



acontis technologies GmbH

SOFTWARE

Hypervisor-WindowsGuest-Guide

acontis Real-time Hypervisor and Windows guest

Version 1.0

Edition: November 3, 2022

© Copyright **acontis technologies GmbH**

Neither this document nor excerpts therefrom may be reproduced, transmitted, or conveyed to third parties by any means whatever without the express permission of the publisher. At the time of publication, the functions described in this document and those implemented in the corresponding hardware and/or software were carefully verified; nonetheless, for technical reasons, it cannot be guaranteed that no discrepancies exist. This document will be regularly examined so that corrections can be made in subsequent editions. Note: Although a product may include undocumented features, such features are not considered to be part of the product, and their functionality is therefore not subject to any form of support or guarantee.

Table of Contents

1	Introduction	4
2	Create folder for running the VM	5
3	Copy Windows ISO installation media to Linux	6
3.1	Using Filezilla (recommended)	6
3.2	Using a Windows Share	6
4	Guest Configuration	7
5	Guest Installation	8
6	Windows Installation	9
7	Hypervisor Guest Tools	11
8	RTOS Communication Support	13
9	Power & sleep	16
10	Windows and Real-time guest in parallel	17
11	Windows System Control	19
11.1	System reboot/shutdown	19
11.2	Start an App on RT-Linux	19
12	Automatic Guest Start	21
13	Graphics (Desktop) configuration	22
13.1	Display both, Host and Guest Desktop	22
13.2	Display Guest Desktop only	22
13.3	Guest Desktop via Pass-Through	24
14	Boot Customization	25
14.1	Splash Screen	25
15	Clone an existing VM	26

1 Introduction

This guide describes how to set up a Windows guest in the acontis Hypervisor. It is assumed, the steps listed in the Hypervisor Quick Start guide have been successfully executed.

2 Create folder for running the VM

In the first step, you have to create a folder where the VM shall be stored. The template folder has to be copied into a new VM specific folder, we will use the folder name `vm1` in this example.

```
$ cd /hv/VMs  
$ cp -R vm_template vm1
```

3 Copy Windows ISO installation media to Linux

In a first step, we need to copy the Windows installation media (ISO file) into the Hypervisor filesystem.

3.1 Using Filezilla (recommended)

See details in the [Ubuntu Guest Guide](#)

3.2 Using a Windows Share

Here we will show how this can be accomplished using a network share (`//NameOrIpAddressOfPcWithShare/NameOfShare`) in the Windows host system. We assume the network share has a directory `/hv` containing the file `Windows.iso`. We need to install the network client software (cifs) in the hypervisor and then mount the Windows network share.

```
$ sudo apt-get install cifs-utils
```

```
$ sudo mount -t cifs //NameOrIpAddressOfPcWithShare/NameOfShare /mnt -o  
→ user=NameOfUserWithAccessToShare
```

Copy the Windows 10 ISO image to `/hv/VMs/vm1`.

```
$ cp /mnt/hv/Windows10.iso /hv/VMs/vm1/Windows10.iso
```

The mount point `/mnt` isn't needed anymore and thus will be unmounted.

```
$ sudo umount /mnt
```

4 Guest Configuration

Prior to installing the guest, we need to configure the respective virtual machine (e.g. number of CPU cores, network settings etc.). The configuration file `vmconfig.sh` is located in the VM directory. The meaning of each configuration setting is explained in detail in this file. You need to edit this file and adjust according to your needs and environment.

The number of CPU cores must **not** exceed the number of physical cores available in the system and not assigned to Real-time guests. For example, if on a *quad core* CPU, you need 2 cores for Real-time guests, the number of cores for the Windows guest must not exceed 2.

```
$ gedit /hv/VMs/vm1/vmconfig.sh
```

```
# Adapt following lines to your system and needs:
vmname=...
vmid=...
windows_guest=...
cdrom_iso=...
num_cpus=...
ramsize=...
```

By default, the network connection is set up automatically (using DHCP). Please check the hypervisor manual for other settings.

Caution: Automatic network setting will only work, if the Ethernet cable is connected!

Caution: Please do not configure more CPUs than physically available (CPUs used for the Real-time OS are not available for the Windows guest). Example: The maximum number of CPUs on a quad-core CPU where 1 CPU is used for the Real-time OS is 3.

Caution: Please do not configure more RAM than available. The VM may unexpectedly crash if too much RAM is configured. You can determine the available RAM as follows:

```
$ cat /proc/meminfo | grep MemAvailable
```

5 Guest Installation

After VM configuration you need to start the script `vmrun.sh` for the first time. The remote viewer will start and boot the guest.

```
$ cd /hv/VMs/vm1
$ sudo ./vmrun.sh
```

If `cdrom_iso` is configured with an ISO file you need to press a key to boot from it:



Fig. 5.1: Boot from ISO file.

Hint: In case no key was pressed in time or `cdrom_iso` was not configured the EFI shell will be started.

- Option a) You can leave into BIOS by entering `exit`. If you also leave BIOS the boot begins again and you can press a key to boot from CD.
- **Option b) You can manually start the CD's bootloader by entering**

```
- $ FS0:
- $ \EFI\BOOT\BOOTX64.EFI
```


6 Windows Installation

- Follow the steps as usual until the **Where do you want to install Windows** dialog shows up

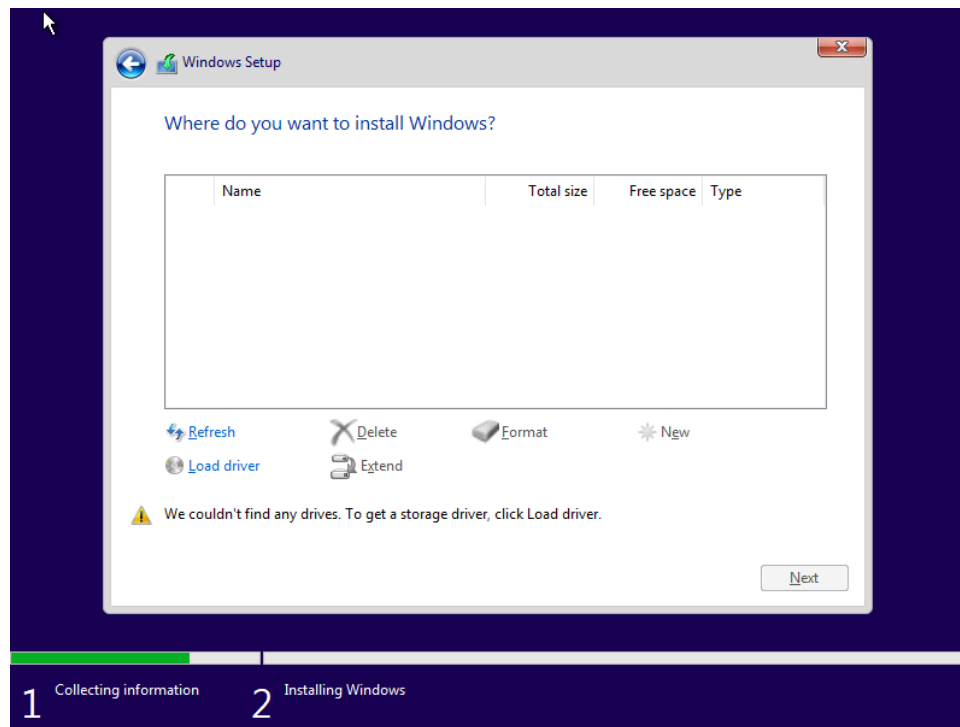


Fig. 6.1: Where do you want to install Windows.

You need to browse to the CD Drive with the virtio drivers. Then select the appropriate driver (viostor).

- Then select **Load driver** → OK → **Red Hat VirtIO SCSI controller” (W10)** → Next

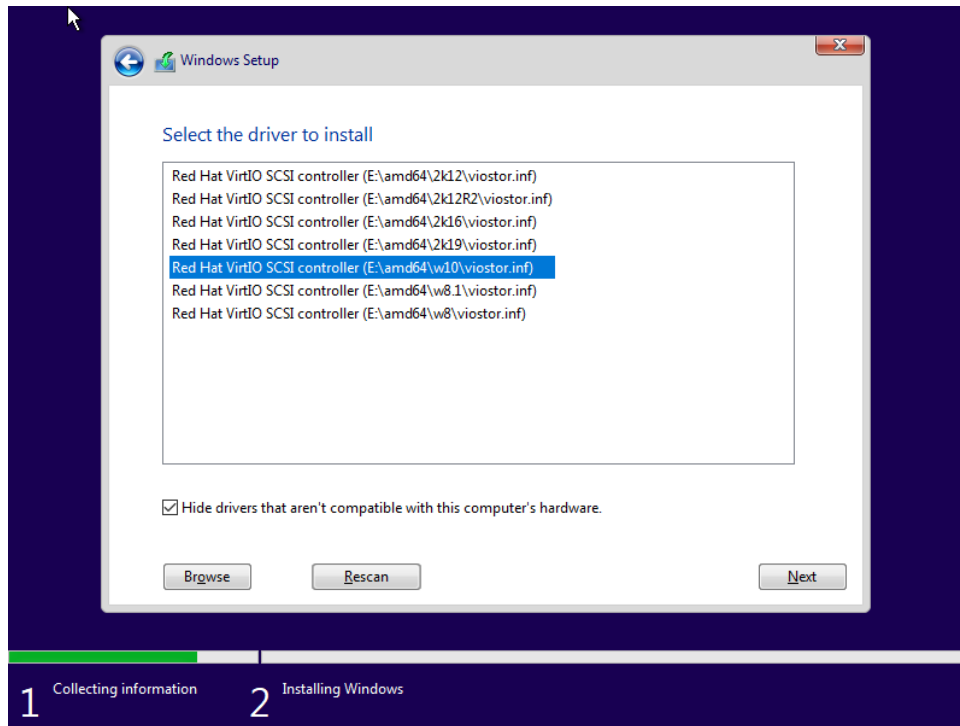


Fig. 6.2: Windows Install driver select.

- Select your drive (50GB)

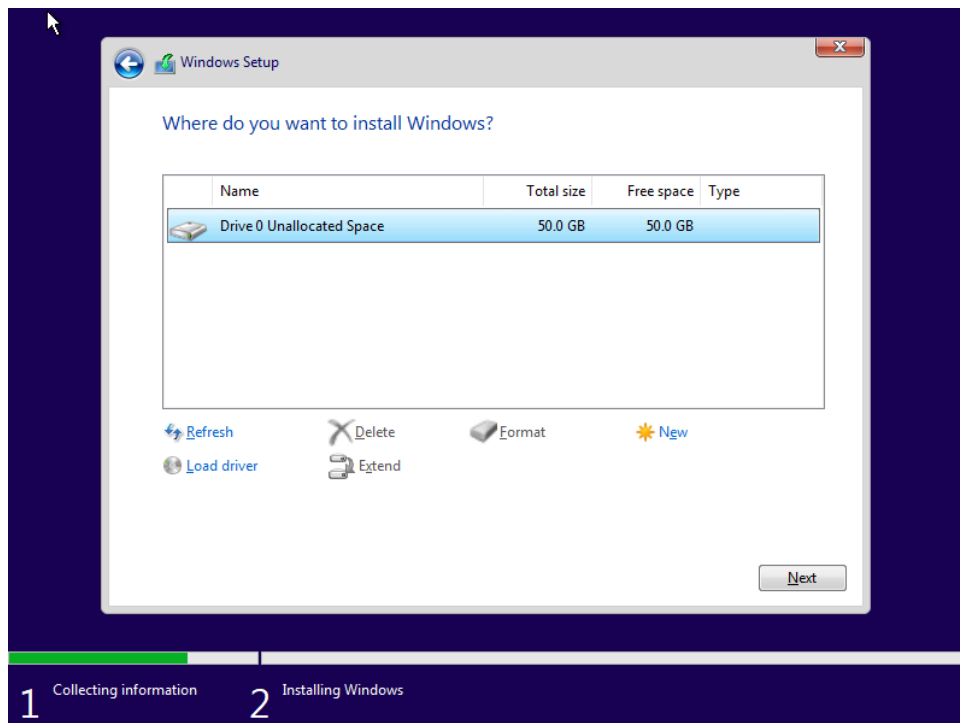


Fig. 6.3: Windows Install disk.

- Finish Installation (please note, there is no network/internet available in this setup)

7 Hypervisor Guest Tools

After successfully installing Windows you need to install the SPICE Windows guest tools.

- Open the Windows Explorer and install the spice guest tools.

The installation file is located in CD Drive (E:) `virtio-win-0.1.215:virtio-win-guest-tools`. Select and start to install these tools. In case a message box appears to confirm driver installation, please confirm install the driver(s).

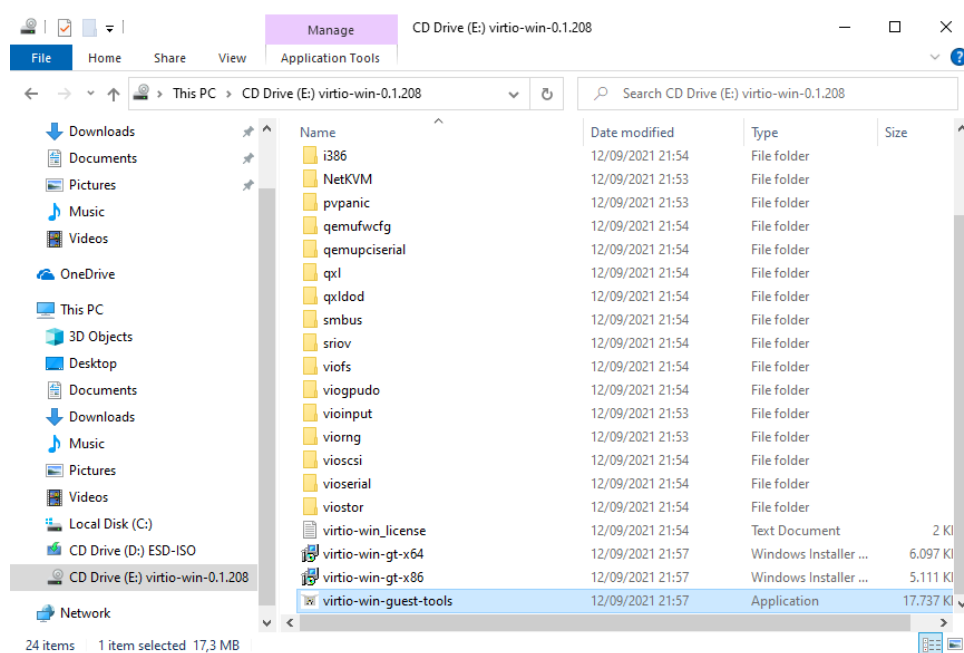


Fig. 7.1: SPICE/virtio-win guest tools.

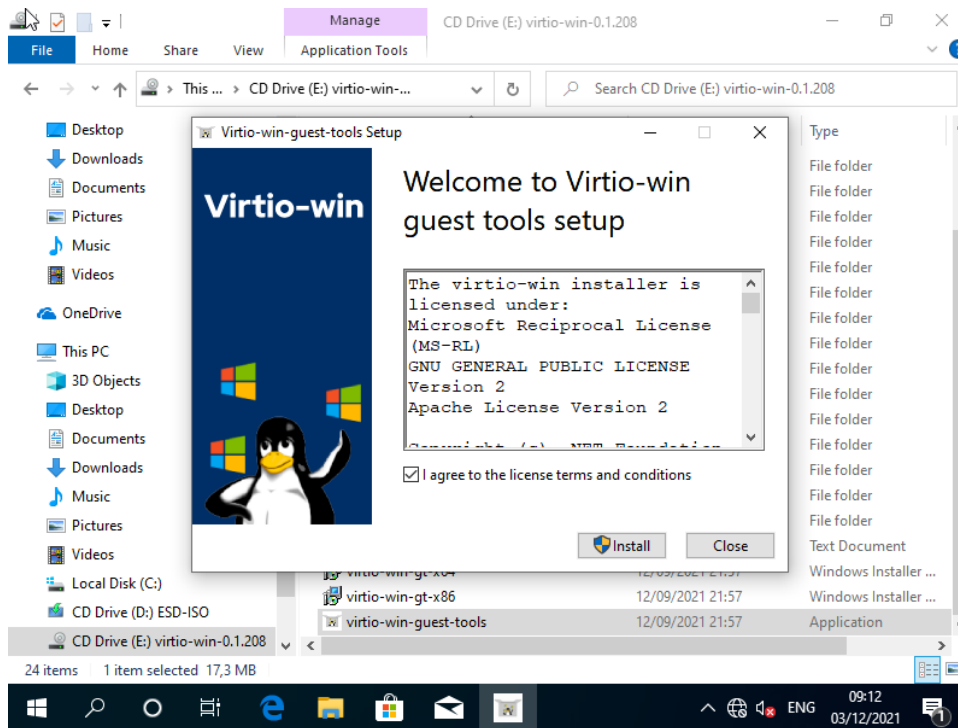


Fig. 7.2: SPICE/virtio-win guest tools setup screen.

After the setup finished, you may not be able to use the mouse anymore. This is related to a new driver installed which is not supported by the viewer application. In any case, close the setup message box (type enter, while the focus is on the finish box).

- Shutdown Windows

You need to shut down the guest (do **NOT** reboot Windows!). If this not possible inside the guest, please use the `./vmshtdn.sh` script. If shutdown is blocked, you need to close the setup windows before (e.g. using tab keys to select the finish button).

- Adjust the file `vmconfig.sh` (to avoid booting the installation media again)

```
$ gedit /hv/VMs/vm1/vmconfig.sh
```

```
comment "cdrom_iso" with #
```

- Start the Windows guest

```
$ ./vmrun.sh
```

Due to hardware changes, Windows may automatically reboot once. Mouse and desktop may still not work properly. In this case, please install all the latest Windows updates.

8 RTOS Communication Support

To communicate with an RTOS via the Hypervisor we need to install `RTOSVisor.exe` containing the required drivers and packages.

Hint: In case you got a **test** version of the acontis hypervisor, **unsigned** drivers are included. You need to enable *testsigning* **before** installing the guest support.

- Step 1: Run Command Prompt as **administrator**.
- Step 2: Input the command in the window: `bcdedit /set testsigning on` and press Enter. Reboot the guest.

Open CMD prompt in Windows guest and enter the following:

```
Call \\10.0.2.4\qemu\RTOSVisor.exe
```

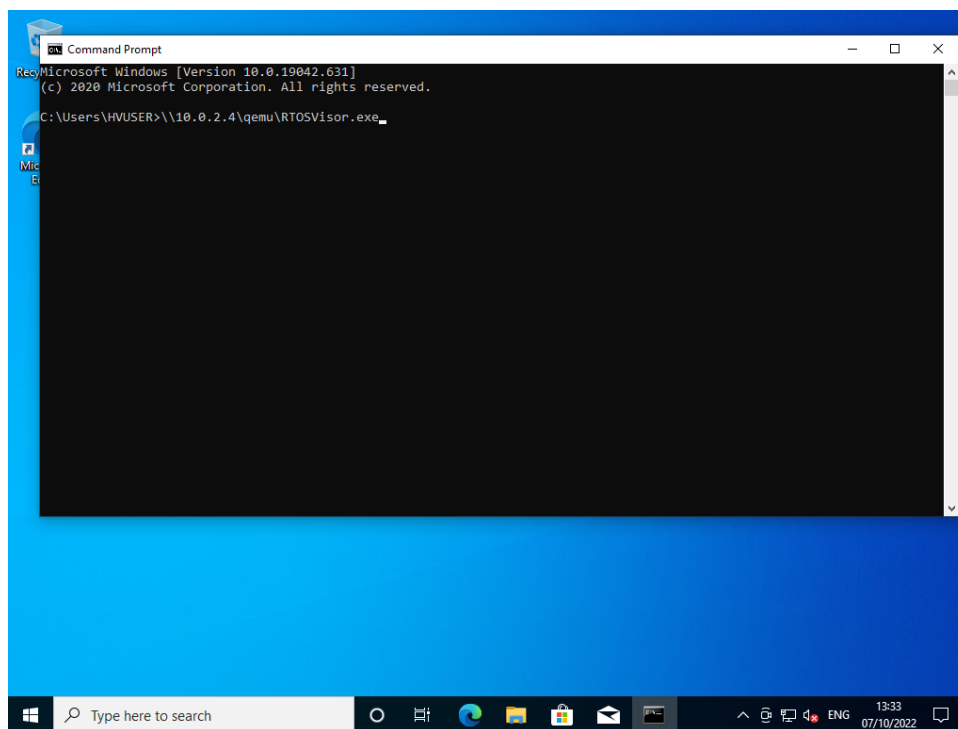


Fig. 8.1: Setup command.

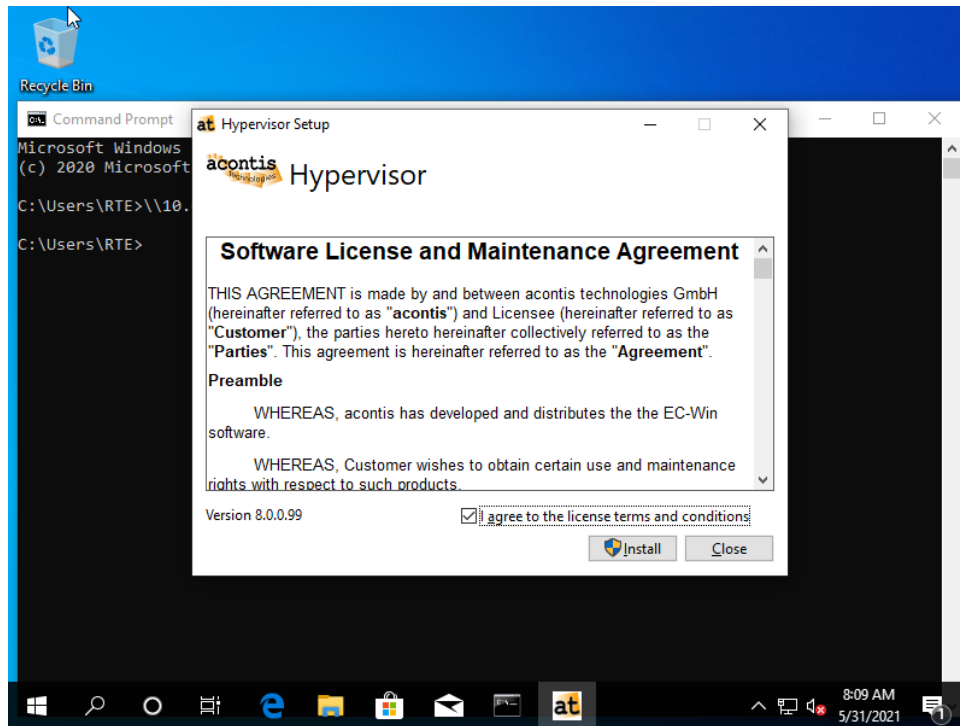


Fig. 8.2: Running setup.

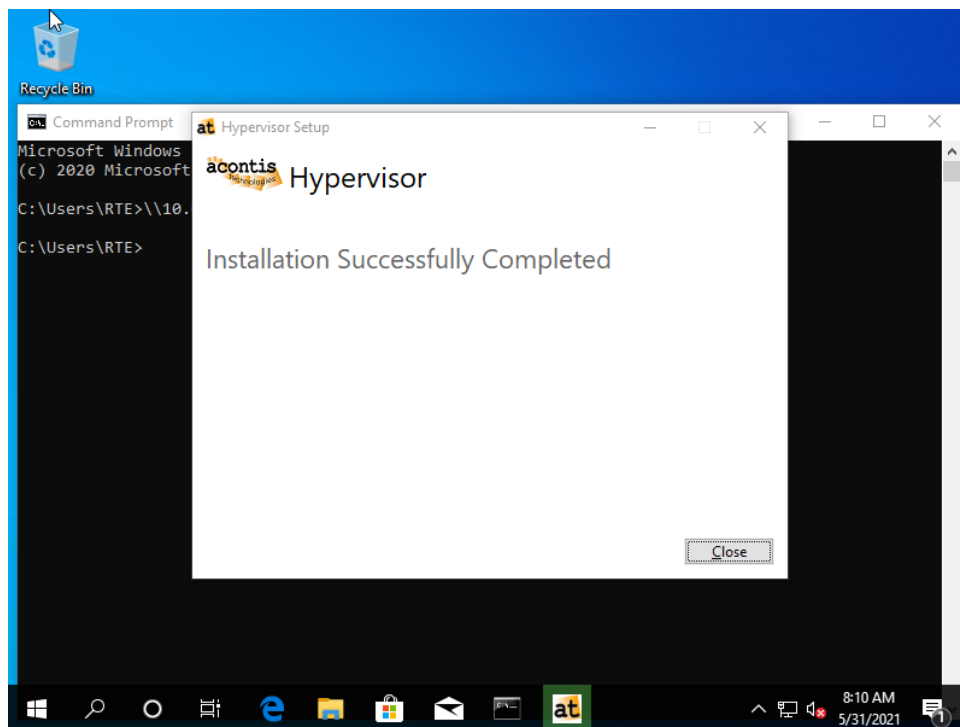


Fig. 8.3: Installation successful.

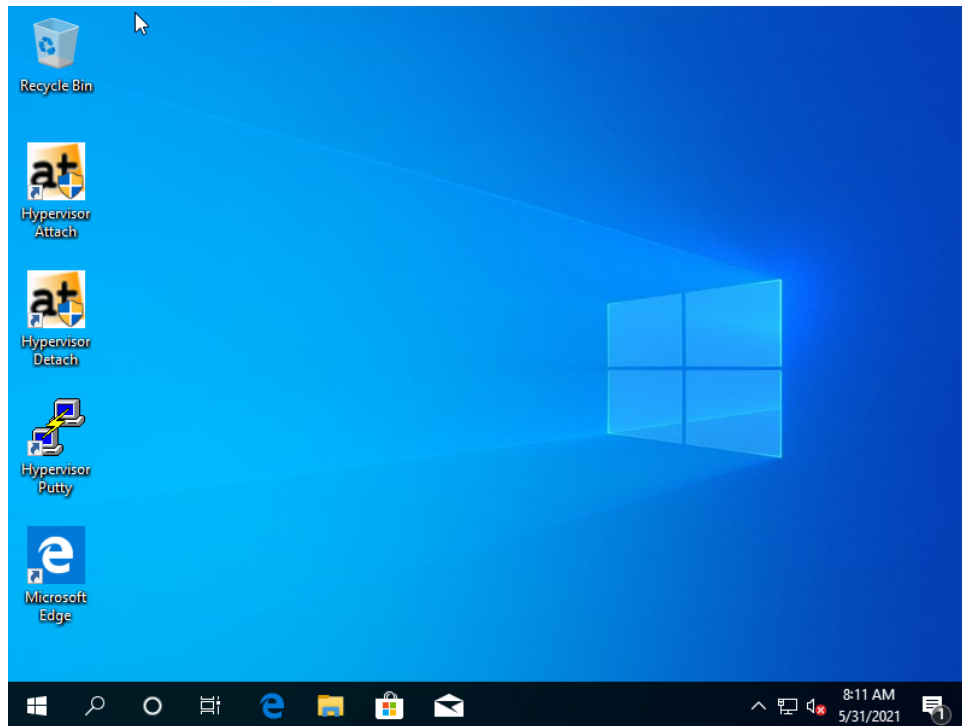


Fig. 8.4: New desktop icons.

9 Power & sleep

Sleep and Hibernate are currently not supported (S3 or S4 mode). Disable Sleep (Windows Settings – System – Power and sleep):

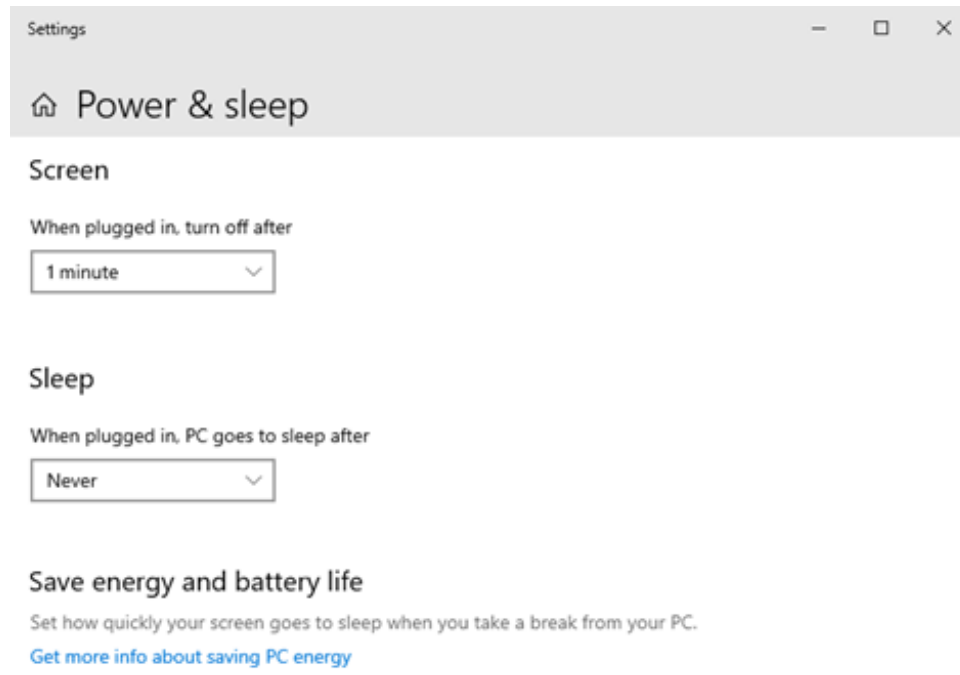


Fig. 9.1: Power and Sleep settings.

10 Windows and Real-time guest in parallel

In this step, we will run Windows and Real-Time Linux in parallel.

- Shutdown Windows, do **NOT** reboot!

- Reboot the Linux host

```
$ sudo reboot
```

- After system reboot, open a shell terminal again (right click on desktop and select *'Open Terminal here'* or press CTRL + ALT + T). Then, run the Real-time Linux guest

```
$ cd /hv/lx
$ sudo ./lx.sh
$ ./dbgcon.sh
```

- Open a second shell terminal

- Start the Windows guest

```
$ cd /hv/VMs/vm1
$ ./vmrun.sh
```

- After logging in into Windows, execute “Hypervisor Attach” (Desktop icon)

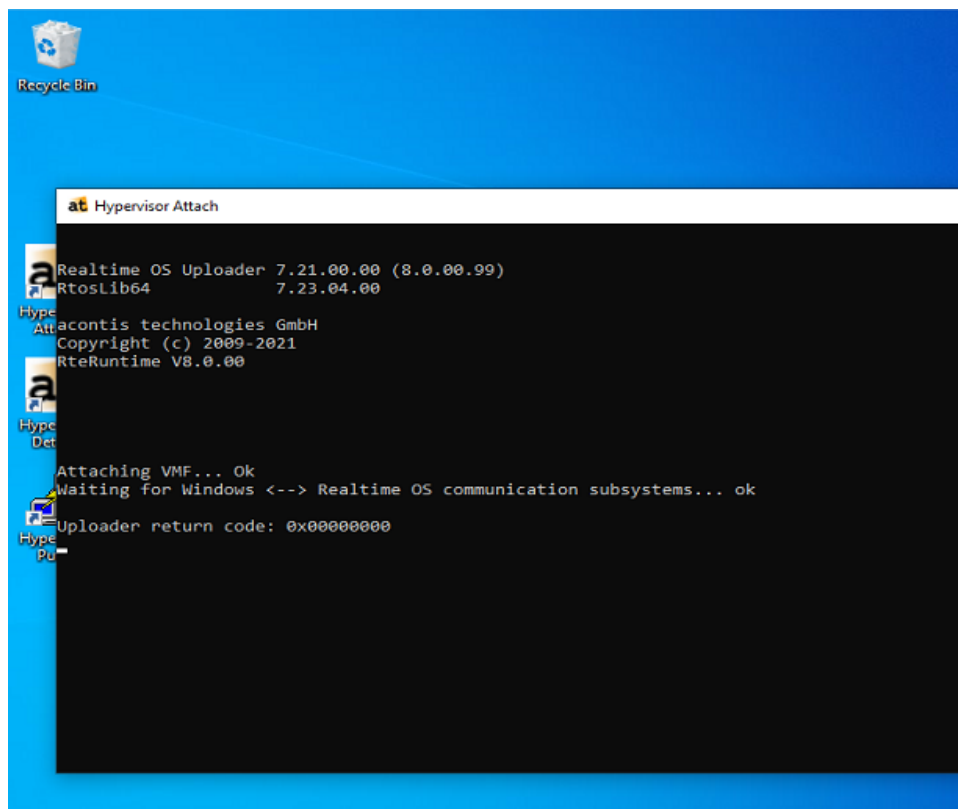


Fig. 10.1: Hypervisor Attach.

- Open the RT-Linux shell: execute “Hypervisor Putty” (Desktop icon)

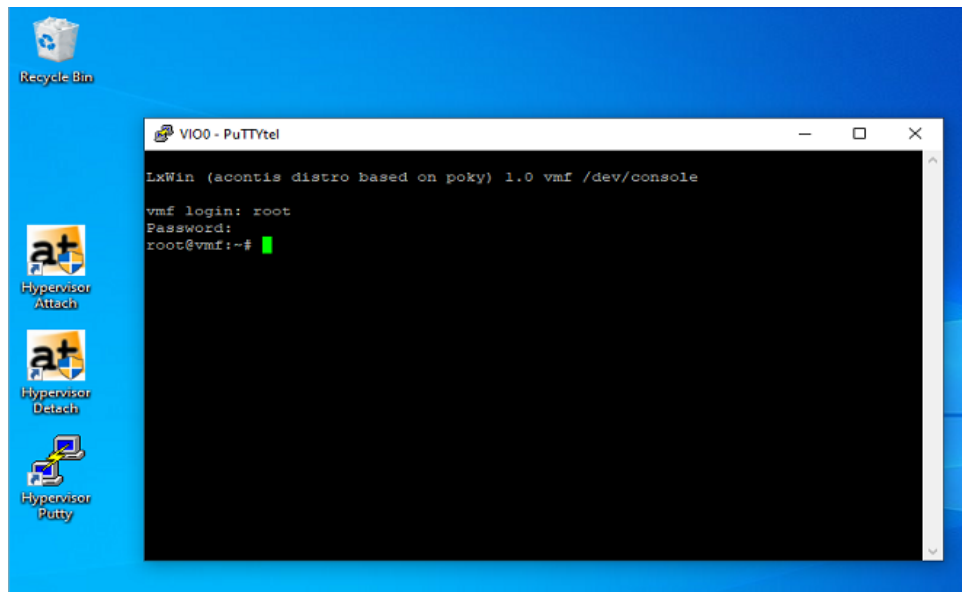


Fig. 10.2: Linux Shell.

- Log in into Real-Time Linux and run the Real-time demo:

```
$ vmf64 login: root
$ password: root
$ RealtimeDemo
```

11 Windows System Control

11.1 System reboot/shutdown

In this section we will show how the whole system can be rebooted or shutdown from within the Windows guest

Hint: This feature is implemented in `/hv/VMs/vm1/shutdown_svc.sh`.

- Start the Windows guest
- Open the Explorer
- Switch to folder `\\10.0.2.4\qemu\WinTools\DesktopShortcuts`
- Copy all the shortcuts onto your desktop (you may have to adjust the shortcuts based on your Windows version and language)
- Run the “*System Shutdown*” or “*System Reboot*” shortcut

Hint: To avoid being asked to allow execution, you may adjust the related Windows settings.

Open the Control Panel, select Internet Options, select security, select local intranet, select sites, select advanced and add `\\10.0.2.4\qemu`

11.2 Start an App on RT-Linux

In this section we will show how to execute an application on the RT-Linux guest from within the Windows guest.

Caution: It is assumed that

- all the provided desktop shortcuts have been placed on the desktop, see last chapter.
- RT-Linux was started before the Windows guest.

- Start the Windows guest
- Download and install .. <https://www.putty.org/>
- Execute the “*Hypervisor Attach*” shortcut
- Open putty and connect to `192.168.157.2`, accept the connection (do **not** login)
- Close putty (we only need to execute this step to accept the security keys, they are needed to run the RT-Linux application)
- Execute the “*Run RT-Linux Application*” shortcut
- A message should be shown.

Hint: This shortcuts executes the "remote_cmd_exec" sample script located in /hv/lx/rtfiles/remote_cmd_exec.sh.

You may adjust this file according to your needs.

12 Automatic Guest Start

In this section we will show how the guest can be started automatically after hypervisor boot.

This will start the VM automatically during boot. The guest desktop will **not** be shown:

```
$ gedit /hv/VMs/vm1/vm_start.sh
$ uncomment ./vmrun.sh
```

The guest desktop can be started manually:

```
$ cd /hv/VMs/vm1
$ ./vmview.sh
```

Hint: Automatic VM startup is done on behalf of the *systemd* service controlled via `/etc/systemd/system/hv_vm1_start.service`.

This service can be enabled or disabled as shown below (it is enabled by default)

```
$ sudo systemctl enable /hv/services/hv_vm1_start.service
$ sudo systemctl disable hv_vm1_start
```

The **Shutdown Hook** `/hv/VMs/vm1/vm_shutdown_hook.sh` is called when the guest is shut down (or crashes) by `/hv/VMs/vm1/shutdown_svc.sh`.

You may adjust the files according to your needs.

13 Graphics (Desktop) configuration

What graphical output the user will see can be configured according to the respective needs.

While during development showing the hypervisor host desktop (the Linux desktop) is very convenient this typically is not wanted when the system is shipped to customers.

13.1 Display both, Host and Guest Desktop

To automatically start the guest, see the previous chapter. In such case you may want the guest desktop to be shown automatically (after login into the hypervisor host). Run the following script to show the guest desktop automatically *after* logging in into the hypervisor host:

```
$ cd /hv/home/.config/autostart
$ ./install_autostart.sh
```

Caution: Do **not** run `install_autostart.sh` with `sudo`!

You may want to also enable *autologin* on the host:

```
$ sudo -e gedit
→ /usr/share/lightdm/lightdm.conf.d/60-lightdm-gtk-greeter.conf
```

```
# autologin-settings to be appended
autologin-user=insert_your_username_here
autologin-user-timeout=0
```

Hint: **Automatic desktop startup** after login is done by `~/.config/autostart/autostart_svc.sh` created by `install_autostart.sh` which also adapts the *sudoers list* automatically.

Disabling desktop autostart can be achieved by removing the files created at `~/.config/autostart`:

```
$ rm ~/.config/autostart/autostart*
```

You may adjust the files according to your needs.

13.2 Display Guest Desktop only

When shipping the system to customers you may want to only display the guest desktop in full screen (= kiosk) mode. If kiosk mode is enabled you run the Windows guest in full screen mode without the option to switch back to the hypervisor host.

This will enable the kiosk mode after the next start of the guest VM:

```
$ gedit /hv/VMs/vm1/vmconfig.sh
export kiosk_mode=1
```

When using the kiosk mode with the previously shown automatic VM start and auto-login the hypervisor desktop is no longer required and should be disabled. In this case the guest is also started faster after power on.

Execute the following command to configure the host into a console-only mode.

```
$ sudo systemctl set-default multi-user.target
```

Without showing desktop you need to enable the `hv_vm1_start_nodsktp.service` to start the guest.

```
$ sudo systemctl enable /hv/services/hv_vm1_start_nodsktp.service
```

After rebooting `$ sudo reboot`, the guest should be started automatically in kiosk mode without showing the host desktop.

Hint: If you want to **reactivate the host desktop**, run

```
$ sudo systemctl disable hv_vm1_start_nodsktp
$ sudo systemctl set-default graphical.target
$ sudo reboot
```

If you want to **temporary start the host desktop**, run

```
$ startx
```

It is even possible to completely **uninstall the host GUI** to save disk space.

Please note, if the GUI is uninstalled, the interaction with the host can be performed only via a SSH connection.

As an alternative, you may configure a separate Grub Menu Entry or Press Ctrl-Alt-F2 to switch to a tty2 console

The following command will completely uninstall the GUI component:

```
$ sudo apt-get purge lightdm
```

Information about the Hypervisor **system services components**, which are automatically started.

```

hv_part.service           - assigns PCI Ethernet Cards to a RTOS (see
  → Hypervisor Quickstart Guide)
hv_rtos_start.service     - starts a RTOS guest (see Hypervisor
  → Quickstart Guide)
hv_vm1_start.service      - starts a Windows/Ubuntu guest with desktop
  → (see Windows Guest Guide)
hv_vm1_start_nodsktp.service - starts a Windows/Ubuntu guest with desktop
  → (see Windows Guest Guide)

```

Each service depends on previous ones and waits for it before its start.

Hint: It is also possible to completely hide all logging messages during the computer boot. Open `/boot/grub/grub.cfg` and find your boot configuration used for the deployment mode. It begins with “`menuentry ..`” text. Find a linux kernel boot line looking like “`linux /boot/vmlinuz-5.5.0-41-generic` “. Add parameters `.. code-block:: bash`

```
“quick splash console=ttyS0”.
```

Save file and reboot.

Hint: It is also possible to completely hide login/password prompt, that is displayed before a VM Guest window is shown in the kiosk mode. Execute the following command: .. code-block:: bash

“systemctl disable `getty@tty1.service`” and reboot.

You should see no login prompt anymore. The screen should remain black and then VM Desktop should be displayed. If you wish to log in, press a hot key `Ctrl-Shift-F2` .. `F6`.

13.3 Guest Desktop via Pass-Through

Using graphics pass-through mode the integrated graphics hardware is assigned to the Windows guest and no longer available for the host. This described in the [Passthrough Guide](#)

14 Boot Customization

14.1 Splash Screen

To disable the “Xubuntu Splash Screen” when booting, please edit `/etc/default/grub` and update the bootloader.

```
$ cd /etc/default
$ sudo gedit grub
```

```
remove "splash quiet" from GRUB_CMDLINE_LINUX_DEFAULT
```

and then run

```
$ sudo update-grub
```

15 Clone an existing VM

To duplicate an existing VM, you should execute the following steps. First, copy the template folder to a new VM specific folder, we will use the folder name `vm2` in this example.

```
$ cd /hv/VMs
$ cp -R vm_template vm2
```

Then you need to copy several files from the VM to be cloned and overwrite the one from the template. In this example we assume to clone the VM stored in folder `vm1`:

```
$ cd /hv/VMs/vm2
$ cp ../vm1/vmconfig.sh .
$ cp ../vm1/OVMF*.fd .
```

The following files need to be copied in case automatic network configuration is used for the external network (which is the default case). It assures, the network settings are identical for both VMs. If you do not copy these files, the VM may be not connected to the external network.

```
$ cp ../vm1/netif_conf.sh .
$ cp ../vm1/dns_conf.sh .
$ cp ../vm1/br_conf.sh .
```

Hint: If automatic network configuration is enabled, the VM started first will set up the overall networking. When a VM is started using the `vmrun.sh` script, this script is not aware that another VM may already run. Thus, the startup of the second VM will also try to initialize the overall networking. This will fail and an error message is generated. After a timeout of 5 seconds, the settings stored in the files `netif_conf.sh` and `br_conf.sh` will be used. This timeout can be eliminated by adjusting the file `/hv/VMs/config/vmconfigbase.sh`

```
$ cd /hv/VMs/config
$ gedit vmconfigbase.sh
```

```
# set the timeout to 0
echo "use automatic settings for external network"
# 1.) determine default gateway
#####
((netif_auto_timeout=5))
```

Caution: If automatic network configuration is enabled the IP address which the hypervisor host will get will most likely change from time to time. The same is true for the DNS server IP address. This would lead to networking issues because only the VM which initializes the network (the one which is started first) will run properly. To avoid this, the `br_config.sh` and `dns_config.sh` scripts of the first VM which is started need to be copied to the location where the second VM is running. For example, if the VM started first is located in `/hv/VMs/vm1` and the second one is located in `/hv/VMs/vm2` the following highlighted command lines be added into the `vmrun.sh` script:

```
$ cd /hv/VMs/vm2
$ gedit vmrun.sh
```

```

: : : :
: : : :
: : : :
export bootvm=1
source $vmidir/vmconfig.sh
# copy the bridge and DNS configuration script
cp $vmsdir/vm1/br_conf.sh .
cp $vmsdir/vm1/dns_conf.sh .
$vmsdir/vmctl/vms.sh&
: : : :
: : : :

```

We need to copy the VM disk image and rename it to vm2 (if you chose another name for the VM folder, please use this name for the disk image name!).

```

$ cd /hv/VMs/vm2
$ cp ../vm1/vm1.qcow2 vm2.qcow2

```

The VM configuration file needs to be adjusted. The name of the VM needs to be the same as the VM folder. The VM id (vmid) needs to be adjusted, it has to be unique over all VMs. You may have to adjust the number of CPUs to be used (num_cpus) and the RAM size (ramsize) of the new VM. Please also consider to adjust the number of CPUs and RAM size used for other VMs. The total number of CPUs should not be higher than the overall number of CPUs for all non real-time guests. Also the cumulated RAM size should not be higher than the available RAM size for all non real-time guests (total RAM size reduced by the size for the Real-time OS partition).

```

$ gedit /hv/VMs/vm2/vmconfig.sh

```

```

# Adapt following lines to your system and needs:
vmname=vm2
vmid=2
num_cpus=...
ramsize=...

```

The second Windows VM has to be added to the configuration.

```

$ gedit /hv/config/windowsvm.config

```

```

# Add the following configuration block at the very end of the file:
[Rtos2]
    "Name"="WindowsVM"
    "Description"="General purpose operating system"
    "AttachOnly"=dword:1
    "MemorySize"=dword:FFFA11A

```

Furthermore the following .config file must be also adjusted (Rtos2 activated):

```

$ gedit /hv/config/memory.config

```

```

[Rtos2]
    "MemorySize"=dword:2000000 ; 2nd RTOS
[Rtos2-End]

```

In case the second OS to be started is configured to start a second instance of the RTOS, this has to be deactivated.

```

$ gedit /hv/config/cpu.config

```

```
# comment the following two lines:
;[Rtos2]
; "ProcessorMask"=dword:4 ; (bit mask, bit 0 = core 0, bit 1 =
↳core 1, ...)
```

and the following entry must be activated:

```
$ gedit /hv/config/membase.config
```

```
[Rtos2]
"MemoryStartAddress"=dword:6000000 ; 96Mb
```

In case of RT-Linux, if the RTOS configuration is prepared for 2 RTOS instances you need to remove the entry for the second instance.

```
$ gedit /hv/lx/hv.config
```

```
# comment the following line:
;#include "../config/hwdevbase_rtos2.config"
```

The same is required for other RTOS like VxWorks (/hv/vx/vxworks.config)

After starting the second Windows VM instance, some settings have to be adjusted. First, you need to change the OSID to which the Windows guest is attached from 1 to 2. Select the “Hypervisor Attach” Shortcut on the Desktop and change the command line to:

```
%RTE_ROOT%\RtosUpload.exe -attach -osid 2
```

Also the Virtual Network settings need to be adjusted. You need to change the IP address of the Virtual Network adapter.

- open the Network and Sharing Center
- Select “change adapter settings”
- Select the “Realtime OS Virtual Network Adapter”
- Open the Properties and set the IP Version 4 address to 192.168.157.4

You need to change the MAC address of the Virtual Network adapter.

- open the Device Manager
- Open the Properties of the “Realtime OS Virtual Network Adapter”
- Go to the Advanced Tab and set the MAC address to AA-BB-CC-DD-EE-03

Finally you should be able to use the Virtual Network connection to the RTOS as well as the RTOS-Library in both Windows VM instances.

Caution: You should start the second VM at least 1 second after the first one to assure the network configuration executed by the VM started first is already finished. Please also note, the autostart mechanism described in the previous chapters is not implemented for a second VM instance. You will have to duplicate and adjust the respective script and configuration files.