boot options and select Safe Mode (method depends on the Windows version; optionally with networking)

Safe Mode should at least give you access to logs (see above).

Please include appropriate logs when reporting bugs/problems. Logs contain the QWT version. If the OS crashes (BSOD) please include the BSOD code and parameters in your bug report. The BSOD screen should be visible if you run the VM in debug mode (`qvm-start --debug vmname`). If it’s not visible or the VM reboots automatically, try to start Windows in safe mode (see above) and 1) disable automatic restart on BSOD (Control Panel - System - Advanced system settings - Advanced - Startup and recovery), 2) check the system event log for BSOD events. If you can, send the `memory.dmp` dump file from `C:\Windows`.

Xen logs in dom0 (`/var/log/xen/console/guest-*`) are also useful as they contain pvdrivers diagnostic output.

If a specific component is malfunctioning, you can increase its log verbosity as explained above to get more troubleshooting information. Below is a list of components:

Component	Description
qrexec-agent	Responsible for most communication with Qubes (dom0 and other domains), secure clipboard, file copying, qrexec services.
qrexec-wrapper	Helper executable that’s responsible for launching qrexec services, handling their I/O and vchan communication.
qrexec-client-vm	Used for communications by the qrexec protocol.
qga	Gui agent.
QgaWatchdog	Service that monitors session/desktop changes (logon/logoff/locking/UAC…) and simulates SAS sequence (Ctrl-Alt-Del).
qubesdb-daemon	Service for accessing Qubes configuration database.
network-setup	Service that sets up network parameters according to VM’s configuration.
prepare-volume	Utility that initializes and formats the disk backed by `private.img` file. It’s registered to run on next system boot during QWT setup, if that feature is selected (it can’t run *during* the setup because Xen block device drivers are not yet active). It in turn registers move-profiles (see below) to run at early boot.
relocate-dir	Utility that moves user profiles directory to the private disk. It’s registered as an early boot native executable (similar to chkdsk) so it can run before any profile files are opened by some other process. Its log is in a fixed location: `C:\move-profiles.log` (it can’t use our common logger library so none of the log settings apply).

If there are network-related issues, the qube doesn’t resolve DNS and has trouble accessing the Internet, this might be an issue with the PV Network Drivers.

In this case it’s recommended that the PV Network Drivers be unchecked during installation of Qubes Windows Tools as seen in the screenshot below.

Updates

When we publish a new QWT version, it’s usually pushed to the `current-testing` or `unstable` repository first. To use versions from current-testing, run this in dom0:

```
[user@dom0 ~] $ sudo qubes-dom0-update --enablerepo=qubes-dom0-current-
```

```
testing qubes-windows-tools
```

That command will download a new QWT `iso` file from the testing repository. It goes without saying that you should **backup your VMs** before installing anything from testing repos.