acontis technologies GmbH

SOFTWARE

SystemManager

System Manager Manual

Version 9.x

Edition: July 29, 2025

Contents

1 Introduction

The System Manager is a web based tool (a Web Application) for configuration and diagnosis of the acontis hypervisor products. To show the graphical user interface, a web browser is required. Hypervisor configuration can be executed on the Hypervisor Host or remote via an IP connection. In addition, it is possible to connect to the Hypervisor Host via an MQTT gateway over the cloud.

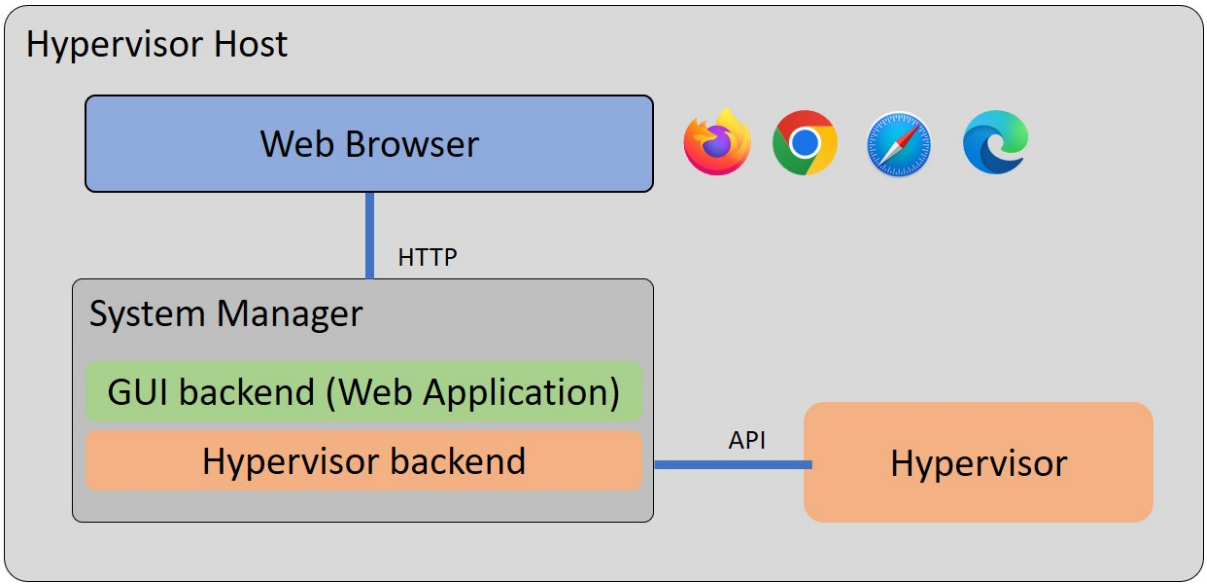
The System Manager consists of several components:

- Hypervisor backend: this is the interface to the Hypervisor Host. This backend interacts with the hypervisor (running on the same Hypervisor Host), e.g. to add guests, adjust configuration files etc. (see chapter *Hypervisor Backend*)
- GUI backend (Web Application): this is the interface to the web browser. The web browser always connects to the GUI backend.
- TCP/IP server: If the GUI backend and the Hypervisor backend run on different computers in the same network, they can be connected via TCP/IP. The Hypervisor Host and the GUI backends will then communicate over TCP/IP.
- MQTT gateway: If the GUI backend and the Hypervisor backend run on different computers in different networks or even at different physical locations, they can be connected via an MQTT broker.
- MQTT test broker: The test broker may be used in conjunction with the MQTT gateway if there is no MQTT broker available. This is NOT recommended and NOT supported for production environments!

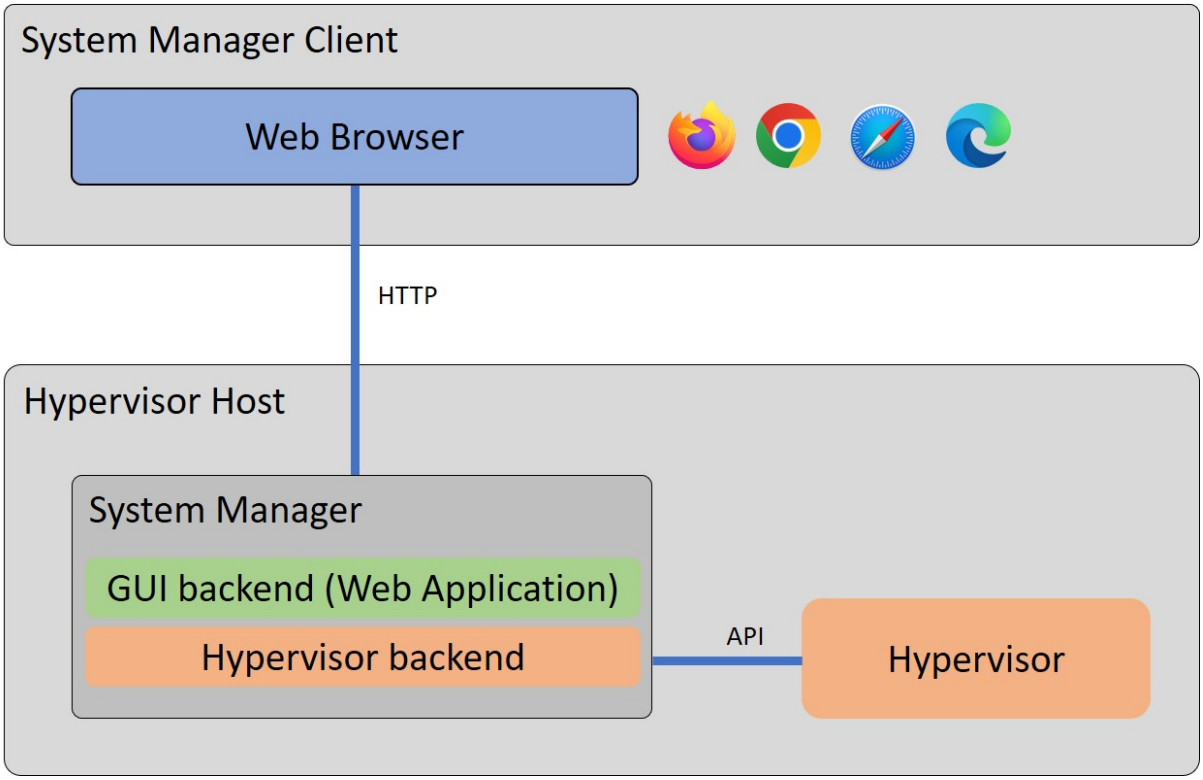
1.1 Architectures

The modular design of the software allows different system architectures.

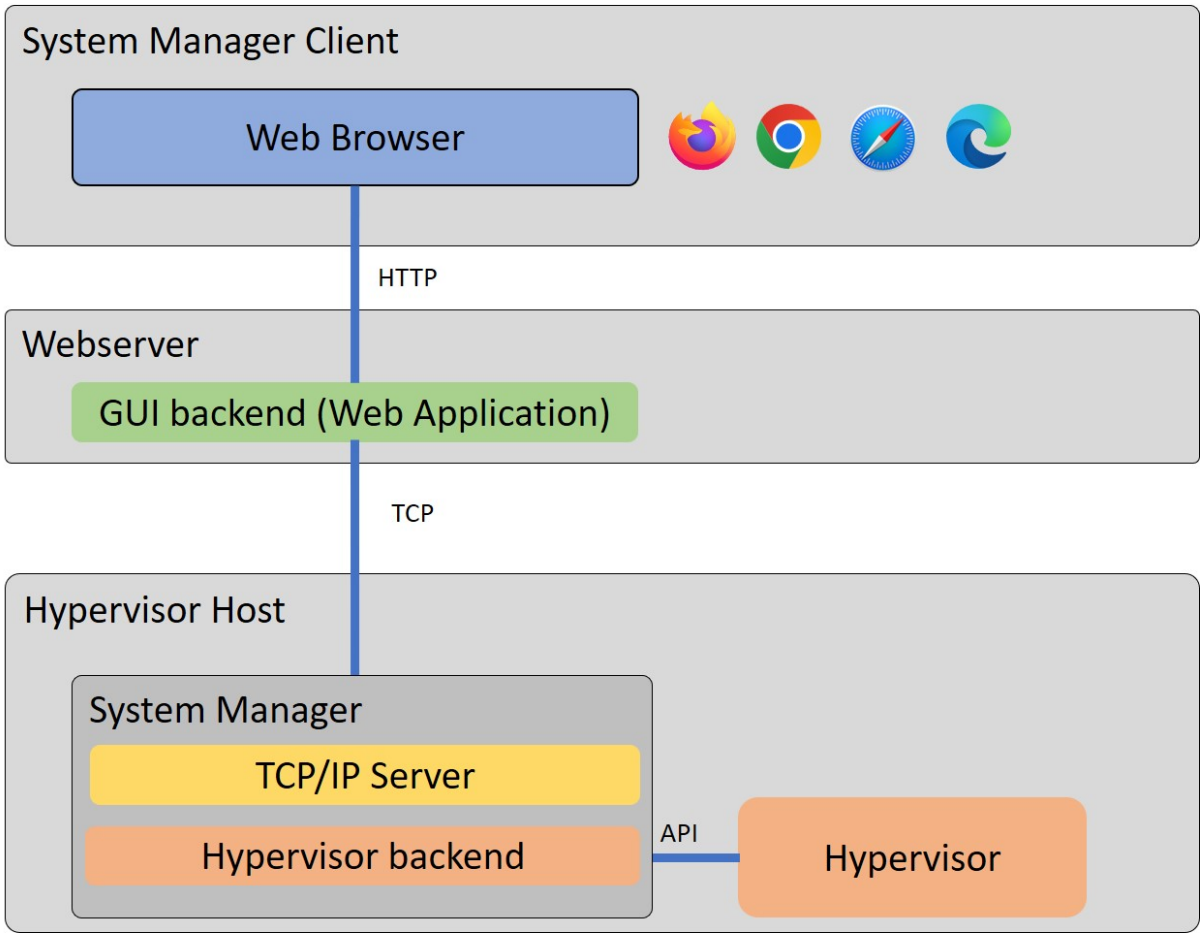
All components on the Hypervisor Host



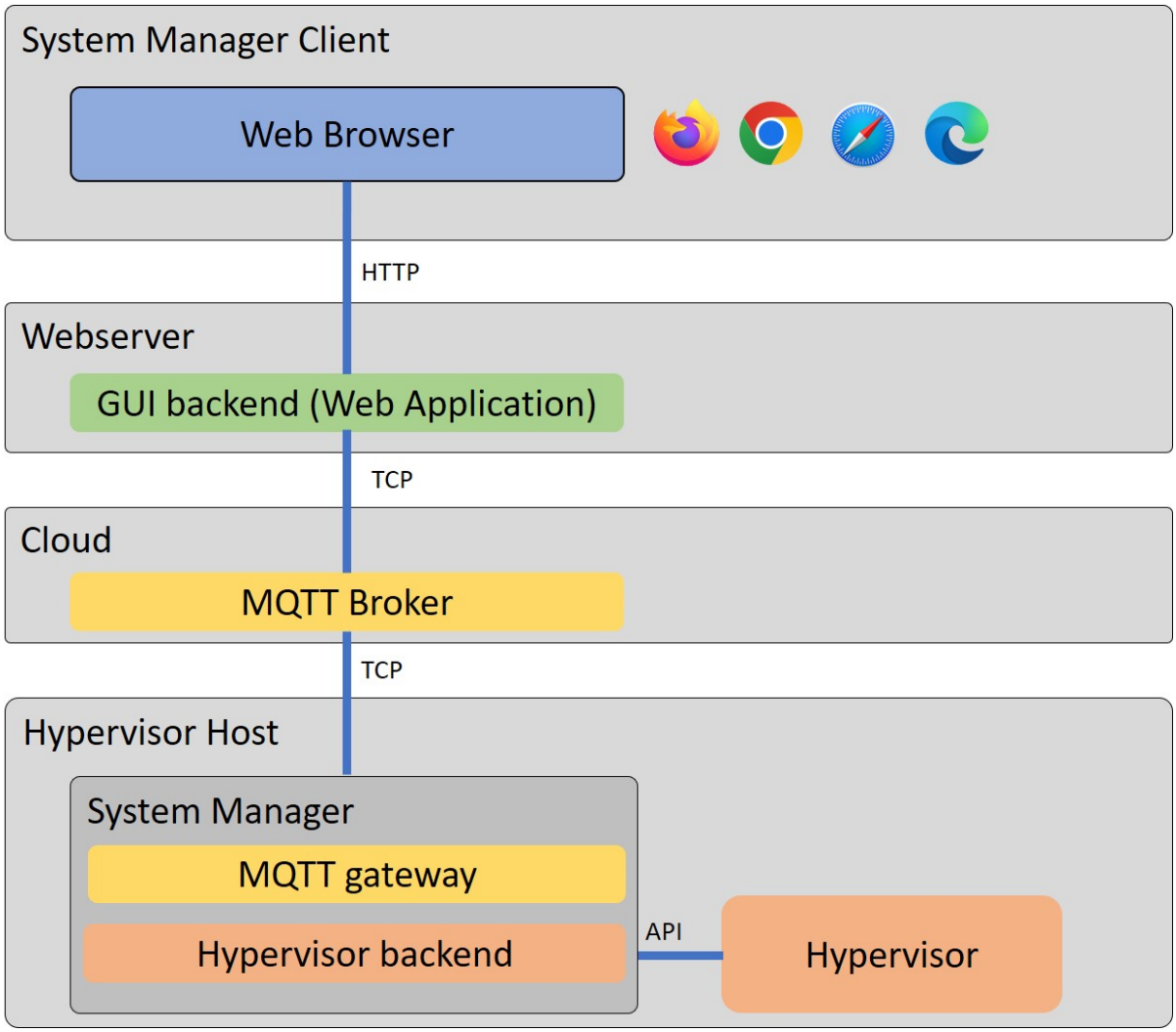
Using the web browser on a System Manager Client



With TCP/IP Server



MQTT Broker in the cloud



1.2 Installation and Update

There are various options, how to run the System Manager or System Manager components.

By default, the System Manager will be installed on the Hypervisor Host only and a browser is used to configure this (local) Hypervisor Host.

Another option is to run the GUI backend on a Windows machine, while the hypervisor backend runs on the Hypervisor Host (and both are connected via TCP or MQTT).

It is always required to install the System Manager on the Hypervisor Host.

If the GUI backend shall run on another computer, the System Manager must also be installed on this computer.

1.2.1 RTOSVisor - Default Installation

The RTOSVisor is shipped with a default installation of the System Manager, its location is in `/hv/sysmgr`. The System Manager can only be used after the basic RTOSVisor initialization was executed. How to initialize the RTOSVisor is described in the Hypervisor Quickstart Guide.

1.2.2 Windows Installation

If the System Manager (or the GUI backend) shall be executed on a Windows computer, you will have to manually install the System Manager on such a computer.

- you have to request the system manager deliverable from the acontis support
- copy the compressed system manager deliverable (e.g. `HvWebApp_Windows-x64_V1.0.16.zip`) into any folder where you want to use it later
- uncompress the deliverables and then remove the compressed file
- copy everything from _patches

```
md V1.0.16\HvWebApp_Windows
uncompress to V1.0.16\HvWebApp_Windows
xcopy /e V1.0.16\HvWebApp_Windows\Templates\_patches\*.* V1.0.16\.
call V1.0.16\HvWebApp_Windows\HvSystemManager.exe
```

1.2.3 Linux Installation

If the System Manager GUI backend shall be executed on a separate Linux computer (i.e. not the Hypervisor Host), you will have to manually install the System Manager on such a computer.

> **Caution:** On the Hypervisor Host you must install the System Manager in `/hv/sysmgr`. On other Linux machines you can store the System Manager in any folder where you want to use it.

- you have to request the system manager deliverable from the acontis support
- copy the compressed system manager deliverable (e.g. `HvWebApp_Linux-x64_V1.0.16.tar.gz`) into this folder
- uncompress the deliverables and then remove the compressed file
- copy patch files
- create doc directory

```
cd /hv/sysmgr
tar -xvzf HvWebApp_Linux-x64_V*.tar.gz
rm HvWebApp_Linux-x64_V*.tar.gz
cd ..
mkdir doc
```

1.2.4 RTOSVisor - System Manager Update

This section shows how to update to a later version of the System Manager for the RTOSVisor.

- Stop the System Manager

```
hv_sysmgr stop
```

- Preserve the old version and create a new folder in the Hypervisor Host

```
mv /hv/sysmgr /hv/sysmgr.orig
mkdir /hv/sysmgr
```

- Then follow the steps in *Linux Installation*
- Start the System Manager

```
hv_sysmgr start
```

1.3 System Manager control

Once the Hypervisor Host has been initialized, the System Manager backend will be executed automatically (as a service) when the computer is booted. This is controlled by the `/hv/services/hv_sysmgr.service` file.

Various commands are provided for System Manager control and diagnosis by calling the `hv_sysmgr` command.

You can execute this command to see a quick overview of the supported commands.

```
hv_sysmgr
System Manager running as a service. Output stored in /hv/sysmgr/sysmgr.log
System Manager Service running
Syntax /usr/bin/hv_sysmgr [status | debug=on | debug=off | log | clearlog␣
↪| start | restart | stop | enable | disable ]
```

- **status**: show current status.
- **log**: show log file.
- **clearlog**: clear log file content.
- **stop**: stop the System Manager.
- **start**: start the System Manager.
- **restart**: restart (stop, then start) the System Manager.
- **disable**: disable System Manager service (it will not be started automatically after reboot).
- **enable**: enable System Manager service (it will be started automatically after reboot).
- **debug=on**: enable additional debug messages.
- **debug=off**: disable additional debug messages.

1.4 Local operation

Local operation means the Hypervisor backend as well as the GUI backend run on the Hypervisor Host.

1.4.1 The System Manager backend

Caution: Prior to configuring the system using the System Manager you have to initialize the Hypervisor and reboot the system. See the Hypervisor Quickstart Guide for details.

The RTOSVisor will automatically start the System Manager backend.

Hint: If you want to directly start the System Manager backend in the RTOSVisor (without using the hv_sysmgr command), you need to stop the System Manager first.

```
hv_sysmgr stop
```

Then you can execute the following commands.

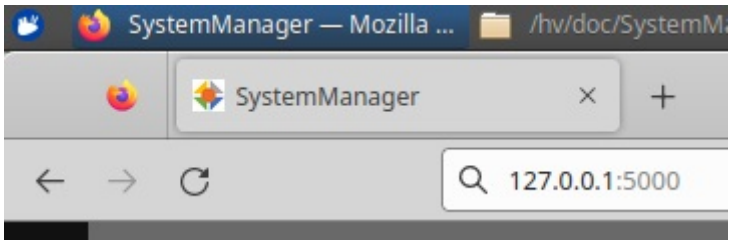
```
cd /hv/sysmgr
sudo ./HvSystemManager
```

Hint: You can type `./HvSystemManager /help` to get information about supported commandline arguments

---

1.4.2 Local Browser

The RTOSVisor is shipped with the Chromium browser to be able to locally configure the system. Open the browser and type in the localhost IP address *127.0.0.1* and connect to port 5000. The following URL can be used: `http://127.0.0.1:5000`



1.4.3 Remote Browser

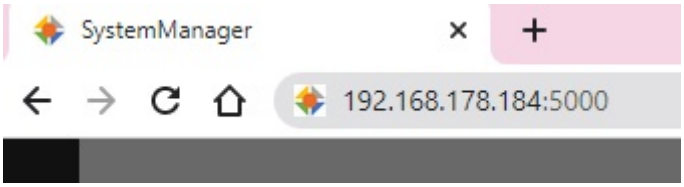
If you want to run the browser on another system (e.g. on your Windows or Linux workstation), you need to have a TCP/IP connection between the computer where the browser is running and the Hypervisor Host.

Open the browser and type in the IP address or the computer name of the Hypervisor Host and connect to port 5000. The computer name can be determined by calling the following command on the Hypervisor Host:

```
hostname
```

In case the hostname is `hypervisor-PC1`, the following URL can be used: `http://hypervisor-PC1:5000`

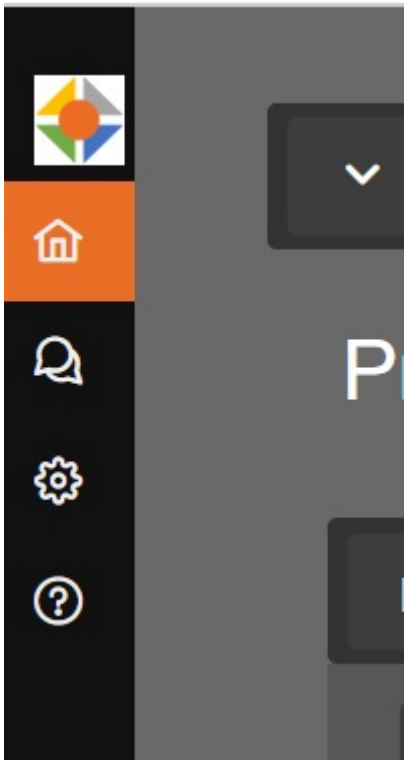
Alternatively, in case the IP address is `192.168.178.184`, the following URL can be used: `http://192.168.178.184:5000`



2 Overview

2.1 Basic GUI elements

On the top left part of the System Manager, one can select the basic GUI elements:



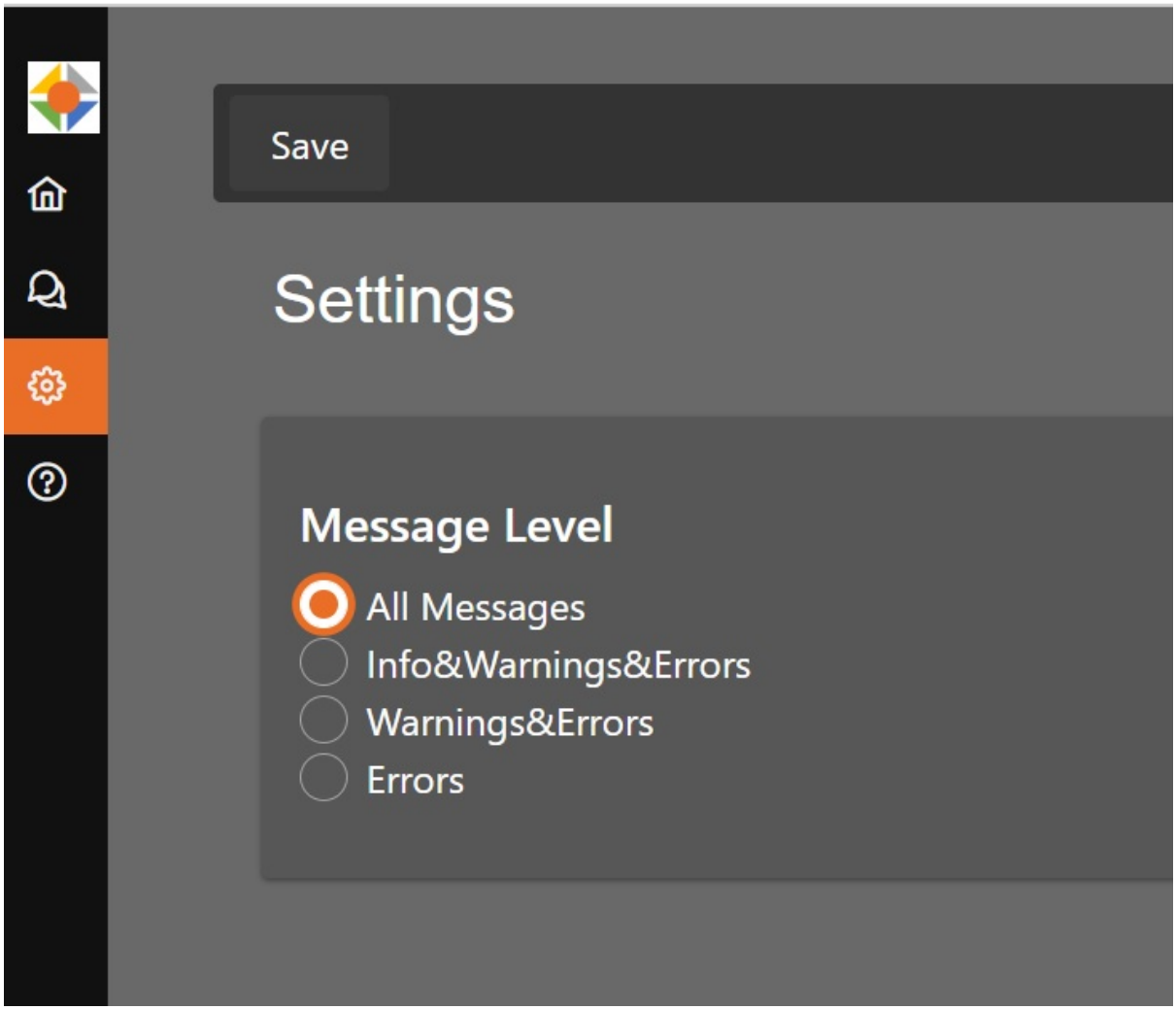
- Home: main user interface
- Messages: show all messages
- Settings: settings dialog
- About: System Manager information

On the top right part of the System Manager, you can switch between `Configuration` mode and `Run` mode.

In `Configuration` mode you will set up and configure general hypervisor settings and setup the guests. In `Run` mode you will be able to start and stop the guests.

2.2 Message Level

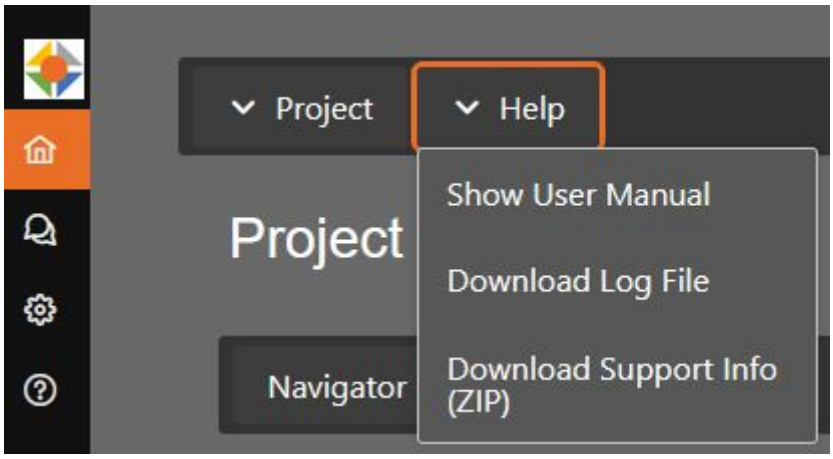
The System Manager provides messages for diagnosis purposes. The message level can be set in the Settings Dialog.



Caution: You need to select `Save` after changing the message level before you leave the dialog!

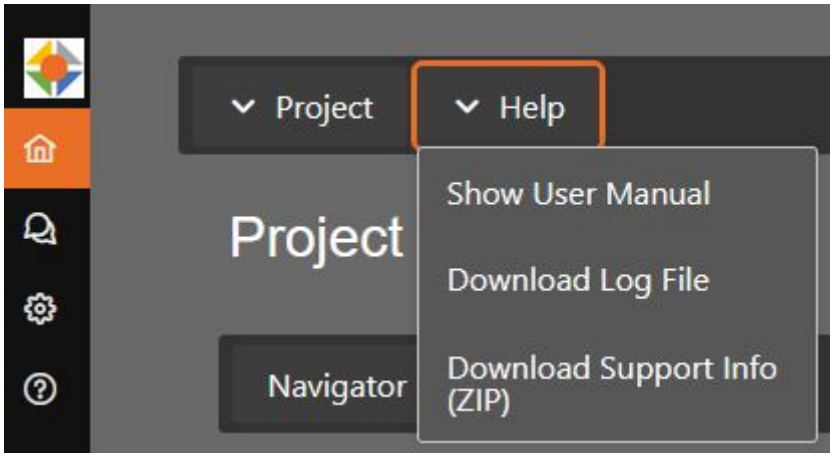
2.3 User Manuals

The user manuals are located on the Hypervisor Host. They are accessible via the Help menu.



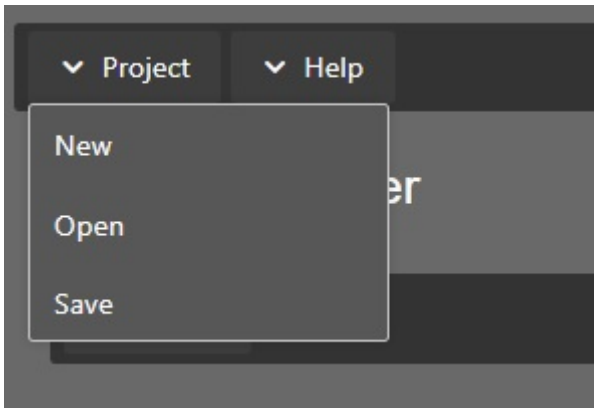
2.4 Logfile Download

In case of issues with the hypervisor configuration, you may download the logfile which contains all the messages. Please select *Download Log File* in the Help menu.

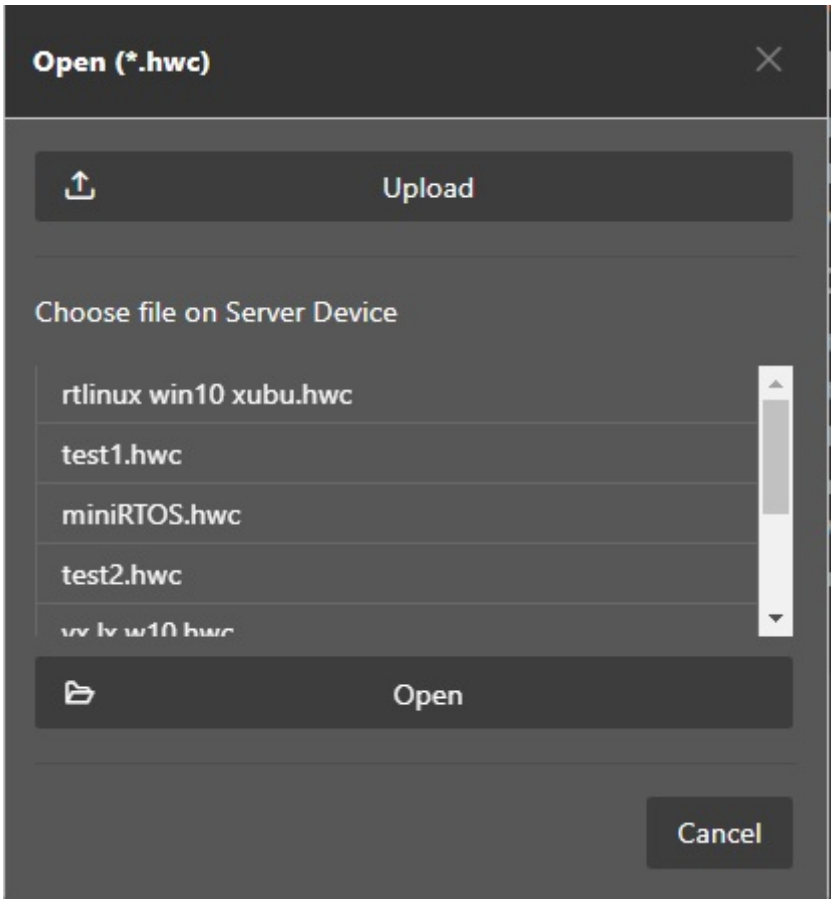


2.5 Projects

Specific hypervisor configurations (guests, partitioning, memory areas, …) can be stored for later use.

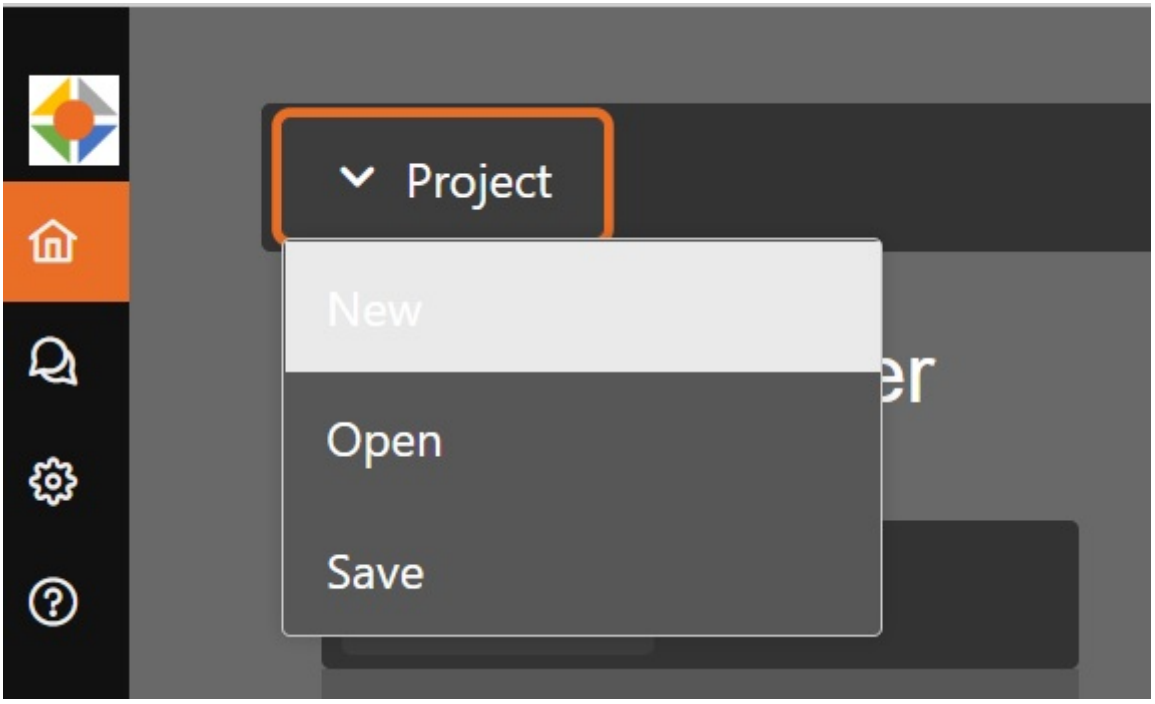


These files (with extension `hwc`) are stored in `/hv/sysmgr/WorkingDir`.



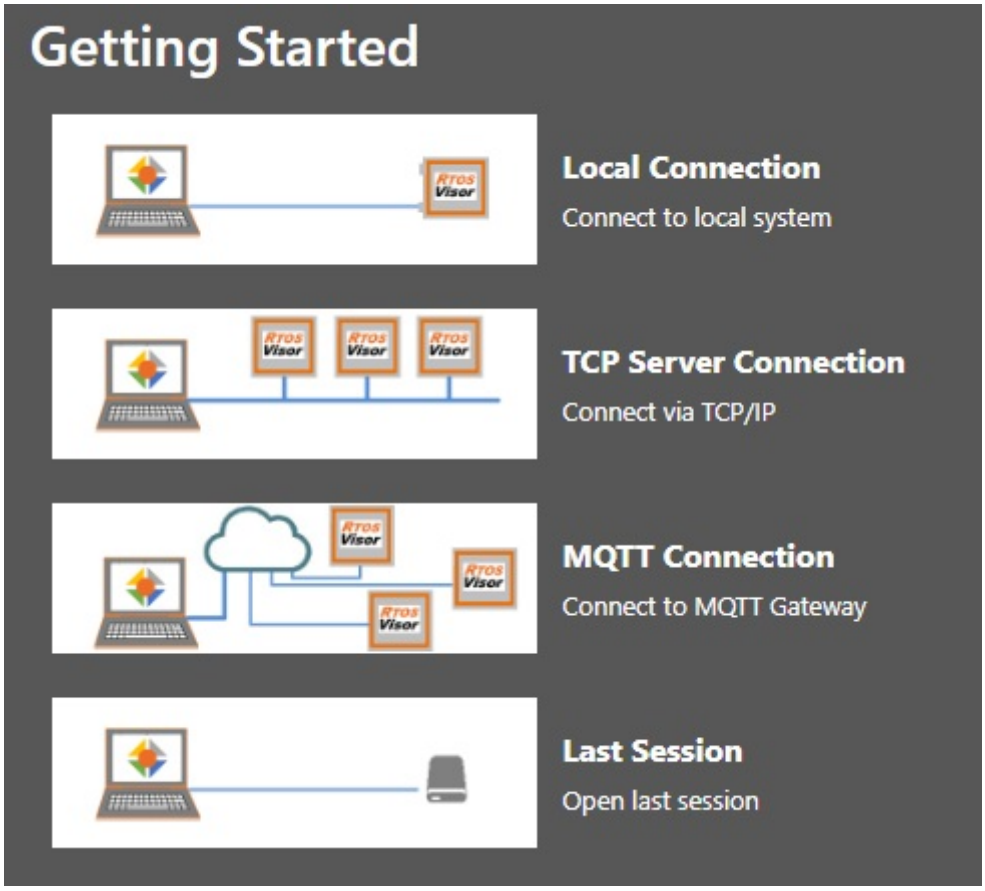
2.6 Welcome Page

After selecting `Project - New` the Welcome page will be shown where you can select how to connect to the Hypervisor Host.



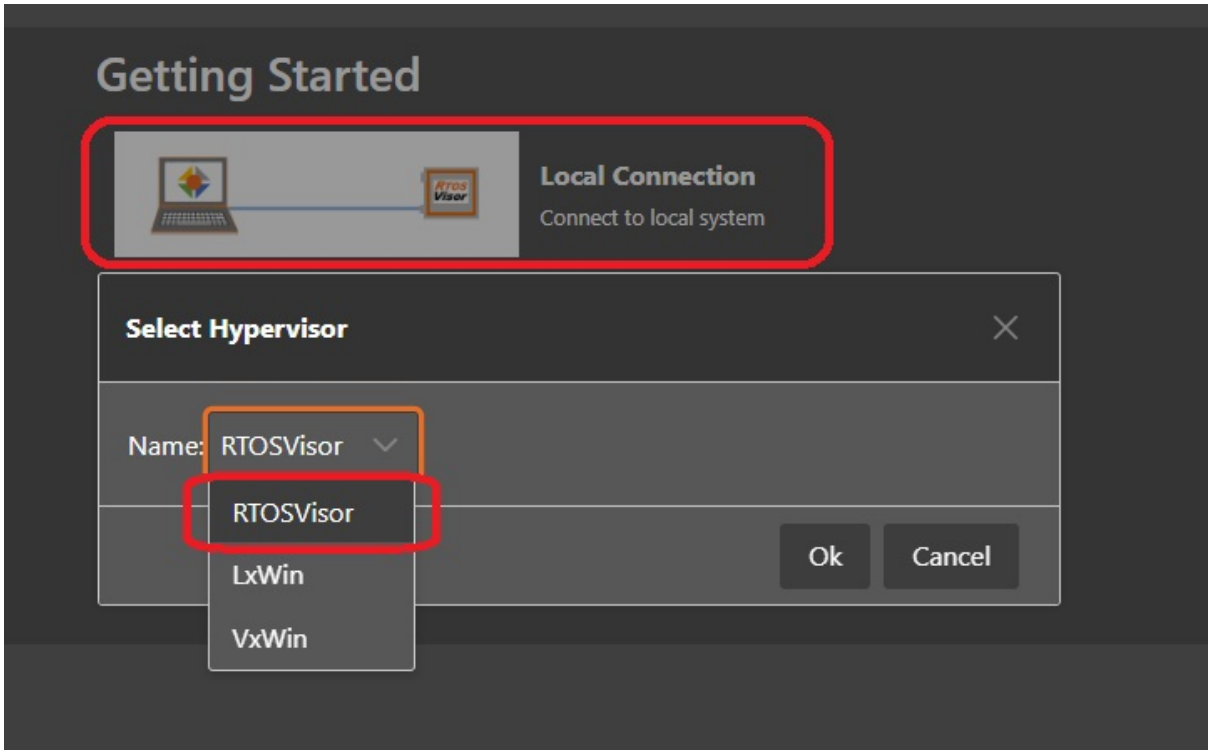
Three types of connection to one or multiple Hypervisor Hosts as well as the last configuration session can be selected:

- local connection to a single Hypervisor Host where the GUI backend (the Web Application) as well as the RTOSVisor backend are executed.
- connection via TCP to one or multiple Hypervisor Hosts in the same network.
- connection via an MQTT gateway to one or multiple Hypervisor Hosts in different networks. This method enables remote configuration and diagnosis over the Internet without the need for a VPN connection.
- the last configuration session.

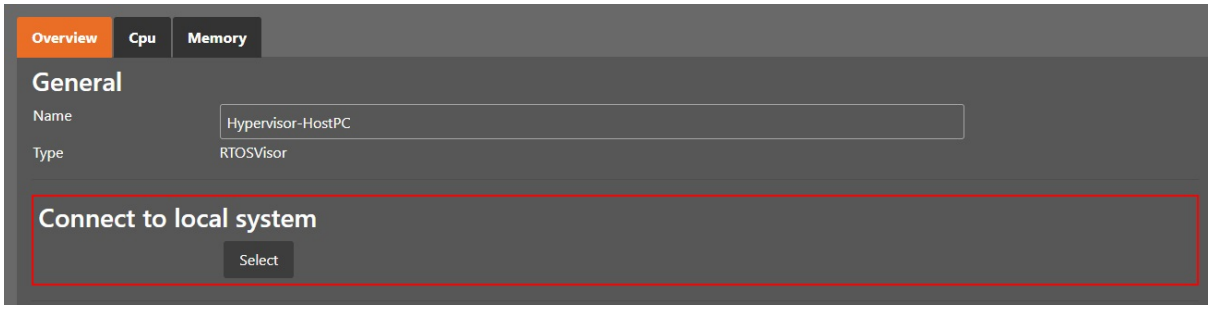


2.7 Local Connection

The most common connection to the Hypervisor Host is the local connection. The System Manager GUI backend runs on the Hypervisor Host. In a first step, you need to select `Local Connection` and the respective hypervisor type (`RTOSVisor`).



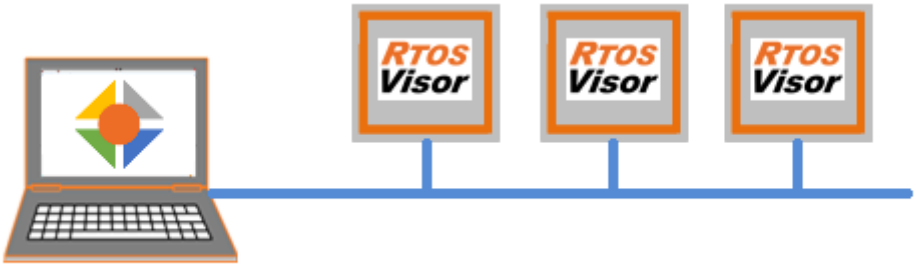
After acknowledging `Connect to local system`, the Hypervisor Host connection dialog will be shown. You should provide an appropriate name for this Hypervisor Host and press the `Select` button.



After selecting the local connection type you will be able to configure one single Hypervisor Host.

2.8 TCP Connection

If you want to configure one or multiple systems remotely via TCP/IP, the hypervisor backend needs to enable its internal TCP Server. The GUI backend (Web Application) may run on the same computer as the browser is executed. Optionally, you can run the GUI backend also on a different computer. The graphics below assumes the GUI backend as well as the browser run on the same computer.



In a first step, you need to start the hypervisor backend with the TCP server enabled. In the following example, the TCP port is explicitly set to `6001`.

```
cd /hv/sysmgr
sudo ./HvSystemManager /tcpserver=*:6001
```

If you simply use the option `/tcpserver`, the default port 6000 will be used.

Hint: By default, sysmgr is started at system boot. If you only want to use the TCP connection on this system, you can stop sysmgr using the command `hv_sysmgr stop`. You can find a description about this in *System Manager control*.

Then you need to start the `HvSystemManager` web application (e.g. on the computer where the browser is running).

- Windows (see *Windows Installation*)

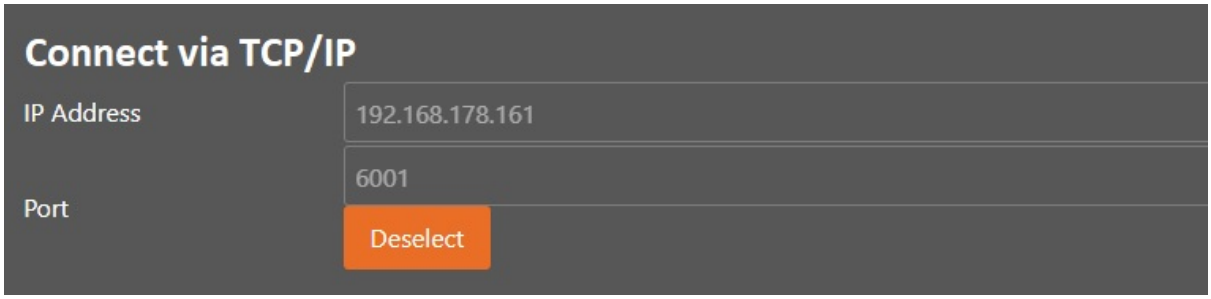
  Run the HvSystemManager.exe application without any parameter.

- Linux (see *Linux Installation*)

  Run the HvSystemManager application without any parameter.

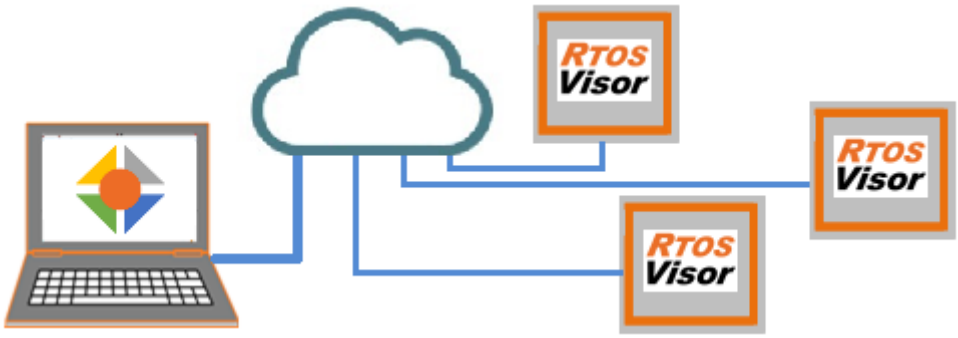
Then start the browser and use the IP address of the web application to show the GUI (use the IP address of the computer where the web application runs and connect with port 5000). If you run the browser and the web application on the same computer, just use the URL `http:localhost:5000` in the browser.

You need to select the TCP Connection option. Set the appropriate IP address of the hypervisor computer as well as the port and press the `Select` button.



2.9 MQTT Gateway Connection

If you want to configure the system remotely using MQTT, the hypervisor backend needs to connect and register with an MQTT broker. The GUI backend (Web Application) may run on the same computer as the browser is executed. Optionally, you can run the GUI backend also on a different computer. The graphics below assumes the GUI backend as well as the browser run on the same computer.



In a first step, you need to connect the hypervisor backend with the MQTT broker (on the hypervisor machine). In the following example, a public test broker at `test.mosquitto.org` is used. The hypervisor backend needs to be registered with a unique name, in our example it will be `TP117`. The default TCP port of MQTT is 1883.

```
cd /hv/sysmgr
sudo ./HvSystemManager /mqttserver=TP117:test.mosquitto.org:1883
```

Then you need to start the `HvSystemManager` web application (e.g. on the computer where the browser is running).

- Windows (see *Windows Installation*)

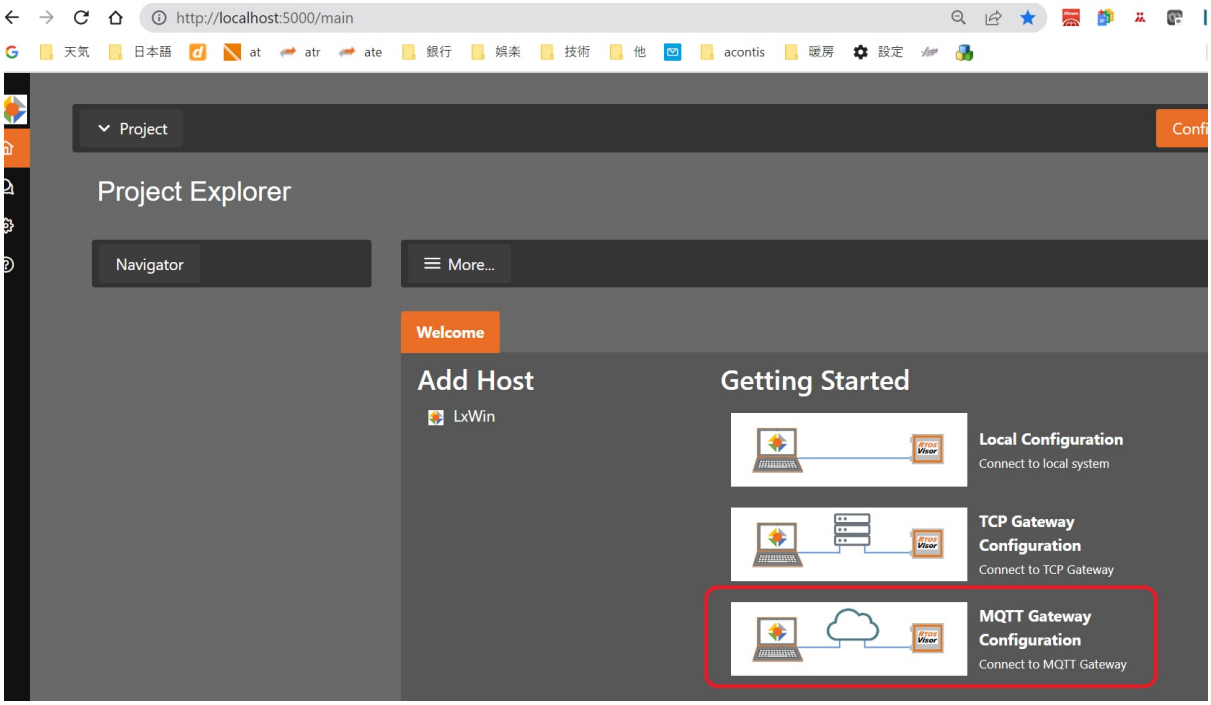
  Run the HvSystemManager.exe application without any parameter.

- Linux (see *Linux Installation*)

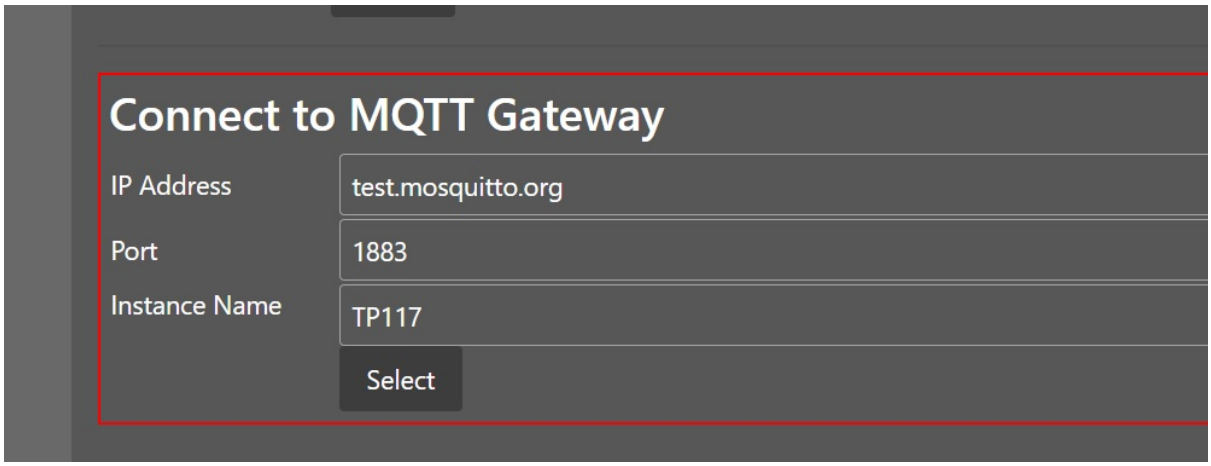
  Run the HvSystemManager application without any parameter.

Then start the browser and use the IP address of the web application to show the GUI (use the IP address of the computer where the web application runs and connect with port 5000). If you run the browser and the web application on the same computer, just use the URL `http:localhost:5000` in the browser.

Finally you need to select the MQTT Gateway Configuration option.



Set the appropriate broker (IP/URL) and hypervisor name (Instance name) and press the `Select` button.



2.10 Last Session

The last configured session will be loaded. This includes the respective connection (local, TCP or MQTT) and the related hypervisor configuration (guests, partitioning etc.).

The System Manager will store the last session in the same way as storing projects. The file `last_session.hwc` will contain the respective data.

3 Quickstart Tutorial

This section gives a quick introduction into the System Manager configuration tool.

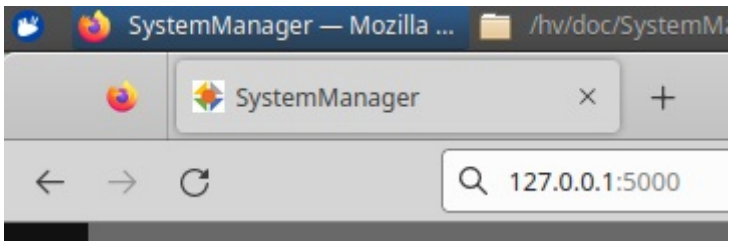
Hint: You should have read the Hypervisor Quickstart Guide as well as the Hypervisor Windows Guest Guide and the previous chapters in this manual before continuing here.

3.1 Browser installation

The System Manager is a web based tool. You can start the firefox browser on the Hypervisor Host:

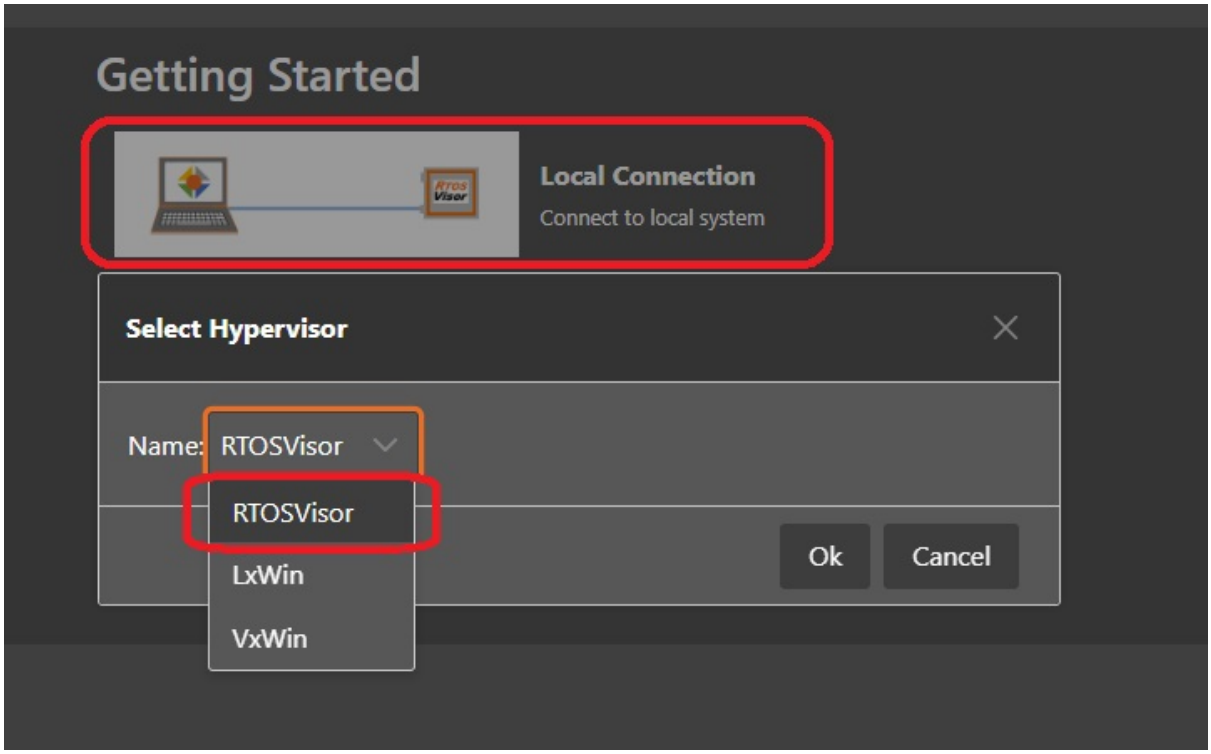
```
firefox&
```

Type in the localhost IP address *127.0.0.1* and connect to port 5000. The following URL can be used: `http://127.0.0.1:5000`

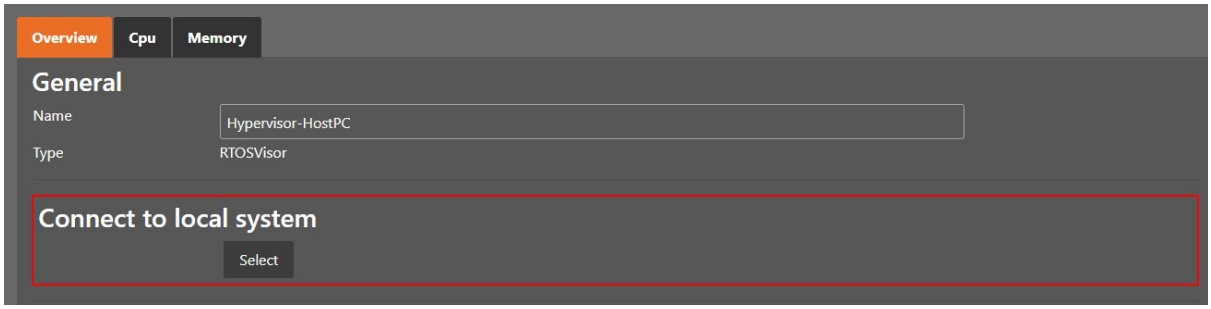


3.2 Local Connection

To connect the browser with the Hypervisor backend, please select `Local Connection` and the respective hypervisor type (`RTOSVisor`).



After acknowledging `Connect to local system`, the Hypervisor Host connection dialog will be shown. You should provide an appropriate name for this Hypervisor Host and press the `Select` button.



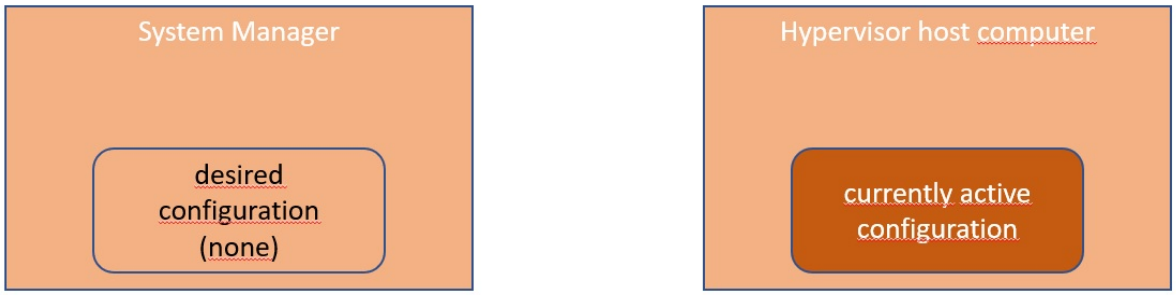
3.3 Synchronization

3.3.1 Paradigm

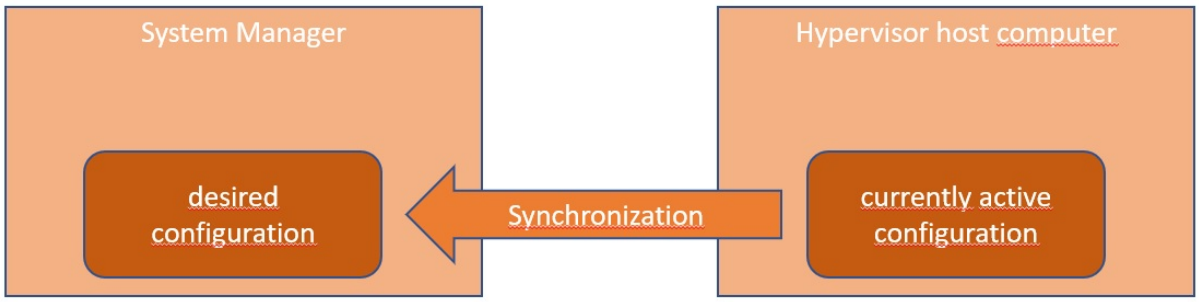
The configuration paradigm distinguishes between

1. The **currently active configuration** (guests, hardware partitioning, hardware information, …). The currently active configuration is stored on the Hypervisor computer.
2. The desired configuration is temporarily stored in the System Manager and can be changed by user interaction (e.g. add additional guests).
3. **Synchronization:** synchronize the **desired configuration** and the **active configuration**.

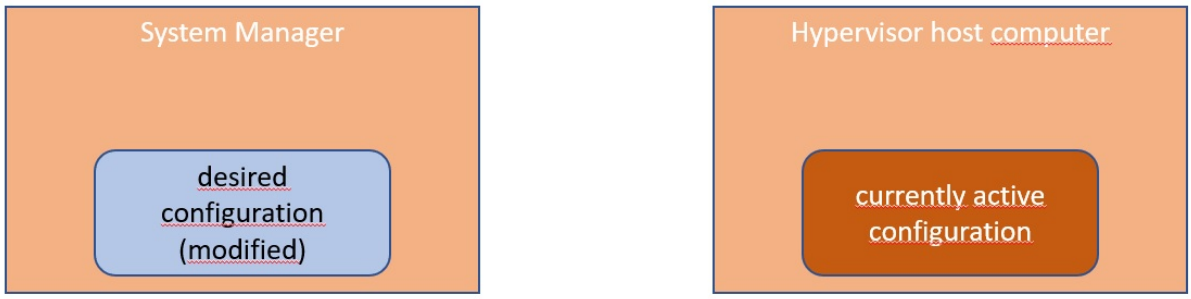
When the System Manager is started for the first time, no **desired configuration** exists.



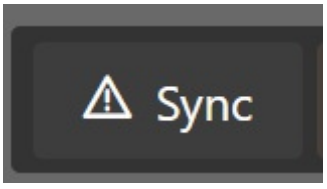
We will need to synchronize first, to load the **currently active configuration** stored in the Hypervisor computer into the System Manager. After this step, the **desired configuration** is identical to the **currently active configuration**.



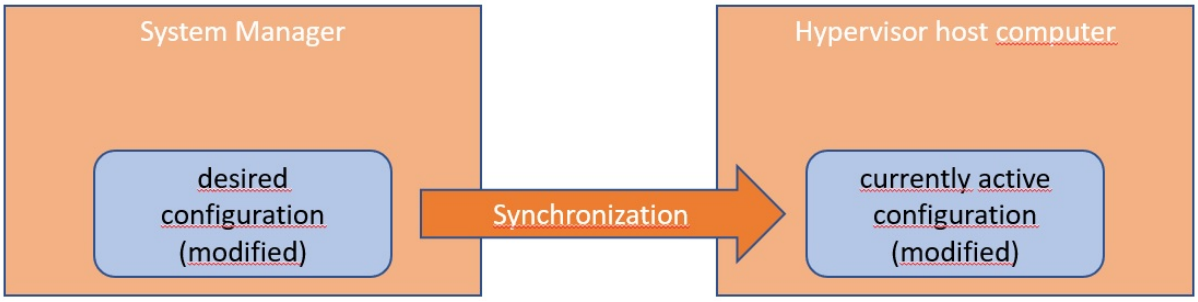
Then you will modify the configuration (e.g. adding guests). All these steps are done within the System Manager only, the currently active configuration will not be changed.



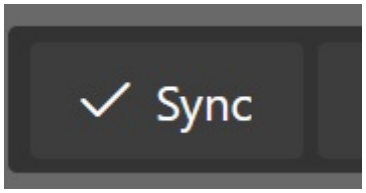
While you are working on the desired configuration, the Sync button will look as follows:



When you have finished changing the configuration, you must synchronize again to bring the new configuration into effect.



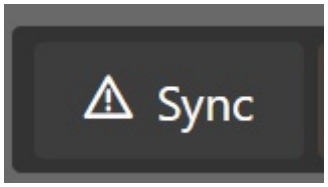
After successful synchronization, the Sync button will look as follows:



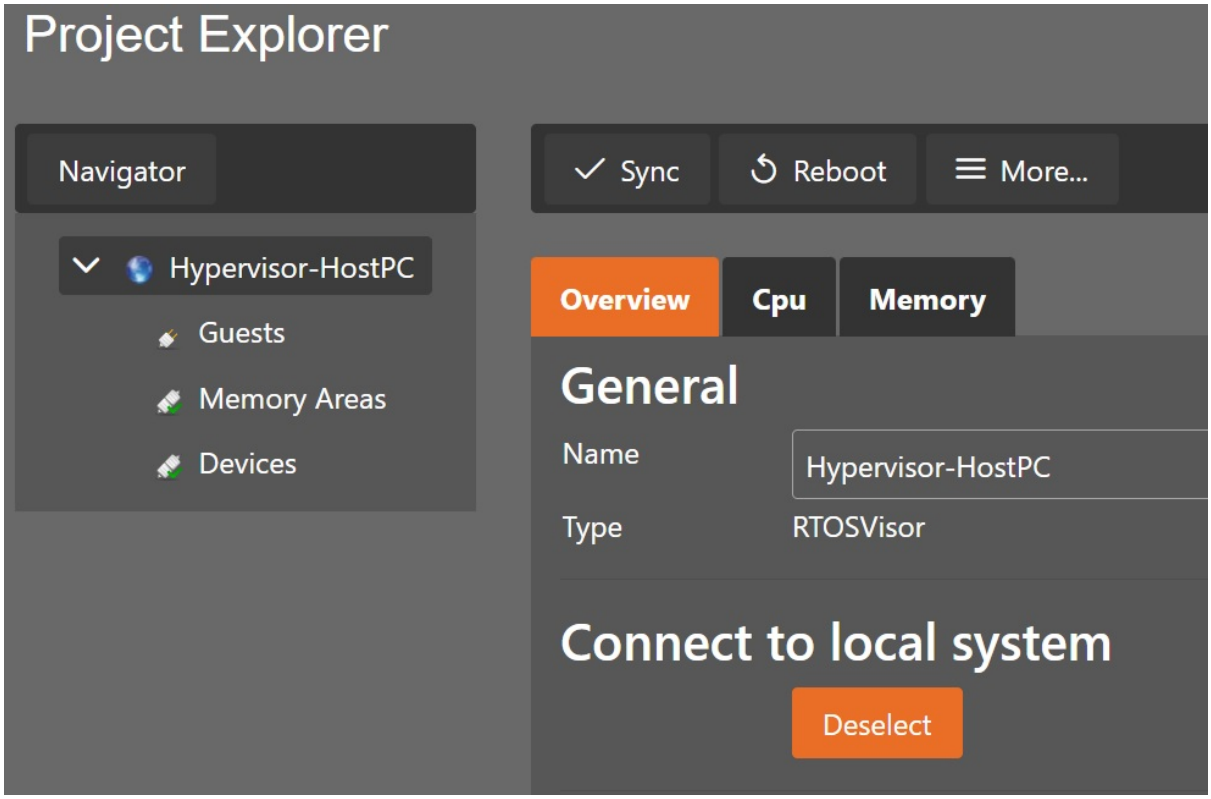
3.4 Initial Synchronization

When you have started the System Manager for the first time, you need to run an initial synchronization step.

Press the Synchronization button.

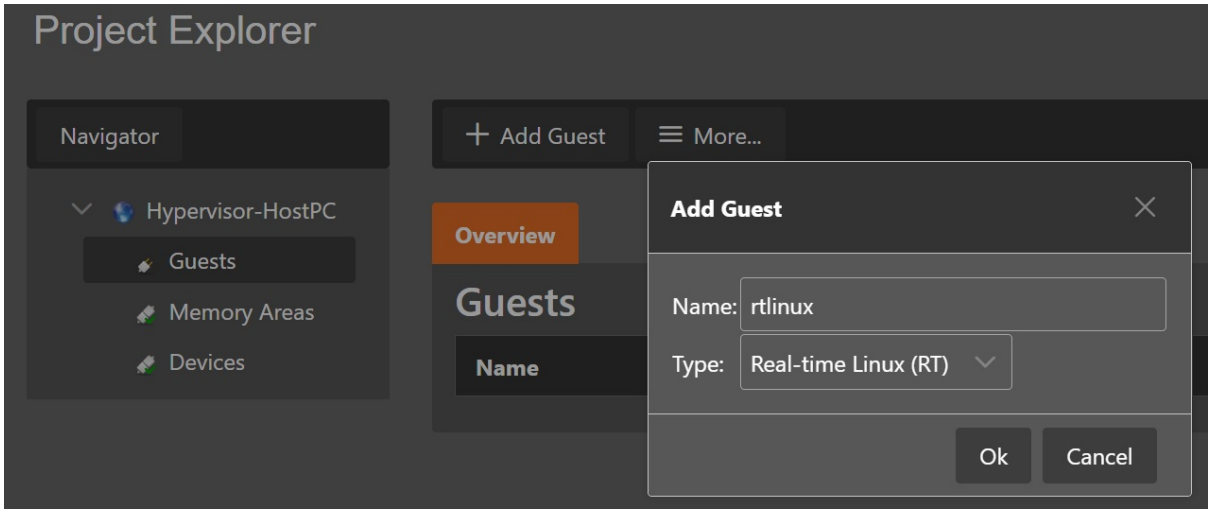


And acknowledge the following dialog (press the *Apply* button). After successfully synchronizing the configuration, the result should look similar to this:

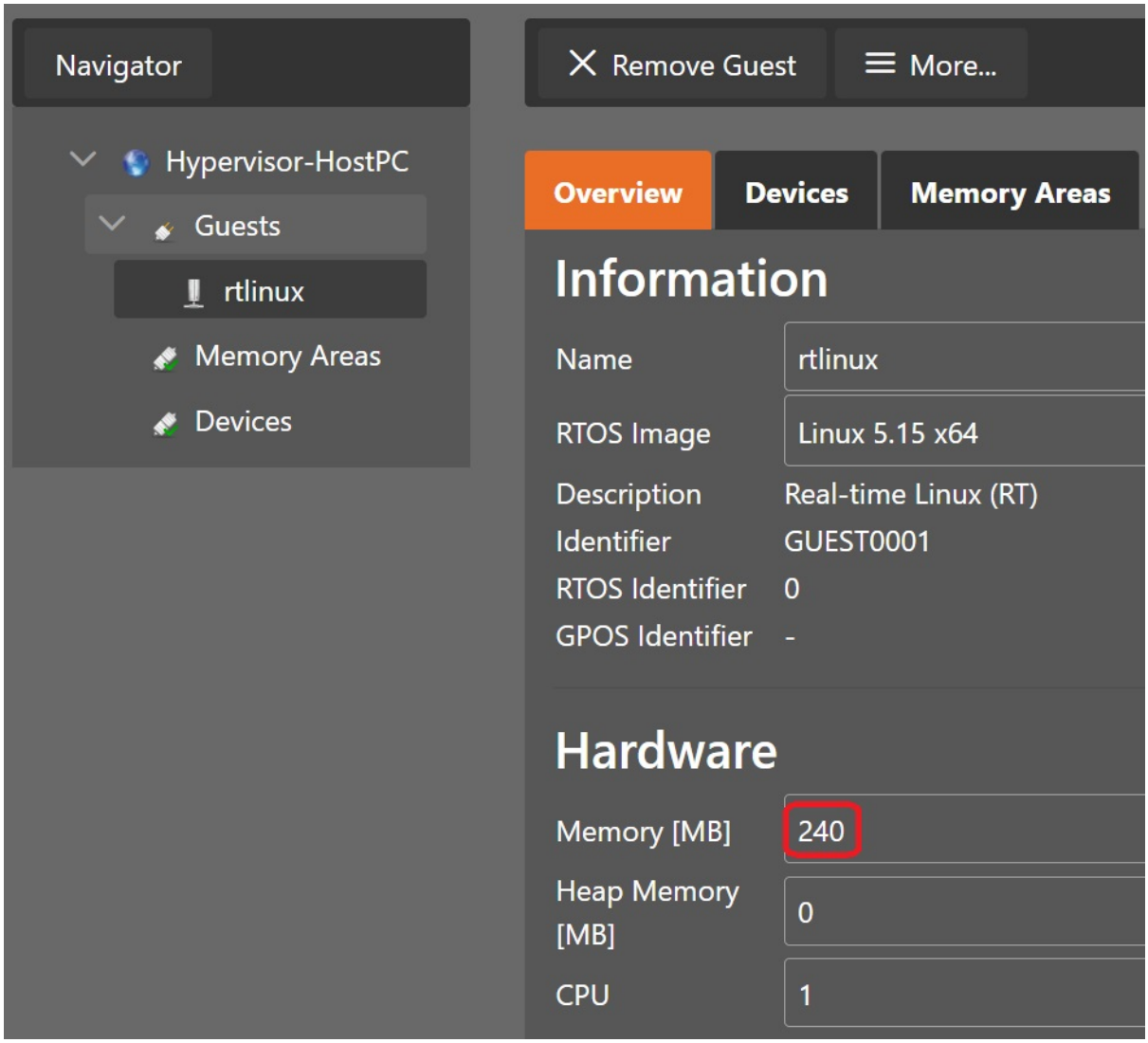


3.5 Add Real-time Linux guest

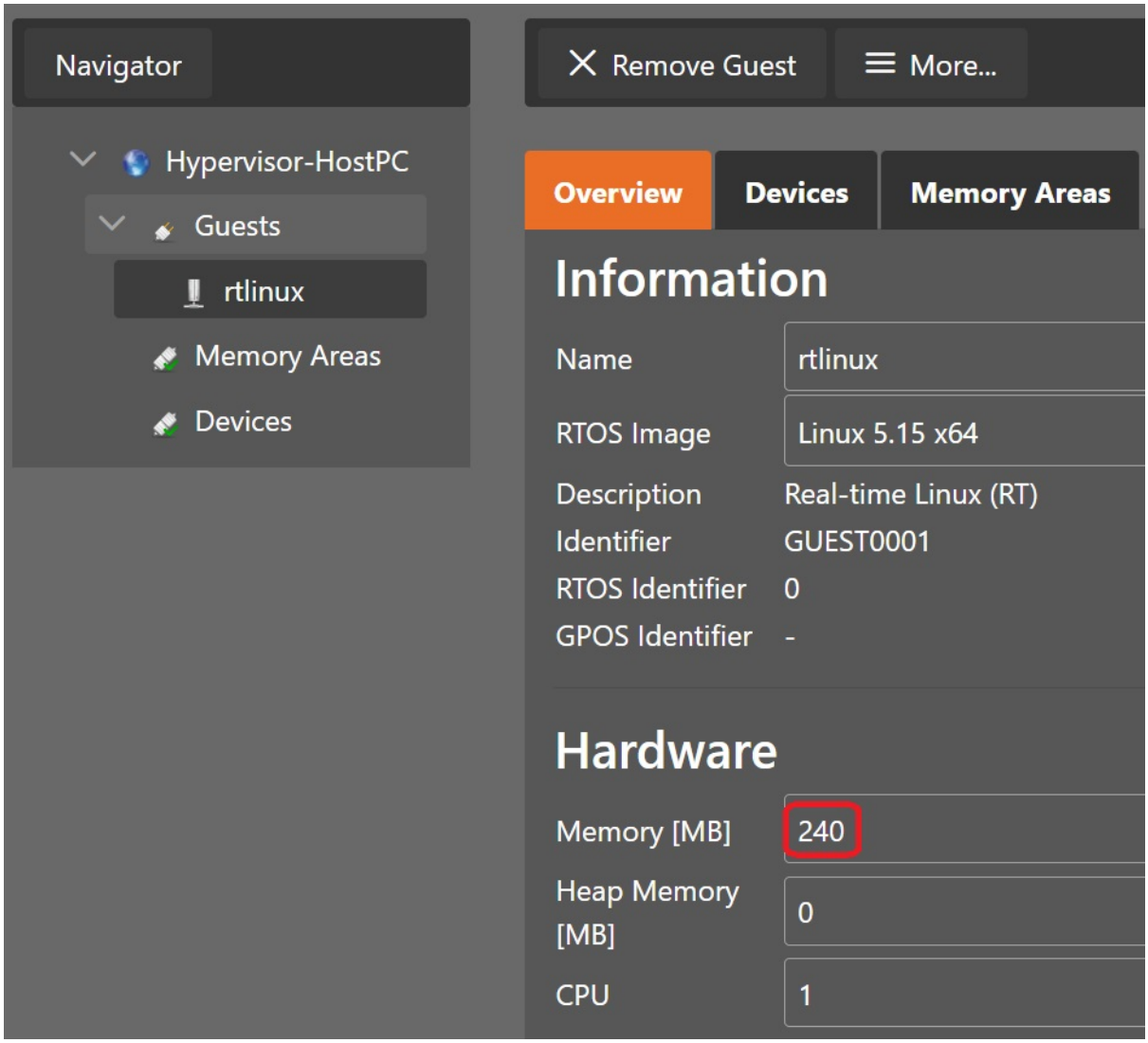
Select *Guests* in the tree view on the left side, then select *Add Guest*. In the *Add Guest* dialog, adjust the name of the guest to *rtlinux* and select *Real-time Linux (RT)* for the guest type.



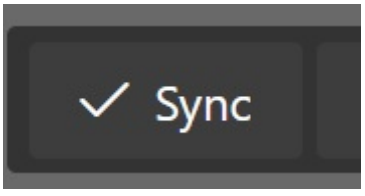
Select the *rtlinux* guest in the tree view and adjust the memory size to 240 MByte.



Finally, select the Hypervisor Host (e.g. *Hypervisor-HostPC*) and press the *Sync* button to write the modified synchronization to the Hypervisor Host. You will get an overview of the configuration changes made. Prior to applying these changes you may verify if the changes match your expectation.



After the new configuration was successfully written, the Sync button changes to the clean state:



3.6 Add Ethernet adapter to Real-time Linux guest

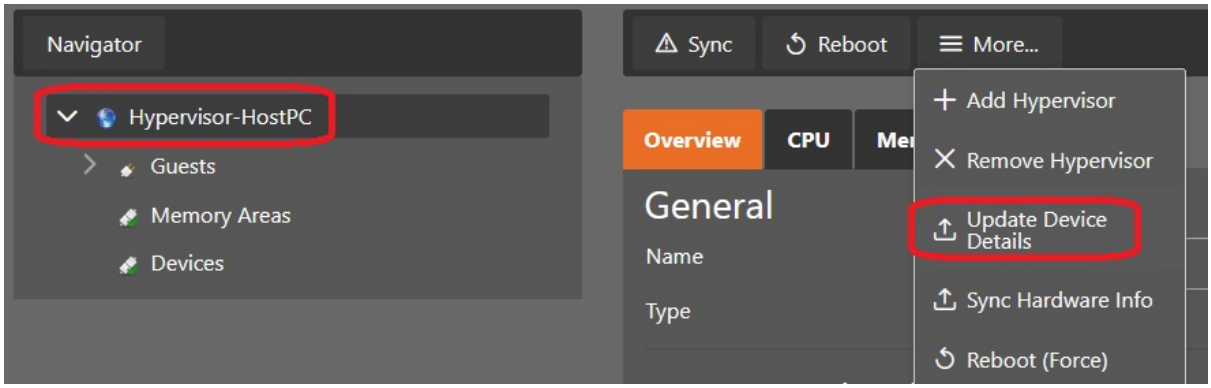
3.6.1 Determine the desired adapter

If your computer has multiple Ethernet adapters, you will have to determine the right one.

In a first step, please physically disconnect the Ethernet cable from the Ethernet device which shall be assigned to the RT-Linux guest.

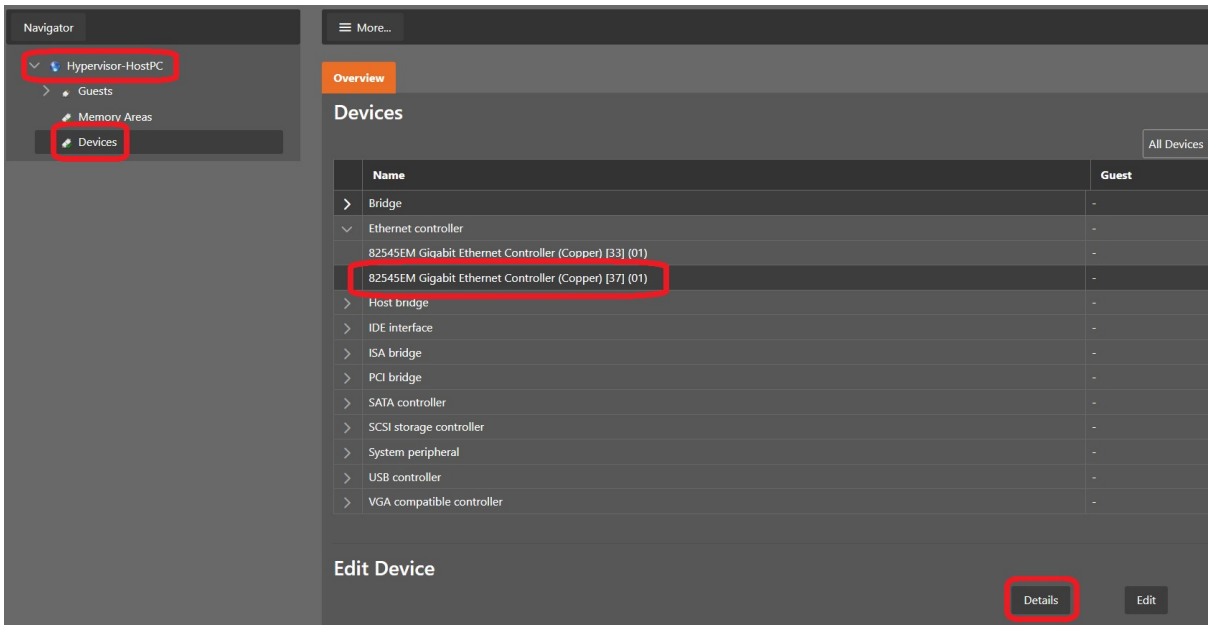
Now we need to update the device information (in case the cable had been connected before). Select the Hypervisor Host (e.g. *Hypervisor-HostPC*) and press the *Update Device Details* entry below the *More...*

combo-box.

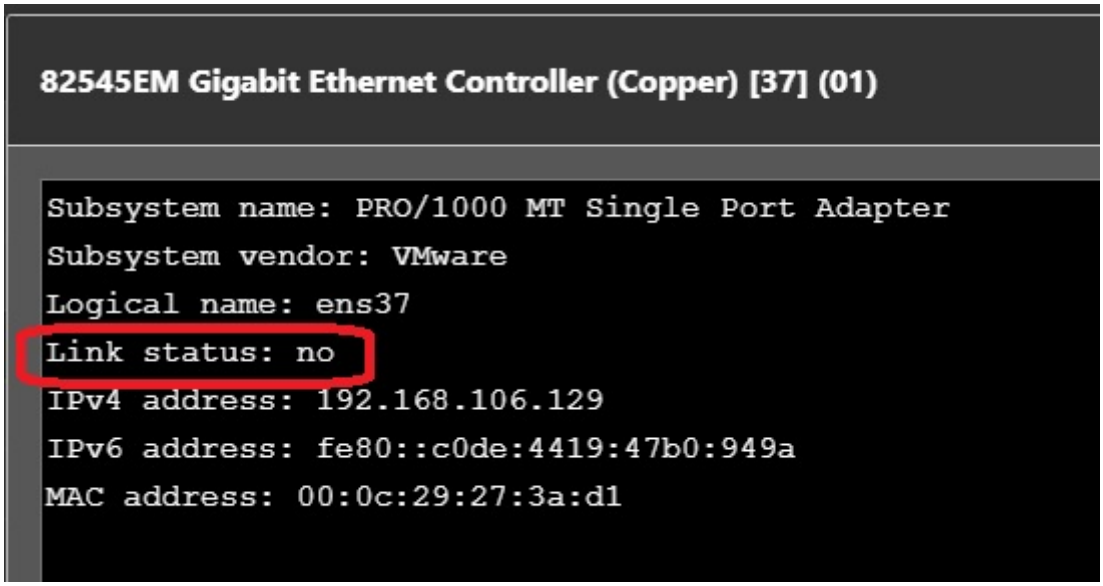


After updating the device information, select the *Devices* tab in the tree view on the left side (do **not** select the *Devices* tab in the rtlinux guest!).

Then unfold the *Ethernet controller* entries, select one of the Ethernet controllers and press the *Details* button.



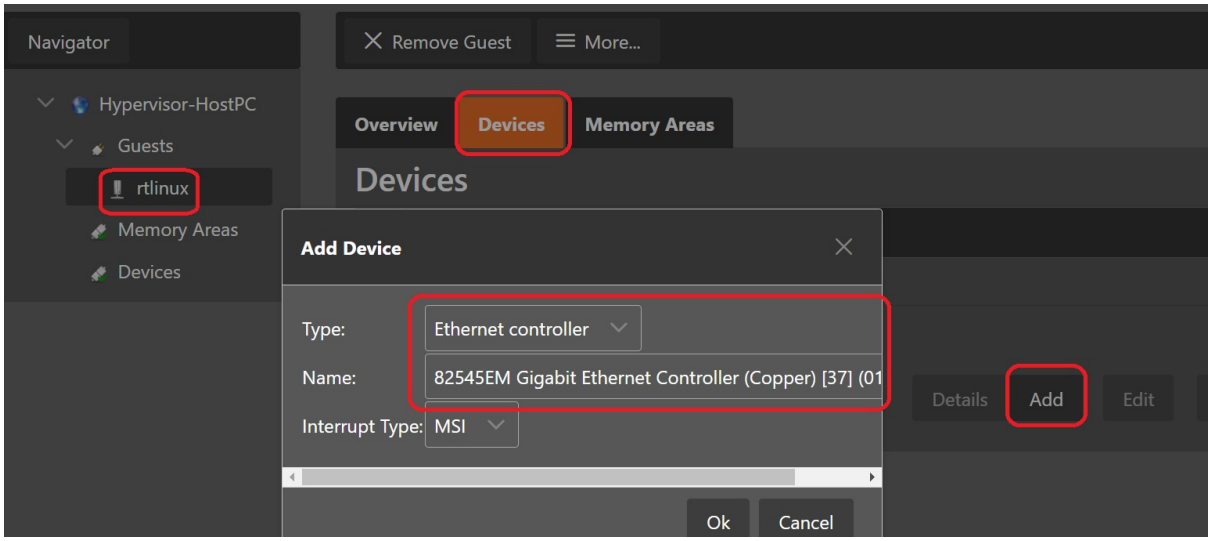
You will see various pieces of information about this Ethernet controller, one of which is the **Link status**. If it is set to **no**, no cable is connected.



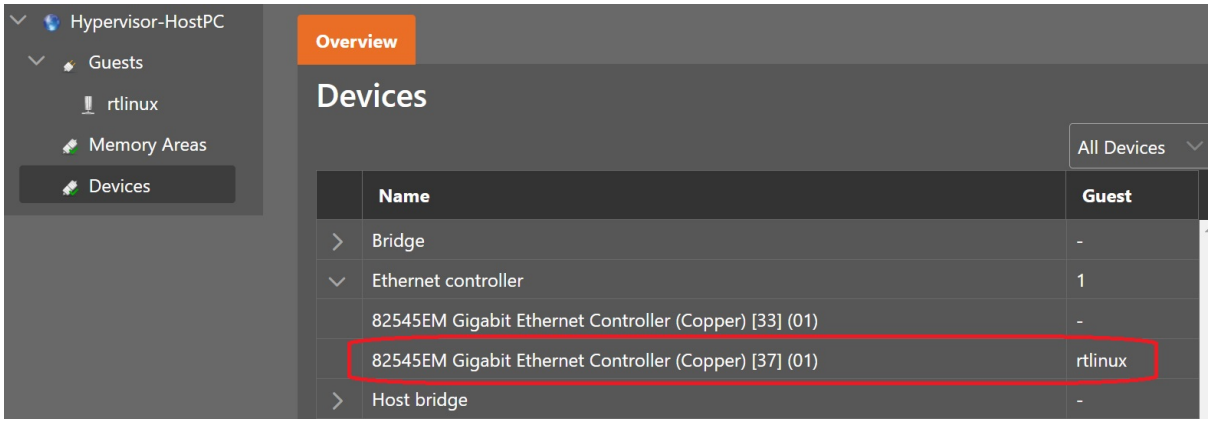
Then you should physically connect the Ethernet cable to the adapter and assure it is connected to a powered on switch. Update the device details again and check if the **Link status** now has changed to **yes**. If this is the case, you have correctly determined the desired adapter, remember its name for later.

3.6.2 Assign the desired adapter to the guest

Now you need to select the rtlinux guest again, switch to the *Devices* tab in the guest view, press the *Add* button and select the Ethernet adapter you want to assign to Real-time Linux.



Verify the result by selecting the Hypervisor Host (e.g. *Hypervisor-HostPC*) and switch to the *Devices* tab.



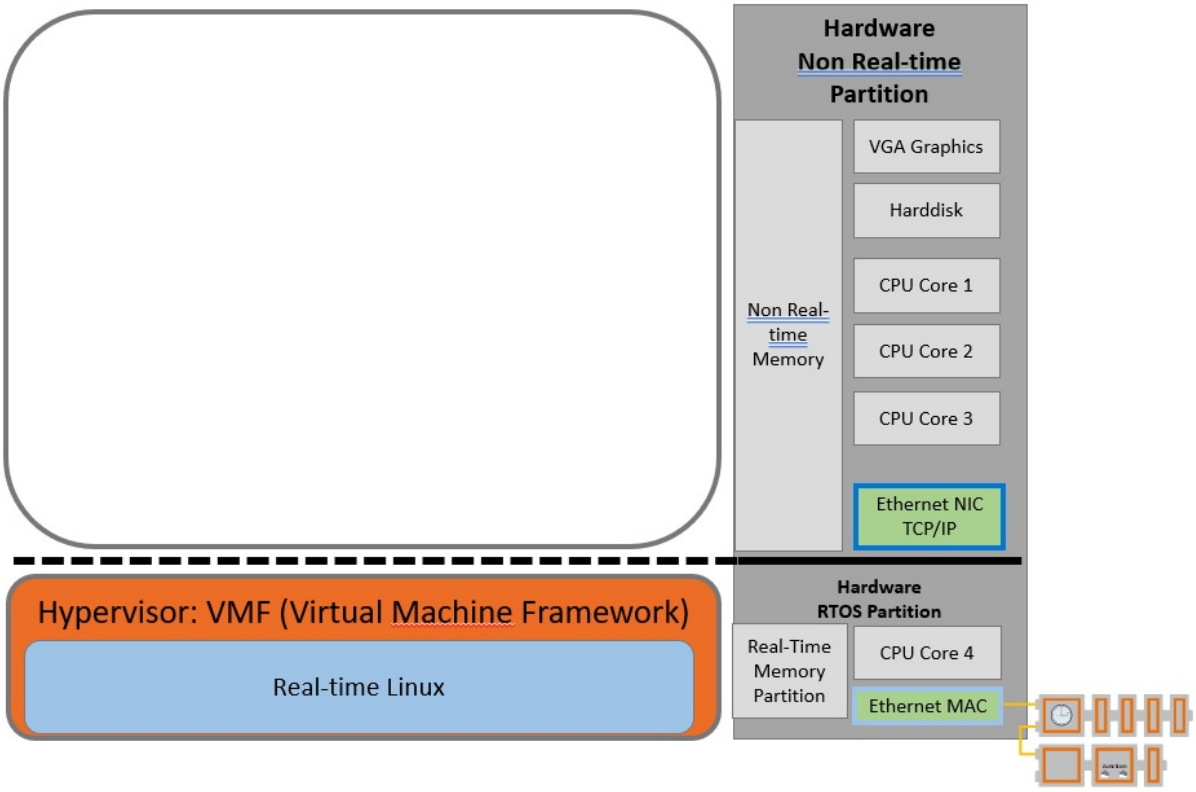
In a final step, you must again synchronize your changes, by pressing the *Sync* button in the Hypervisor Host section.

3.7 Start the VMF (Virtual Machine Framework)

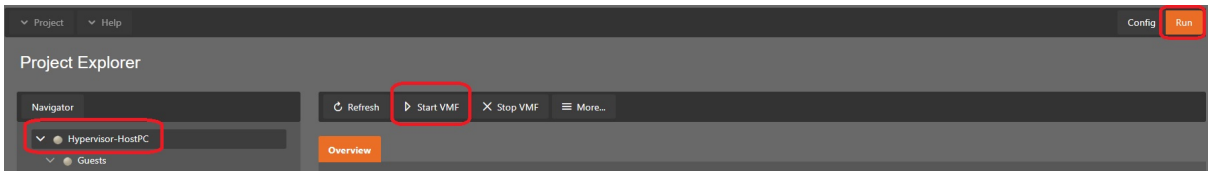
Before we can launch the Real-time Linux guest, we need to start the Virtual Machine Framework (VMF). The VMF will load all configuration information into memory (e.g. partitioning information about the CPUs to be used for Real-time Linux, memory usage, device usage).

In addition, the VMF provides basic services for the Real-time Linux OS (timer handling, virtual console, …). And finally, the VMF provides communication services for all guests to interact with each other (the Communication Subsystem).

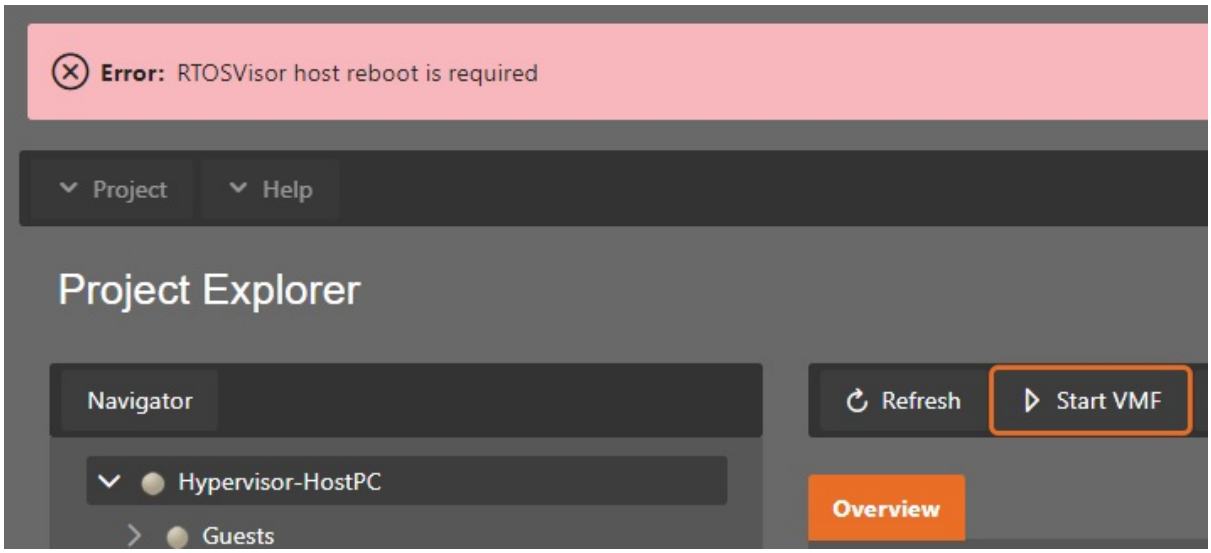
Caution: Every time the configuration has changed, the VMF needs to be restarted to make the updated configuration effective.



Switch into *Run* mode and press the *Start VMF* button.



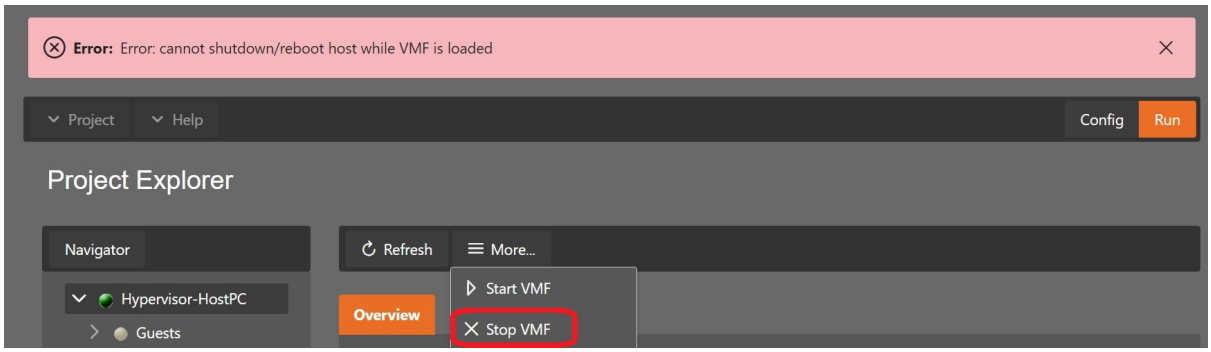
Hint: You may get an error message *RTOSVisor host reboot is required.*



In that case, switch back into *Config* mode, select the Hypervisor Host and press the *Reboot* button.

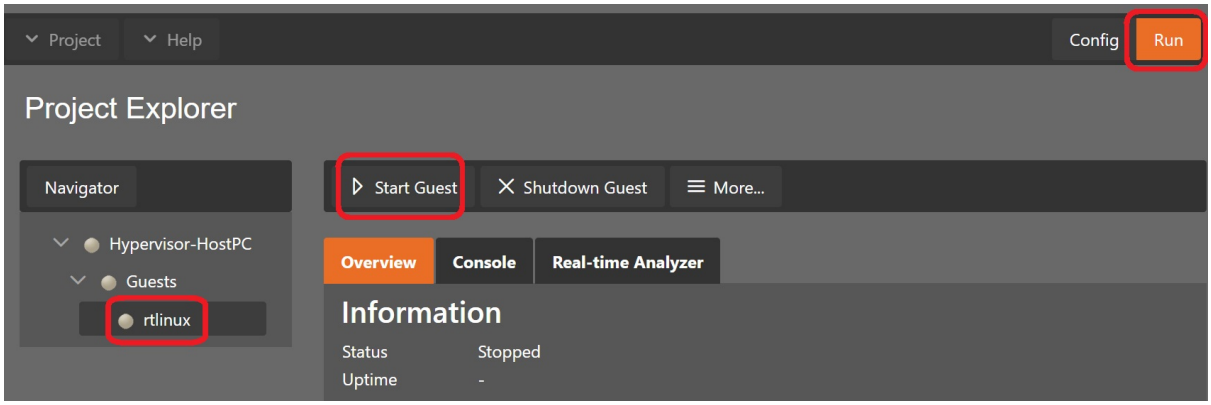
After rebooting, you need to refresh the browser, re-connect with the Hypervisor Host, synchronize, switch into *Run* mode and try to start the VMF again.

In some cases, the System Manager will return errors caused by the VMF still being loaded.
In such a case you need to stop the VMF first.



3.8 Launch the Real-time Linux guest

Switch into *Run* mode, select the *rtlinux* guest and press the *Start Guest* button.

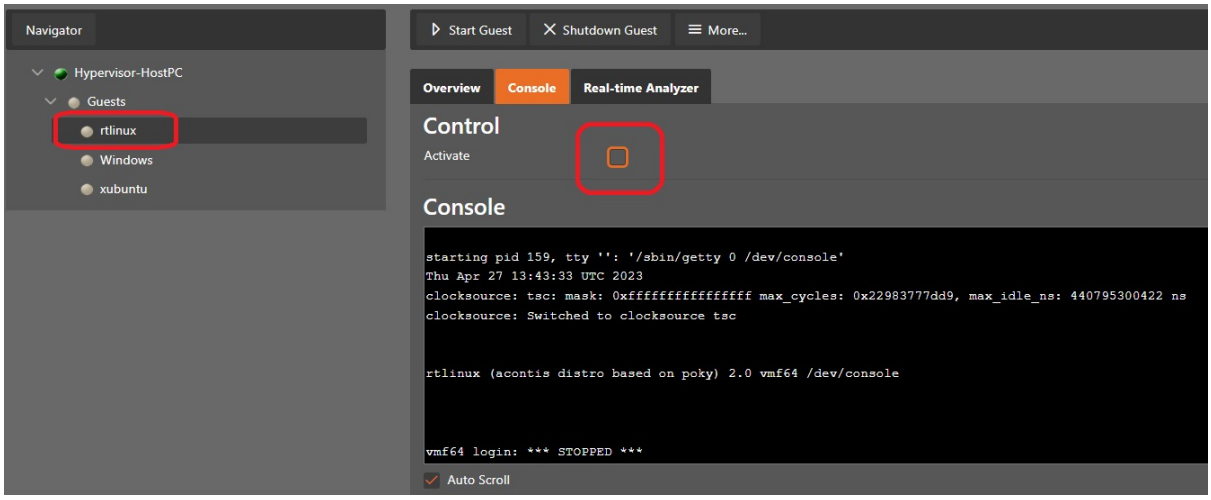


After you started the guest, you need to wait for some time until finally the status indicators should have switched to green.

Select the *rtlinux* guest in the tree view and then switch into the *Console* tab in the right part. You will see the boot messages of Linux until finally you will be able to log in into the Linux shell (user = root, password = root).

Finally you can shutdown the guest by pressing the *Shutdown Guest* button.

If you prefer to use a local terminal to work with the guest console, you must de-activate the console **before** starting the guest.



3.9 Commandline operation (Shell)

Once you have created a guest using the System Manager, you may also run guest commands like *hv_guest_start* or *hv_guest_console* in the shell.
All guests created in the System Manager are located in the `/hv/guests` folder.

Hint: If multiple guests are created by the System Manager, please use the *guest identifier* to find the respective folder where the guest is located.

Please open a shell terminal and type in the following commands.

```
cd /hv/guests/guest0001
hv_guest_start
hv_guest_console
```

You will see the guest booting and you may log in into the Real-time Linux guest.

Press Ctrl-C and then shutdown the guest.

```
cd /hv/guests/guest0001
hv_guest_stop
```

3.10 Add Windows guest

Caution: Assure that the Real-time Linux guest is not running before continuing with this section.

3.10.1 Preparations

There are two options how to set up a Windows guest.

1. Install a new virtual machine (VM) based on an ISO installation media.

2. Re-use an existing VM (an existing guest from a previous configuration).

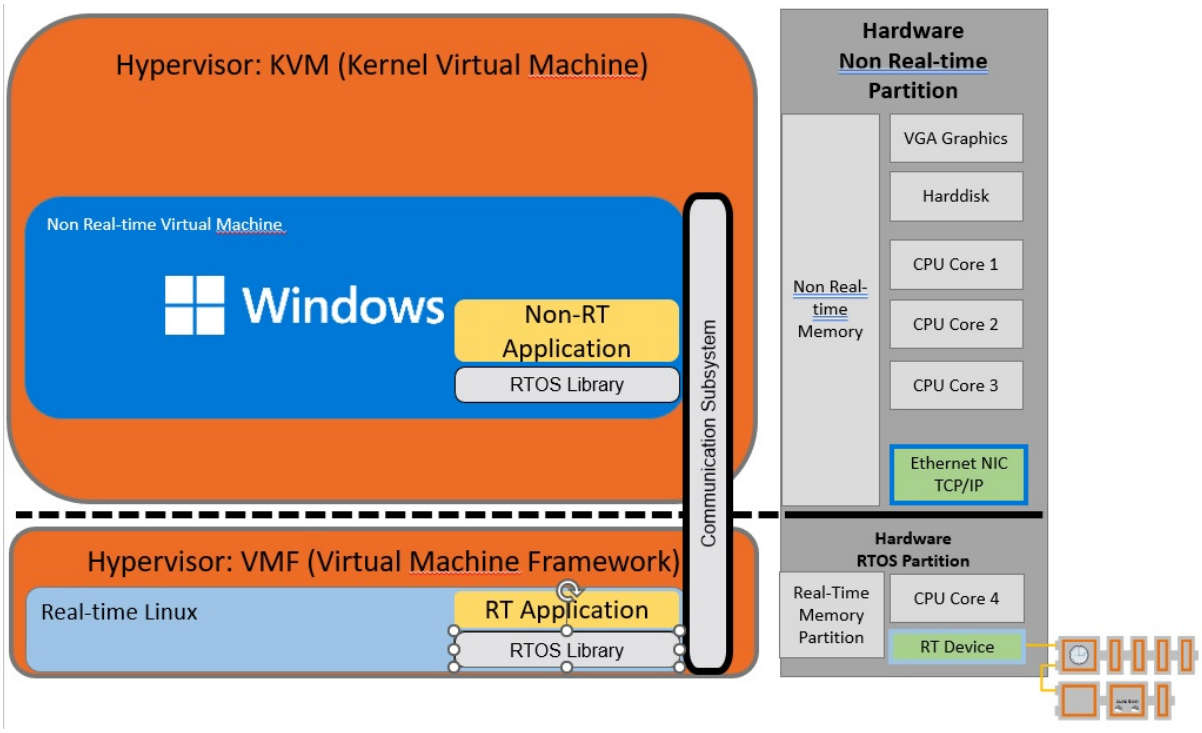
In case you want to install a new VM, you need to copy the installation media onto the Hypervisor Host. How to accomplish this is described in the *Windows Guest Guide*.

In case you want to re-use an existing VM you need to copy the following files onto your Hypervisor Host (into the same folder):

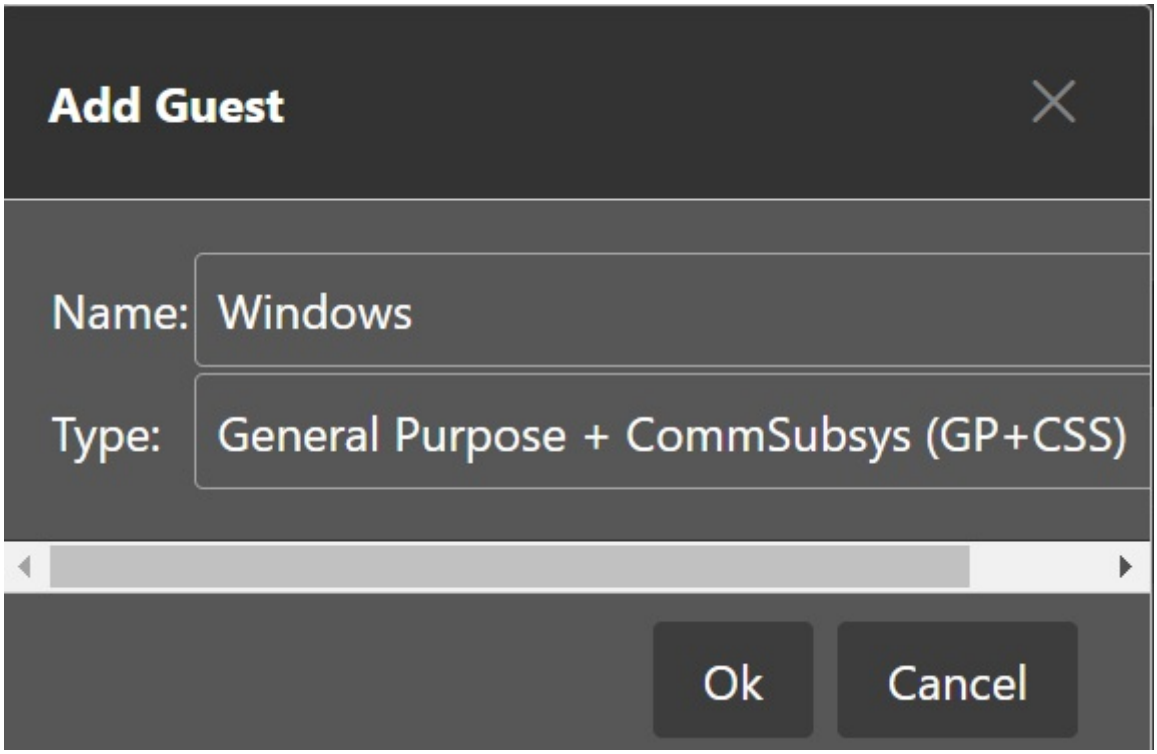
1. all `.qcow2` files
2. the `OVMF_CODE.fd` file
3. the `OVMF_VARS.fd` file

3.10.2 Add the guest

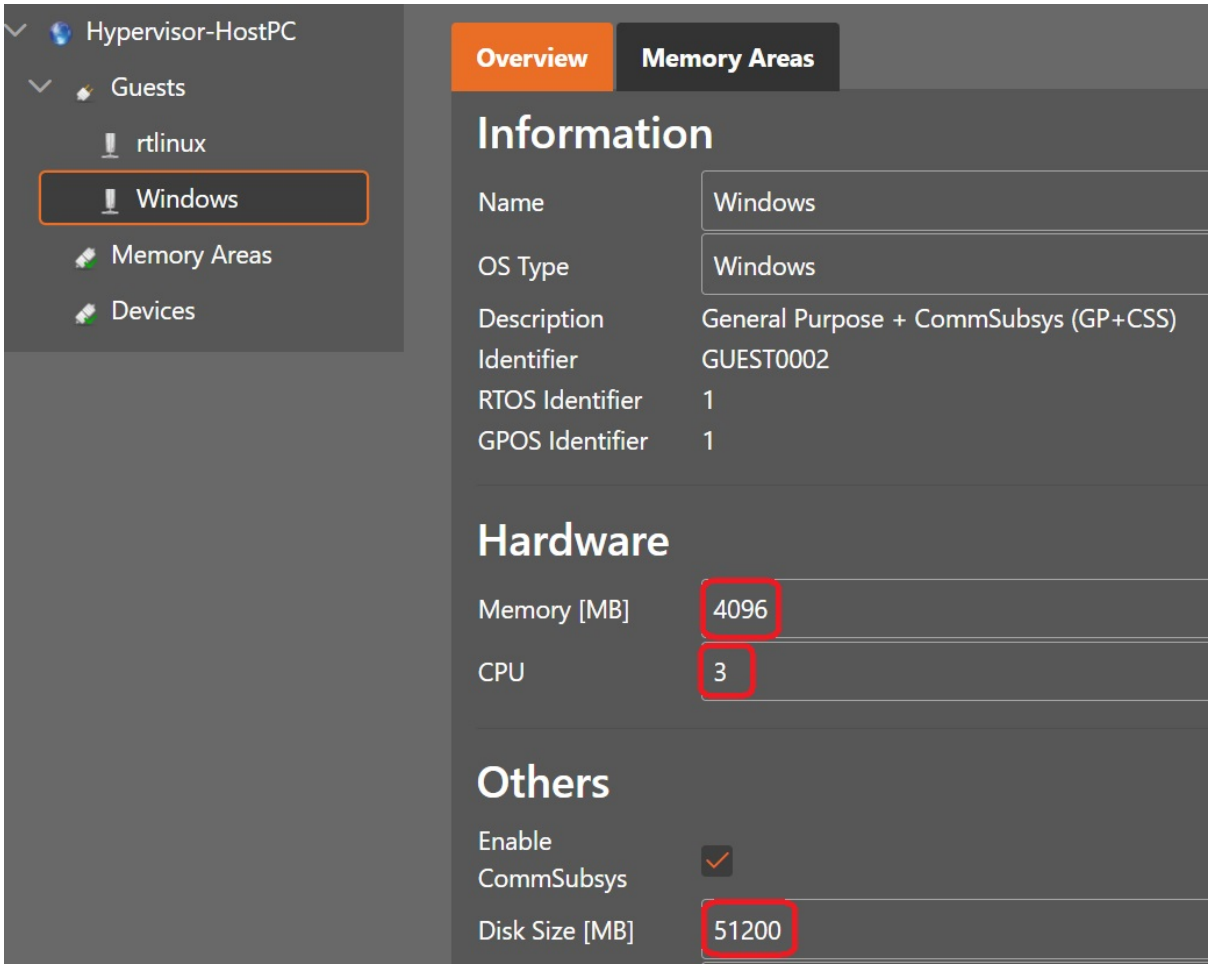
In the next step we will add a Windows guest. This guest runs under control of the KVM Hypervisor. We will also enable a connection to the Communication Subsystem which will allow communication between the Windows guest and Real-time Linux.



Select *Config* mode and add a Windows guest.
To achieve this, you must select the *General Purpose + CommSubsys (GP+CSS)* guest type.
Use *Windows* for the guest name.

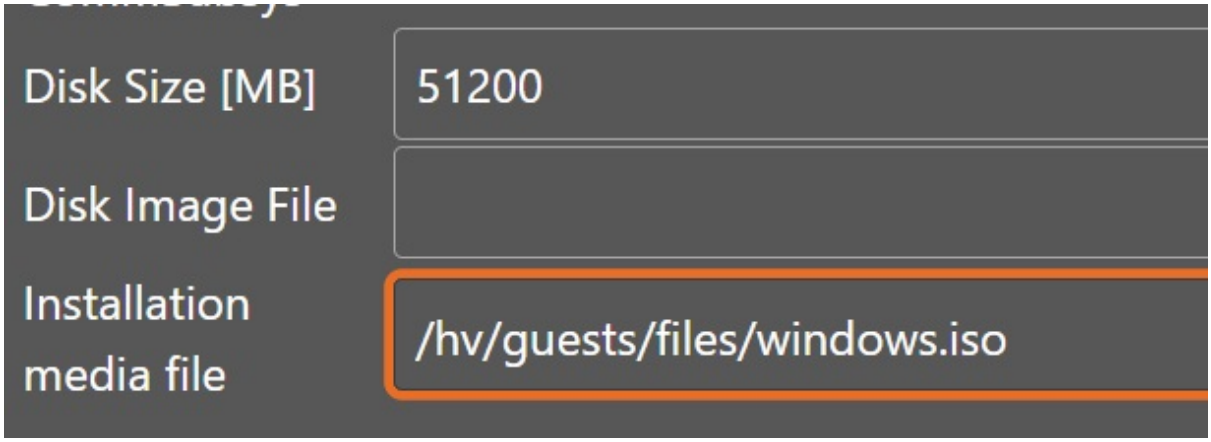


You may adjust some basic properties in the *Overview* tab of the guest.

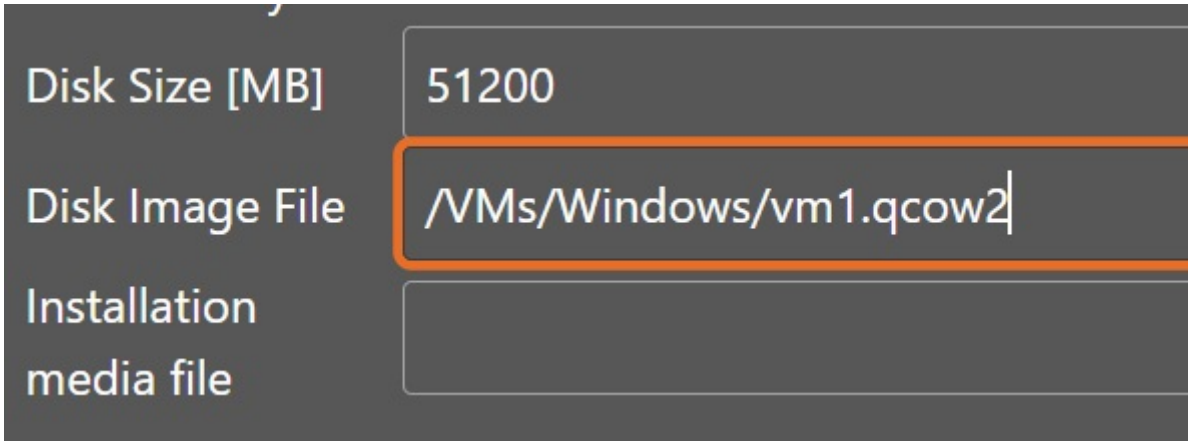


Caution: The Disk Size will become effective only when a new VM is created. If the disk size shall be adjusted at a later time, you need to use the *qemu-img* tool. See here for more information: https://blog.programster.org/qemu-img-cheatsheet

If you want to install a new VM, insert the installation media file in the *Installation media file* edit field:



If you want to re-use an existing VM, insert the .qcow2 file in the *Disk Image File* edit field:



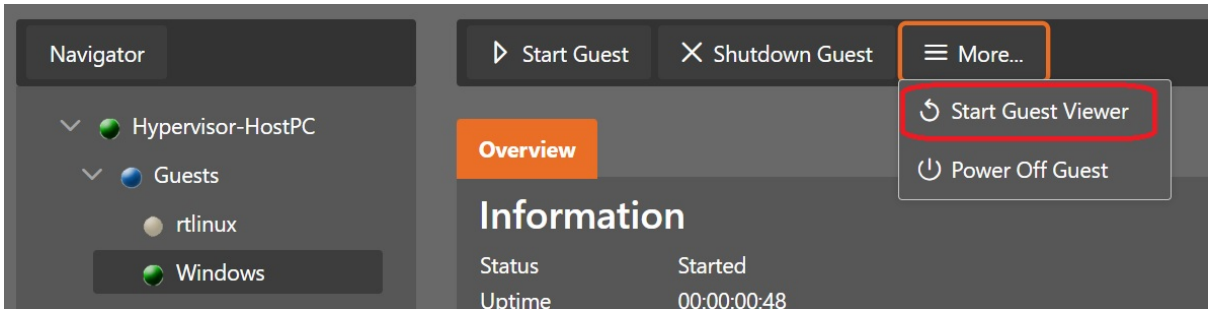
You need to press the *Sync* button to make the configuration effective.

3.11 Launch the Windows guest

Switch into *Run* mode, select the *Windows* guest and press the *Start Guest* button.

After you started the guest, you need to wait for some time until finally the status indicators should have switched to green.

Select the *Windows* guest in the tree view and then start the guest viewer via the *More…* combo-box.



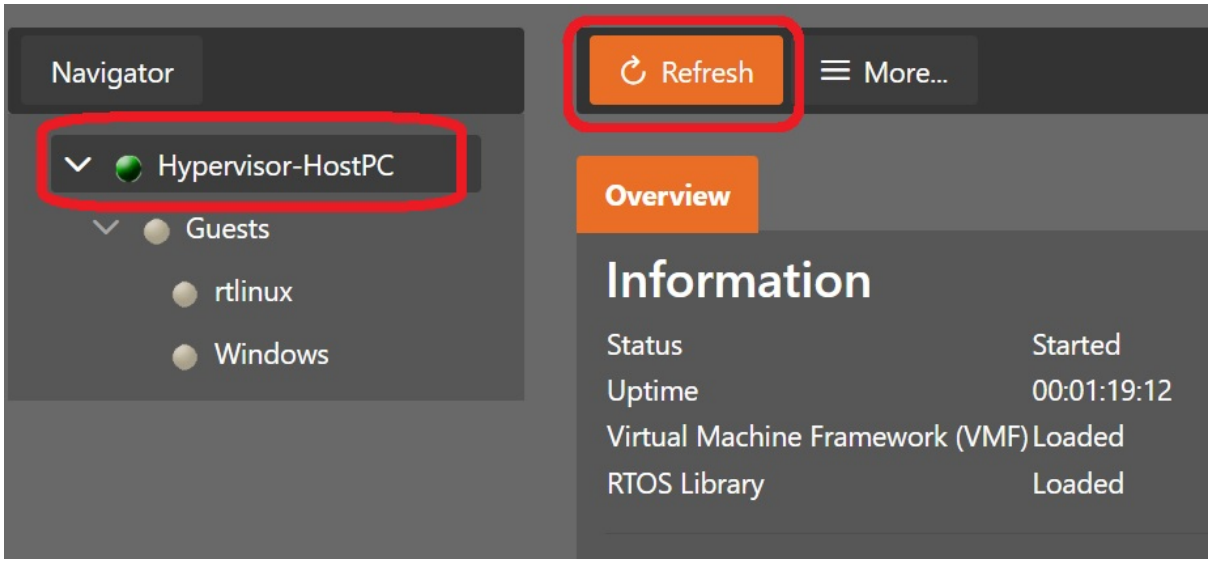
In case you get an error *No Display device defined, please login via the graphical desktop and run the command: hv_sysmgr restart*, then you need to start a terminal in the Hypervisor Host (**not** in a remote logged in shell) and run the `hv_sysmgr restart` command. After running this command you need to refresh the browser and reconnect again.

Hint: You can launch the viewer also from a shell console in the Hypervisor Host.

```
cd /hv/guests/guest0002
hv_guest_console
```

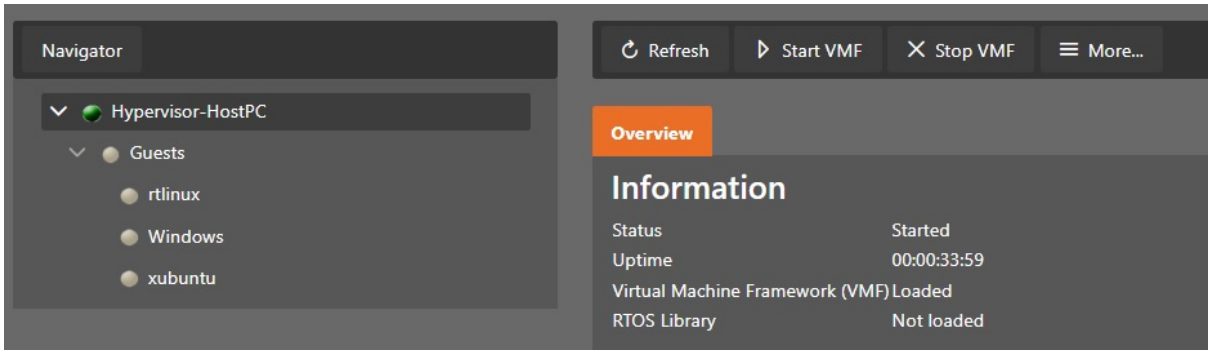
If you have chosen to install a new VM, please follow the instructions in the *Windows Guest Guide* on how to install this VM.

Finally you can shutdown the guest by pressing the *Shutdown Guest* button. Shutdown of a KVM guest may take some time. The System Manager will not wait until the shutdown finished. If you want to verify if the guest actually is shut down, you need to select the Hypervisor Host and press the *Refresh* button.

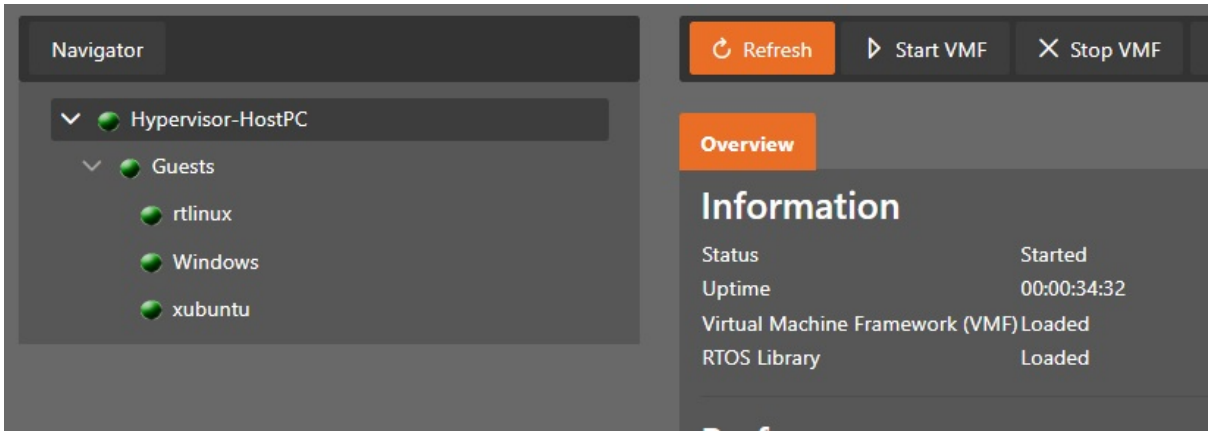


3.12 Start and/or stop all guests

If you want to start or stop all guests, switch into *Run* mode and press the *Start all Guests* or *Stop all Guests* button in the *More…* menu. Please note, you must start the VMF before starting any guests. This will initiate the startup or shutdown of the guests. The System Manager may not correctly show the status of the guests.



You should press the *Refresh* button to update the guest status.

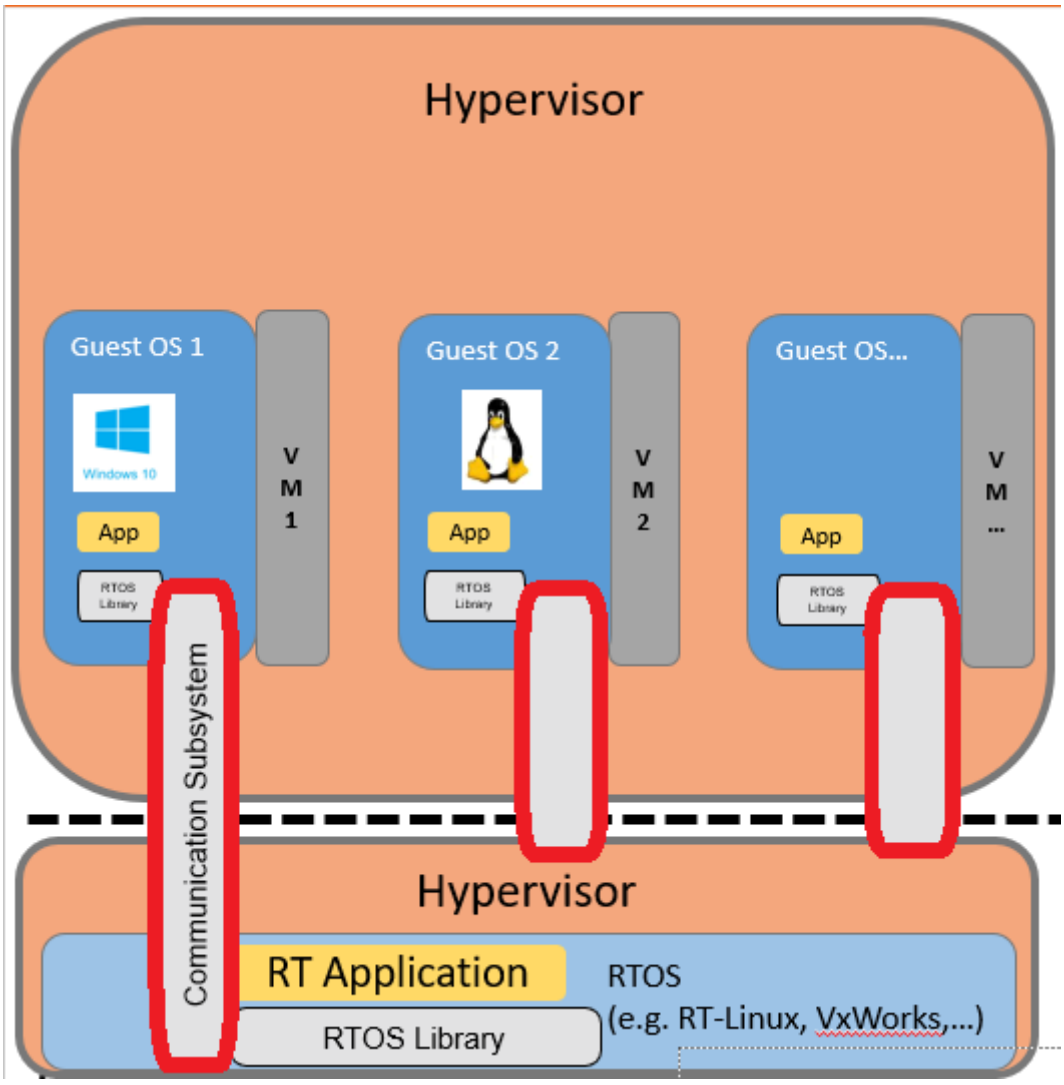


4 Guest Configuration

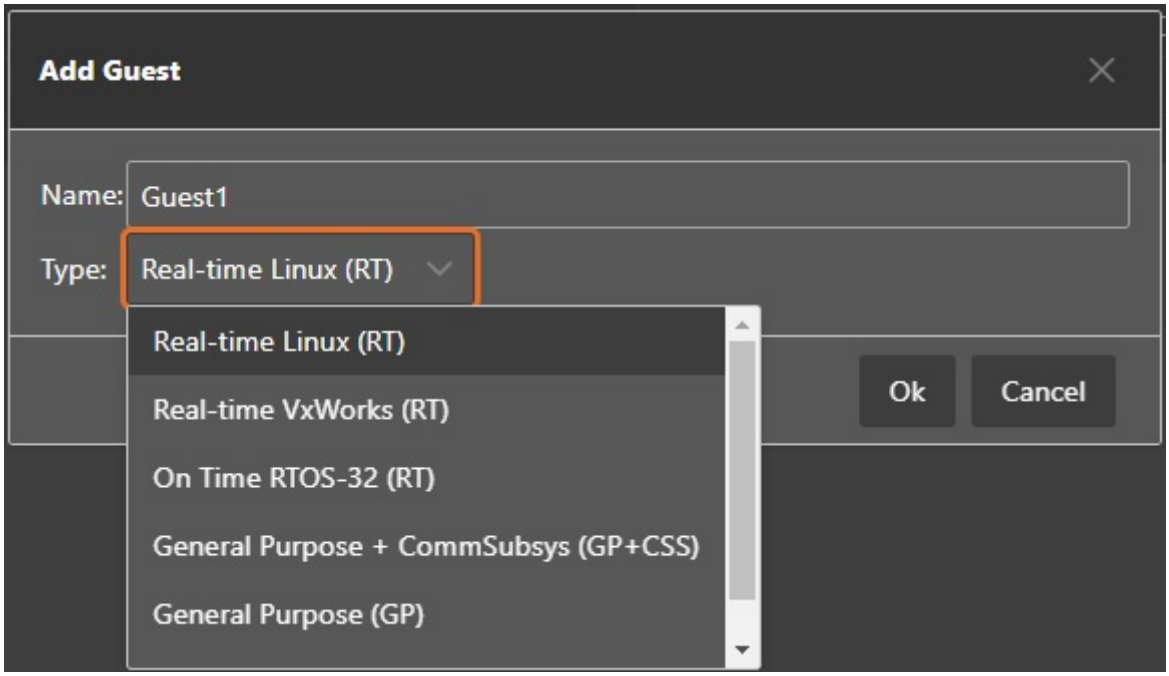
4.1 Guest Types

The RTOSVisor supports Real-time operating system guests (e.g. Real-time Linux) as well as non Real-time (General Purpose) operating systems (e.g. Windows). While the Real-time operating systems are adapted (para-virtualized) to the underlying Real-time hypervisor and shipped as part of the RTOSVisor, the General Purpose operating systems can be used without modification and need to be installed prior to using them.

All guests can use the Communication Subsystem which provides various means to communicate between the guests. General Purpose operating systems may optionally be used without a connection to the Communication Subsystem. They may interact with the other guests using TCP/IP bridging, see the Hypervisor Manual for details.



The following guest types are supported.

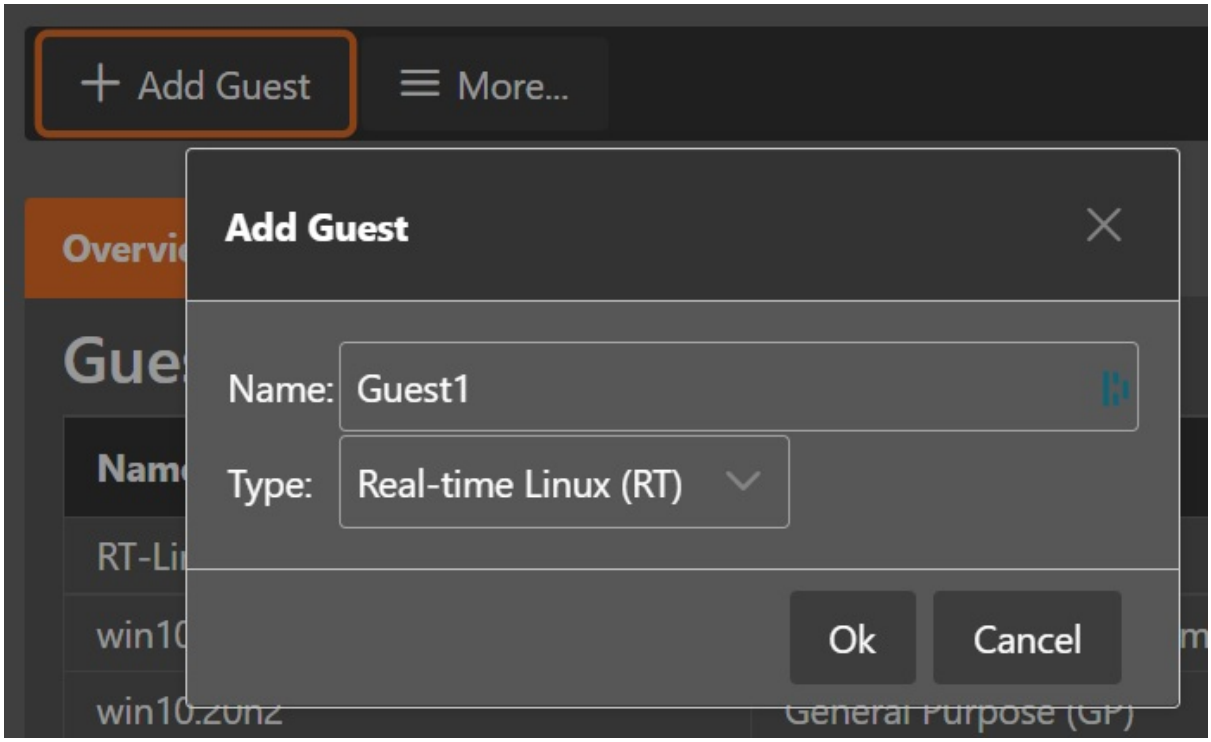


- **Real-time Linux:** Real-time Linux real-time guest
- **VxWorks:** VxWorks real-time guest
- **On Time RTOS-32:** RTOS-32 real-time guest
- **General Purpose + CommSubsystem:** A General Purpose guest which can attach to the Communication Subsystem
- **General Purpose:** A General Purpose guest without Communication Subsystem access

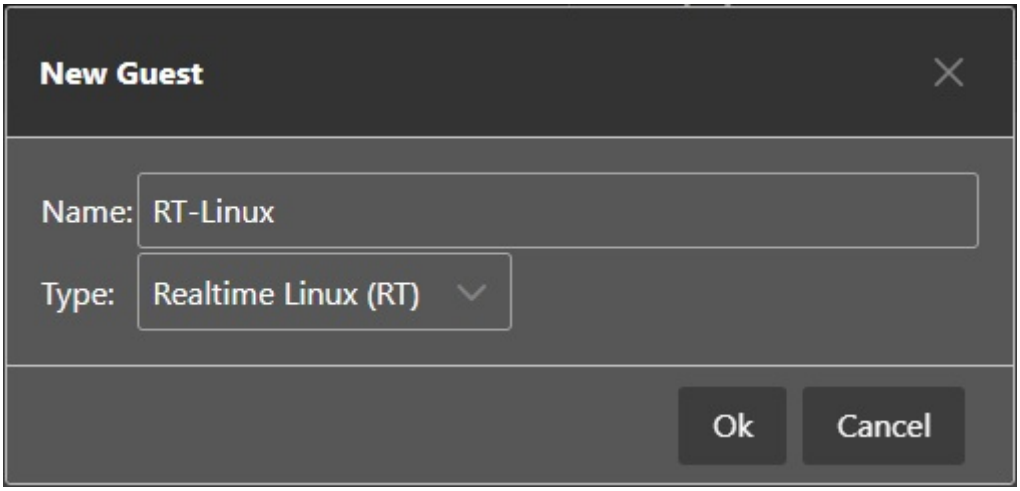
Caution: In case you intend to add at least one Real-time guest or one General Purpose guest with Communication Subsystem access, the **first** guest must be a Real-time guest!

4.2 Add RTOS guests

Select `Guests` and then `+ Add Guest`.
A list of operating systems is shown.

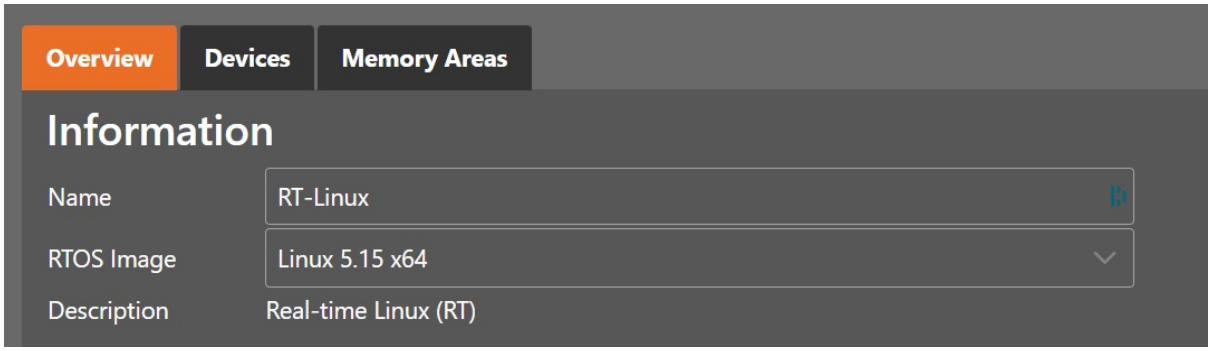


Then select one of the RTOS guests (e.g. Real-time Linux). You should also assign a reasonable name for this guest (e.g. RT-Linux).



4.2.1 Select the RTOS image to boot

The RTOS may offer multiple binary images which can be booted (e.g. different RTOS versions).
You need to select the RTOS in the `Overview` tab.

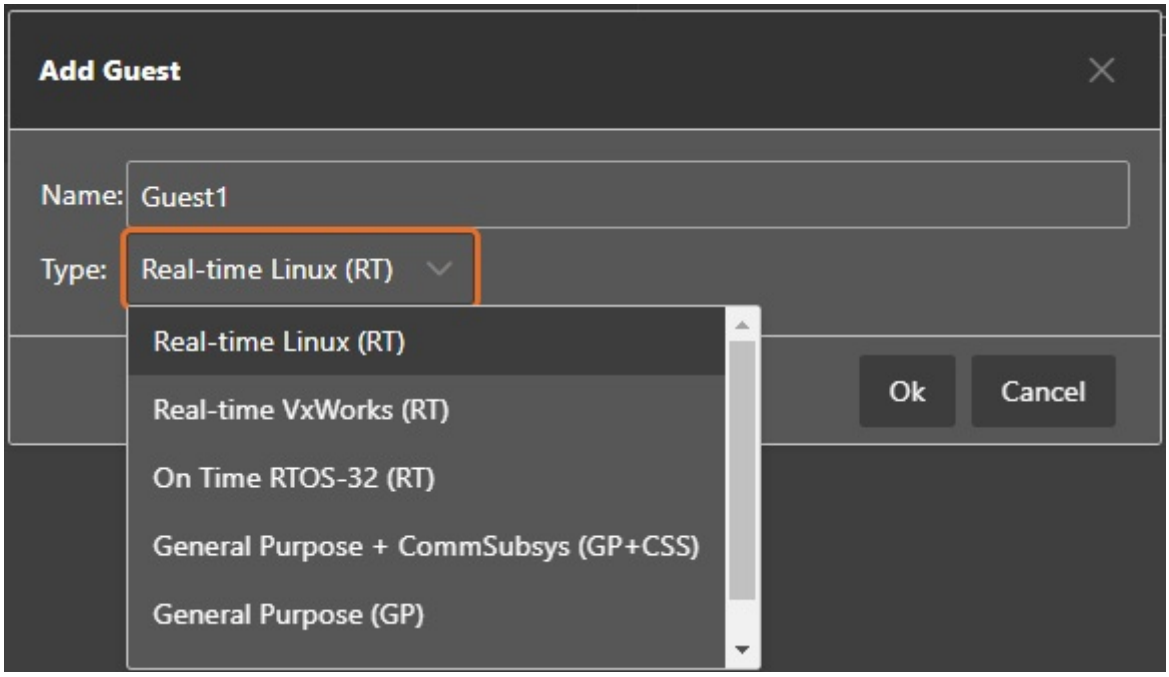


Then select the image you like to use.

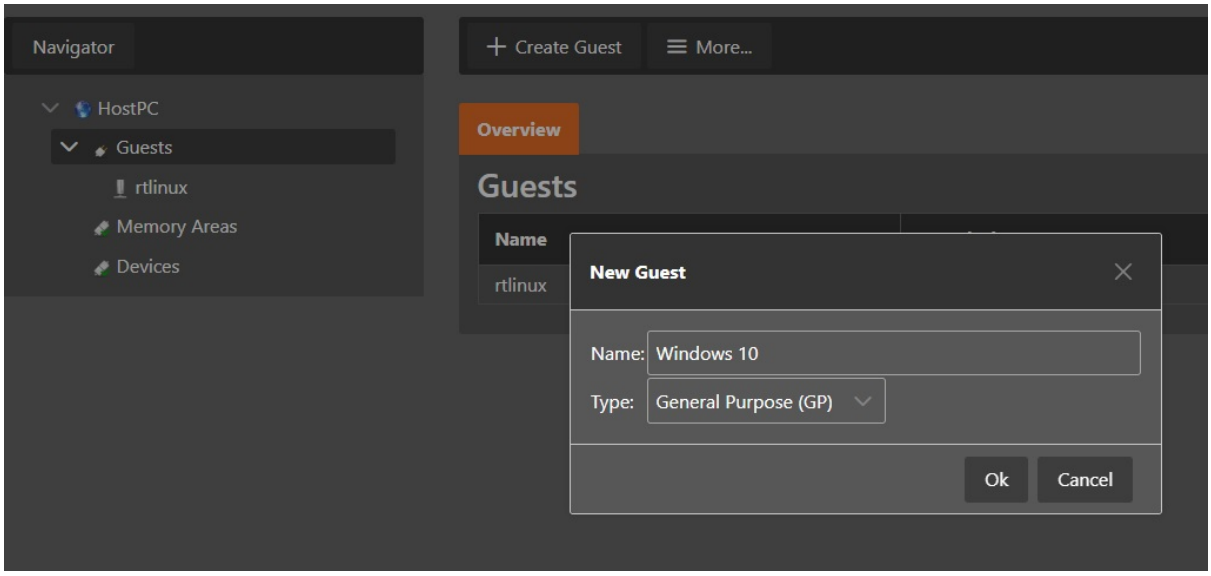
4.3 Add General Purpose guests

Adding a General Purpose guest is done in a very similar way as for Real-time guests. You can select whether such a guest shall be able to attach to the Communication Subsystem or not.

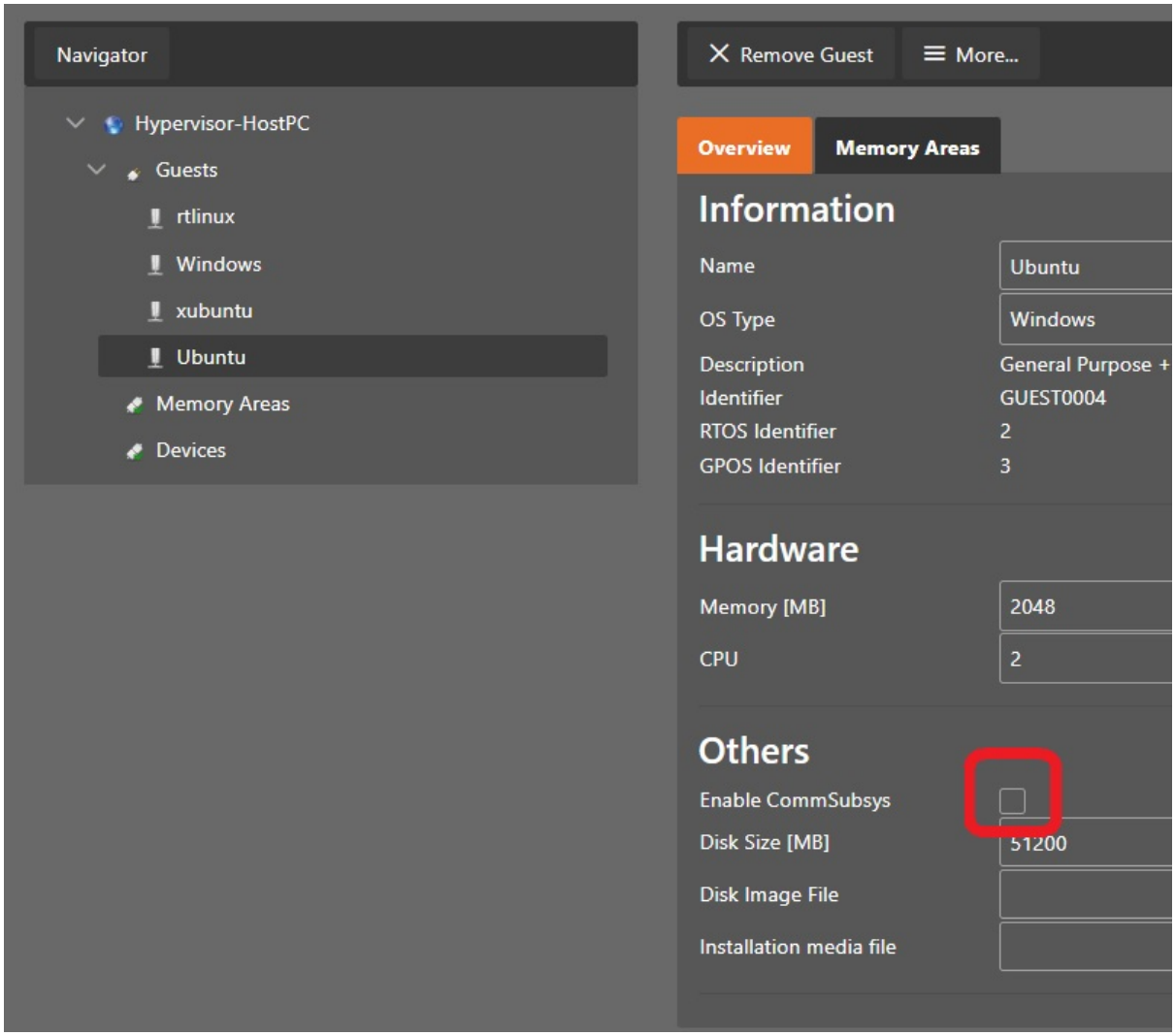
Select `Guests` and then `+ Add Guest`.
A list of operating systems is shown.



Then select either the `General Purpose` or `General Purpose + CommSubsys` guest.
If you want the Guest to use the communication subsystem (virtual network, shared memory etc.) to communicate with the RTOS, then you need to select the latter one.
Please note, the number of OSes which can use the communication subsystem is limited (depending on the version of the software).



Hint: If the guest is configured to get access to the Communication Subsystem, by default access is enabled. Prior to starting such a guest, the VMF (Virtual Machine Framework) needs to be started. You may optionally disable access to the Communication Subsystem, then you do not need to start the VMF.



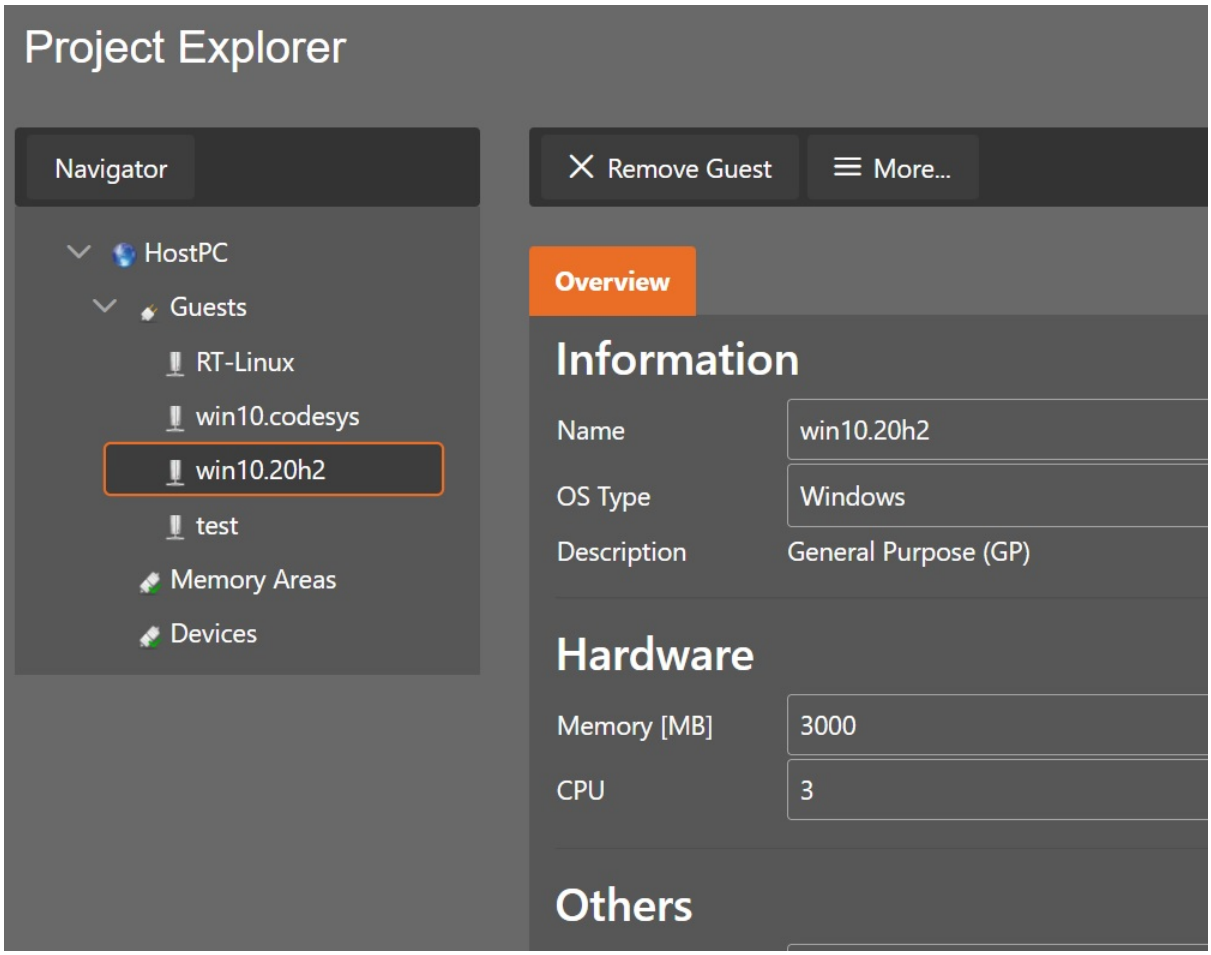
4.3.1 Install Windows/Linux guest

If you want to install a new Windows or Linux guest, please follow the steps below.

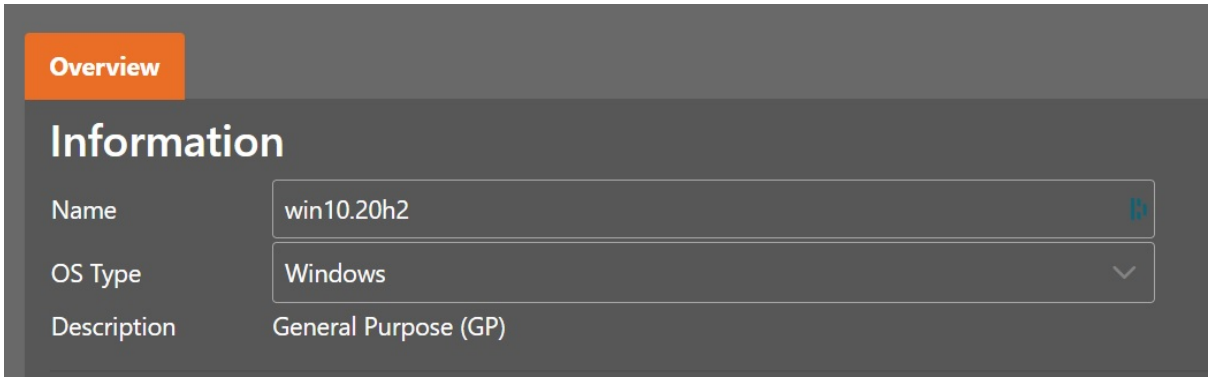
First you need to copy the installation media ISO file to the Hypervisor Host (e.g. `/hv/iso`)
Then you need to configure the guest hardware: Memory (RAM), number of CPUs, disk size, disk image path.
The installation media file must be set appropriately.
By specifying the `Disk Image File`, the location and name of the guest image files can be determined. The `Disk Image File` entry always consists of a path (starting with `/hv/VMs`) and filename with the `.qcow2` extension (e.g., `/hv/VMs/windows/win10_20h2.qcow2`).

Caution: If no `Disk Image File` is specified, then the image will be created in the guest folder, and if the guest is deleted, the image and thus the entire installation and all settings in the Windows or

Ubuntu guest will also be deleted.



You must set the “OS Type” to the appropriate type.

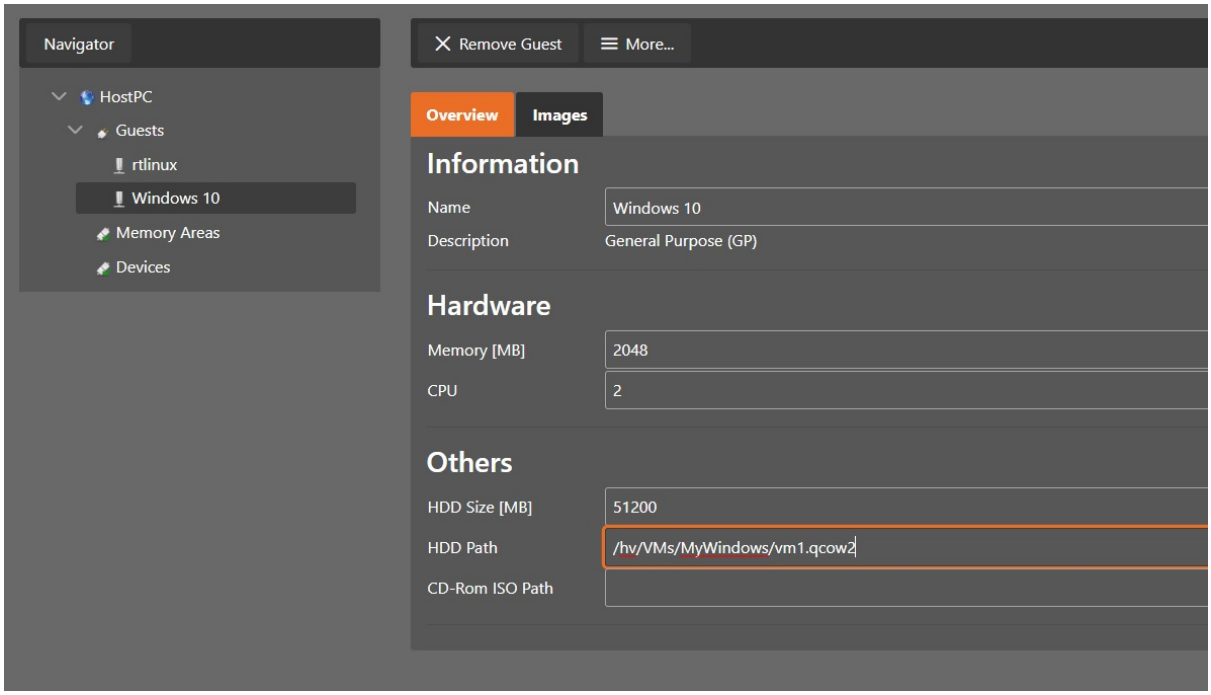


To install the guest, you need to *synchronize* the configuration with the Hypervisor Host first and then run the guest. See *Guest Operation* for more details. The *Hypervisor Windows Guest Guide* also shows step by step, how to install a new Windows guest.

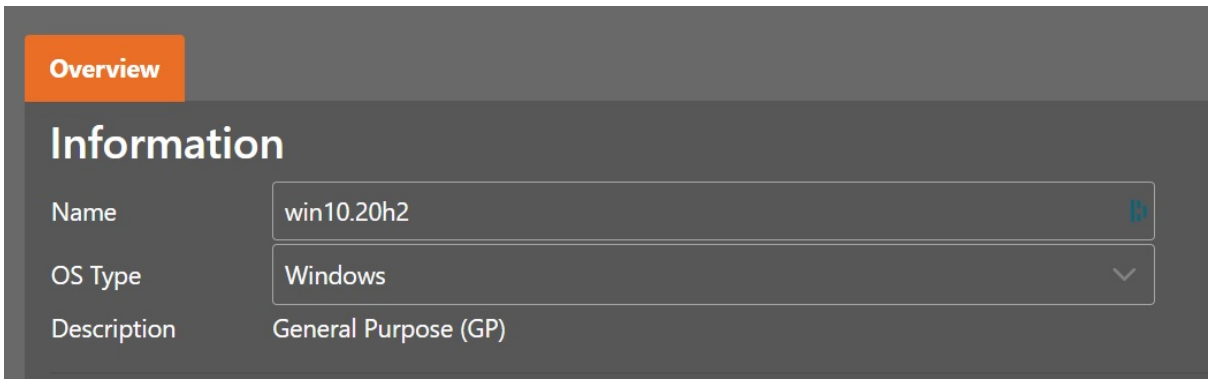
4.3.2 Select existing Windows/Linux guest

If you want to re-use an existing Windows or Linux guest image, please follow the steps below.

First you need to copy the guest image files to the Hypervisor Host.
Assure the guest image files are complete, they consist of one or multiple `.qcow2` file(s) and two EFI BIOS files: `OVMF_CODE.fd` and `OVMF_VARS.fd`.
Then you need to configure the guest hardware: Memory (RAM), number of CPUs and the disk image path.
The guest disk image filename must be set appropriately.



Set the “OS Type” appropriately.



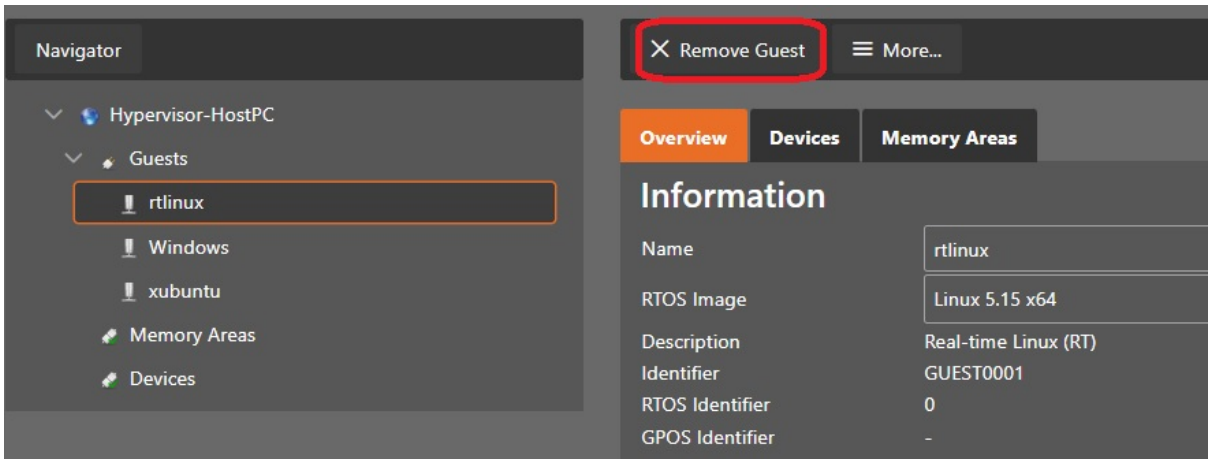
To run the guest, you need to *synchronize* the configuration with the Hypervisor Host first and then run the guest. See *Guest Operation* for more details.

4.3.3 Step by step example

For an example, how to add a Windows guest, see *here*.

4.4 Remove guests

To remove a guest, you need to select the respective guest and then press the *Remove Guest* button.



Caution: All configuration settings for this guest will be removed, it is not possible to restore those settings. You may save the current project to preserve the settings of all currently configured guests: *Project Management*.

4.5 Synchronization

Before the configuration becomes effective, you must synchronize the System Manager configuration with the Hypervisor Host. Take a look into the *Synchronization* chapter for details.

Caution: Writing a new configuration to the Hypervisor Host will destroy the currently active configuration!

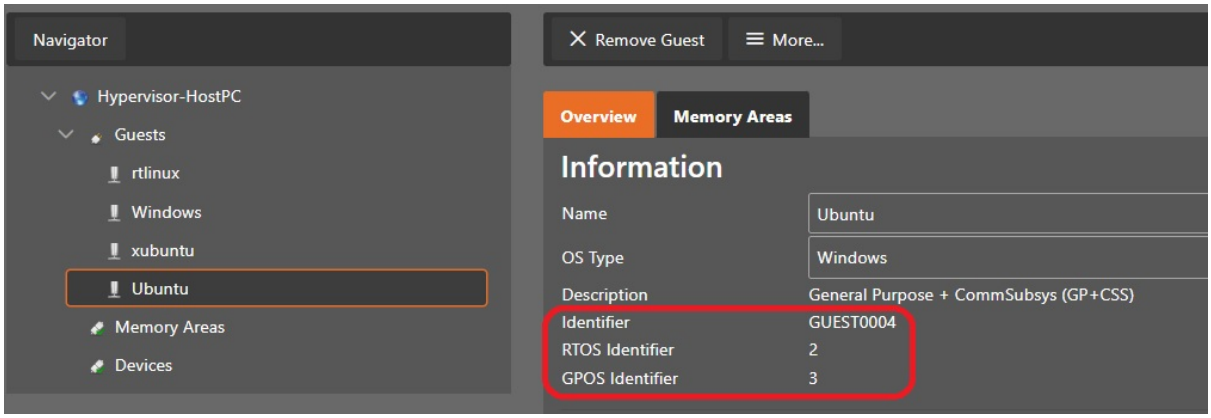
4.6 Guest Identifiers

When adding a new guest, it will be assigned a unique **Guest Identifier** GUEST0001, GUEST0002 etc.. All configuration settings of the guests are located in the respective guest folders, for example `/hv/guests/guest0001` or `/hv/guests/guest0002`.

All Real-time guests as well as General Purpose guests with access to the Communication Subsystem will get a unique identifier, the **RTOS Identifier**. The number of such guests is currently limited to 5, the range for the *RTOS Identifier* will be from 0 to 4. This RTOS identifier is required as a parameter for General Purpose guests when they attach to the Communication Subsystem.

All General Purpose guests, regardless whether they have access to the Communication Subsystem will also get a unique identifier, the **GPOS Identifier**. The number of such guests is currently limited to 9, the range for the *GPOS Identifier* will be from 1 to 9.

This following example shows a Ubuntu guest with access to the Communication Subsystem. The guest id is GUEST0004, which means that all settings are stored in `/hv/guests/guest0004`. The RTOS identifier is *2* which means that the guest needs to use this value when attaching to the Communication Subsystem. The GPOS identifier is *3* which means it is the third General Purpose guest configured in this project.



4.7 Configuration Files

All configuration files for a specific guest are located in `/hv/guests/guest0001`, `/hv/guests/guest0002` and so on. See also *Guest Identifiers*.

The following files are created automatically, when any guest is generated:

- **guest_config.sh**

  Basic guest settings, needed when starting the guest. **Do not change!**

- **usr_guest_config.sh**

  User specific guest settings, used when starting the guest. You may override settings defined in `guest_config.sh`.

- **guest.config**

  Default guest configuration settings. **Do not change!**

- **usr.config**

  User specific guest configuration settings. You may override settings defined in `guest.config` (and/or in `device.config` for Real-time guests).

- **hv_guest_autostart_xfce.sh**

  Helper script to launch the guest console, used if a guest shall be started automatically. **Do not change!**

- **sysmgr_guest.config**

  System Manager specific configuration information. **Do not change!**

- **vm_shutdown_hook.sh**

  Script which is executed after the guest was shutdown.

4.7.1 RTOS guests

The following files are created automatically, when an RTOS guest is generated:

- **device.config**

  Guest device configuration settings. **Do not change!**

- **hv_guest_autostart.service**

  Service file, used if the guest shall be started automatically. **Do not change!**

4.7.2 General Purpose guests

The following files are created automatically, when a General Purpose guest is generated:

- **guest_gateway.config**

  Guest gateway configuration settings. Specific ports inside the guest can be forwarded to a specific IP address and port. Currently only supported for Windows guests.

- **hv_guest_autostart.service**

  Service file, used if the guest shall be started automatically. **Do not change!**

- **hv_guest_autostart_standalone.service**

  Service file, used if the guest shall be started automatically in standalone mode. **Do not change!**

- **OVMF_CODE.fd, OVMF_VARS.fd**

  Default EFI files, used if the guest is configured to use UEFI BIOS. These files can be removed if Legacy BIOS is used.

4.8 User specific configuration

If settings not supported by the System Manager need to be changed, they must be modified by adjusting the corresponding `usr.config` or `usr_guest_config.sh` files. See *Configuration Files* for an overview of all configuration files.

Especially for General Purpose guests there are many settings which can only be adjusted in the `usr_guest_config.sh` file.

4.9 RTOS-32 guests

4.9.1 Boot Image

The image file which is loaded into memory is defined in `guest_config.sh` using the **osImage** setting. By default the RTOS-32 application Loader image will be booted.

4.9.2 Filesystem

The file system root is defined in `guest.config` (section [Host\FileServer], parameter "HomeDir"). By default it will point to the RTOS-32 example guest. You may change this to another location, for example the guest folder (e.g. `/hv/guests/guest0001`).

4.9.3 DLL Loading

If the default image (the RTOS-32 Loader image) is booted, the application itself is stored in one or multiple DLLs (DLMs). The main DLL is defined in `guest.config` (section [Rtos\Loader], parameter "DllName"). By default it will point to a link defined in the RTOS-32 example guest.

> **Caution:** All DLLs must be located in the filesystem root folder.

4.10 RTOS Guest (Project) Templates

The System Manager provides two XML-based template files under */hv/templates/* that define the “skeleton” of any new RTOS guest:

- **os.xml** Defines the available operating system templates (types, subfolders to copy into the guest folder, etc.). See excerpt below:

```
<?xml version="1.0" encoding="utf-8"?>
<Config>
  <OperatingSystem>
    <Property Key="Id" Value="RtLinux_SMPAUTO" />
    <Property Key="Name" Value="Real-time Linux (RT)" />
    <Property Key="Type" Value="linux_SMPAUTO" />
    <Property Key="SubFolders" Value="rtos_generic;rt-linux;rt-linux_
↪SMPAUTO" />
    <Property Key="HasRtSupport" Value="true" />
    <Property Key="HasVmfAccess" Value="true" />
    <Property Key="HasRtAnalyzer" Value="true" />
    <Property Key="HasAutoSmpSupport" Value="true" />
  </OperatingSystem>
  ...
</Config>
```

- **images.xml** Defines the VMF-loadable RTOS guest images (paths, memory sizing, alignment, SMP flags, etc.). See excerpt below:

```
<?xml version="1.0" encoding="utf-8"?>
<Config>
  <Image>
    <Property Key="Id" Value="Linux515x64" />
    <Property Key="Name" Value="Linux 5.15 x64" />
    <Property Key="Type" Value="linux" />
    <Property Key="Path" Value="../../templates/rt-linux/rtlinux515.
↪x64.bin" />
    <Property Key="Arch" Value="64" />
    <Property Key="MinSize" Value="192M" />
    <Property Key="DefaultSize" Value="192M" />
    <Property Key="Align" Value="4M" />
    <Property Key="ShmAsOsHeap" Value="multiple" />
  </Image>
  …
</Config>
```

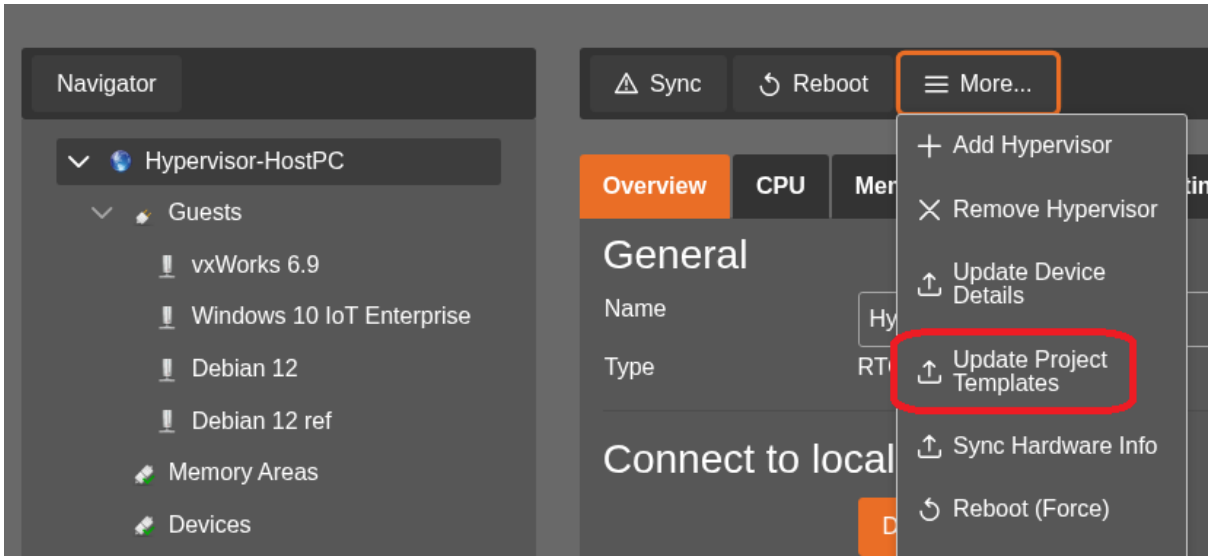
4.10.1 Example: Adding a New Real-time Linux SMP Image

To add add a new custom Real-time Linux image with SMP support, edit **images.xml** as follows:

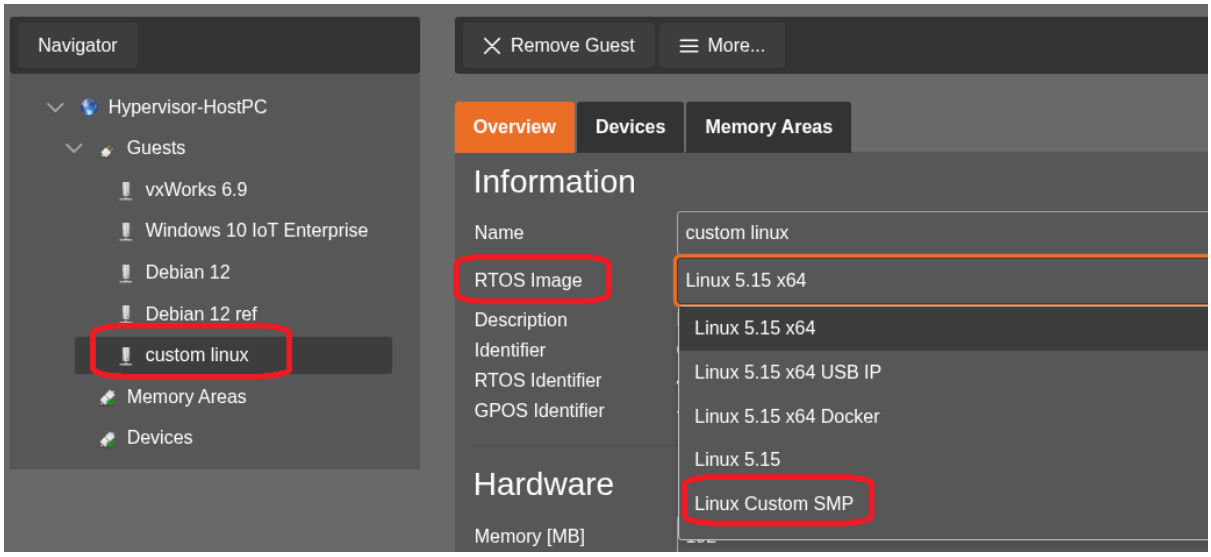
1. **Open** */hv/templates/images.xml*.
2. **Duplicate** an appropriate existing image entry <*Image*> block inside the top-level <*Config*>: In our example, this would be *Linux515x64_SMPAUTO* for a SMP supported Linux image. Take specific care of the *Type* property which defines the operating system type: *Linux_SMPAUTO*. This value must match with the *Type* property in the **os.xml** configuration file. Adjust the other values according to your need (e.g. image name to load, RAM size).

```
<Image>
  <Property Key="Id"                    Value="Linux_Custom_SMPAUTO"␣
↪/>
  <Property Key="Name"                  Value="Linux Custom SMP" />
  <Property Key="Type"                  Value="linux_SMPAUTO" />
  <Property Key="Path"                  Value="../../templates/rt-
↪linux/rtlinux_custom.bin" />
  <Property Key="Arch"                  Value="64" />
  <Property Key="MinSize"               Value="512M" />
  <Property Key="DefaultSize"           Value="512M" />
  <Property Key="Align"                 Value="4M" />
  <Property Key="ShmAsOsHeap"           Value="multiple" />
</Image>
```

3. **Save** the file.
4. In the *Config* mode of the System Manager, select the Hypervisor in the Navigator tab. Then select *More - Update Project Templates* to update the template.



5. The new entry **Linux Custom SMP** will now appear in the RTOS Image combobox when a Real-time Linux guest is configured.



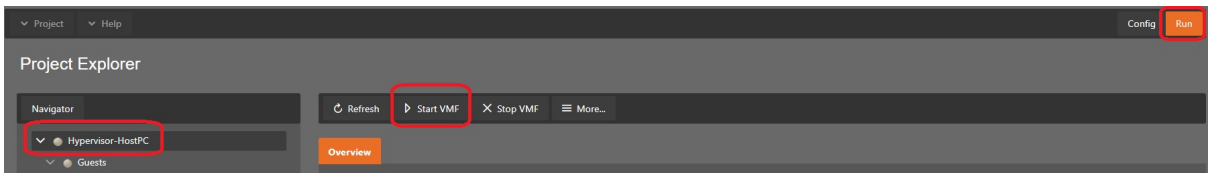
5 Guest Operation

5.1 Synchronization

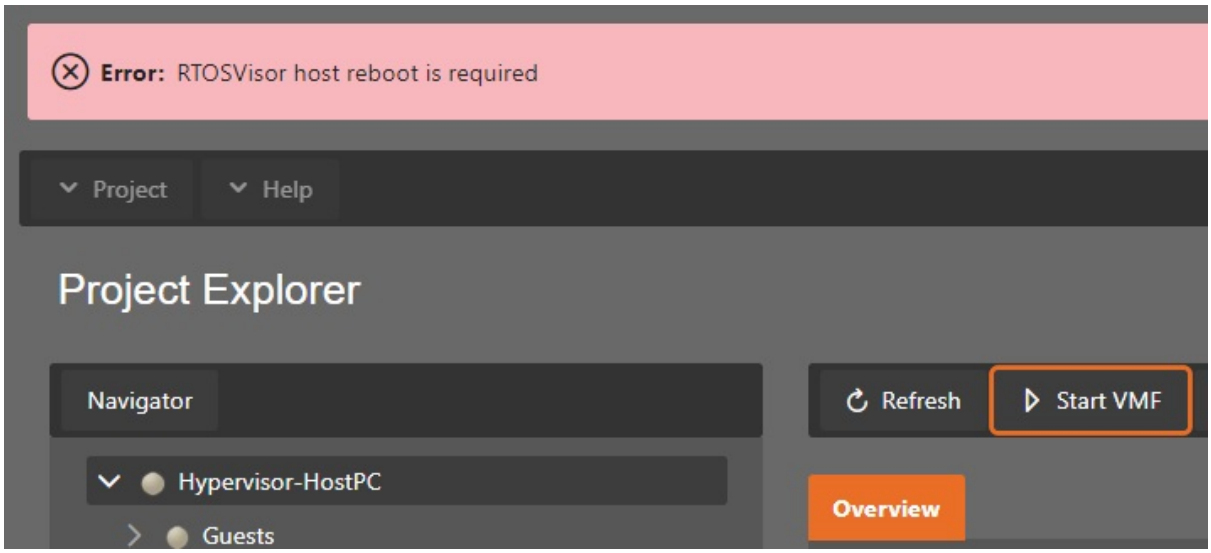
After finishing the configuration you need to *synchronize* the updated configuration with the Hypervisor Host.

5.2 Start the VMF (Virtual Machine Framework)

To start a guest you need to switch into *Run* mode first. If any of the configured guests is using the Communication Subsystem, you need to *start the VMF first*.



Hint: You may get an error message *Hypervisor Host reboot is required.*

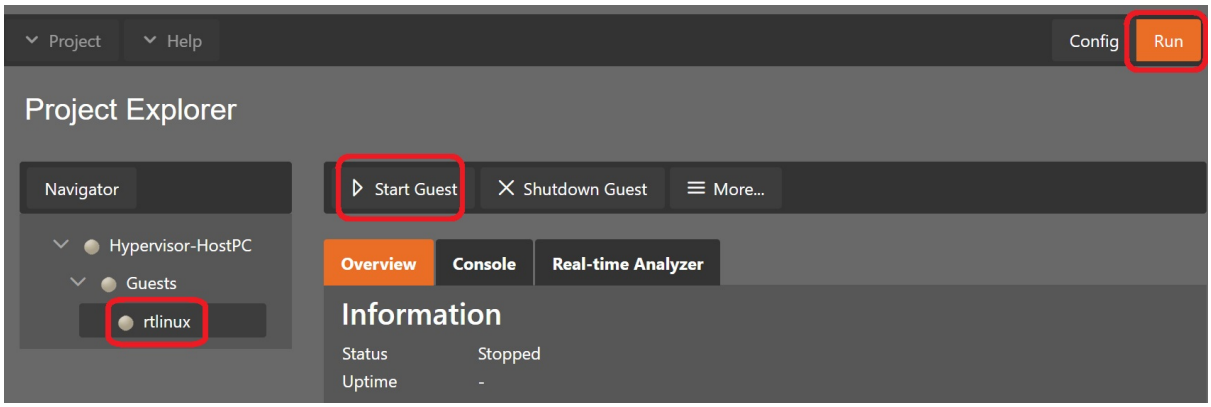


In that case, switch back to *Config* mode, select the Hypervisor Host and press the *Reboot* button.

After rebooting, you need to refresh the browser, re-connect with the Hypervisor Host, synchronize, switch into *Run* mode and try to start the VMF again.

5.3 Start a RTOS guest

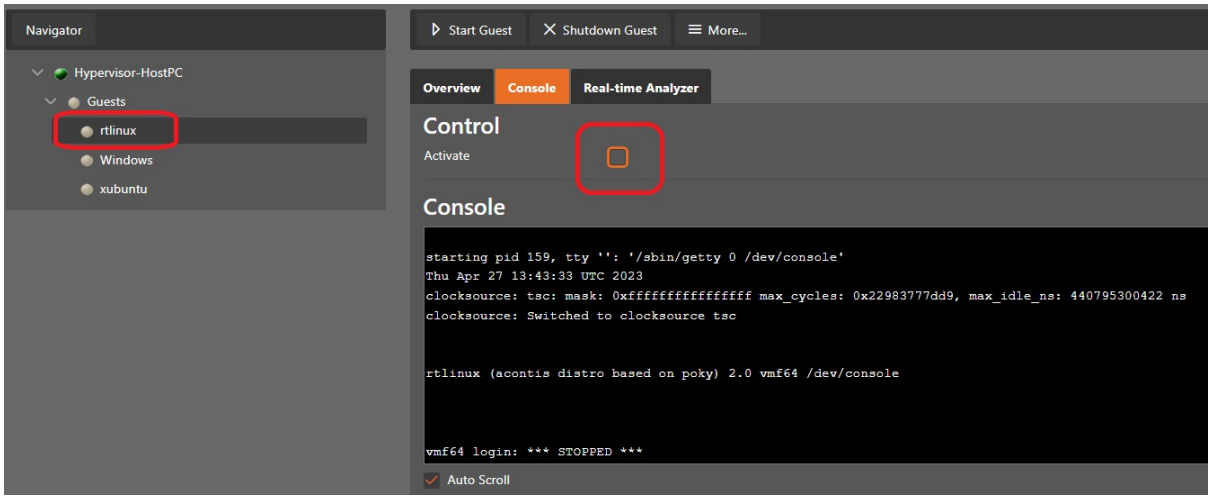
Select the guest and press the *Start Guest* button.



After you started the guest, you need to wait for some time until finally the status indicators should have switched to green.

Select the guest in the tree view and then switch into the *Console* tab in the right part. You will see the boot messages of the RTOS and finally, if supported by the guest, you will be able to log in into its shell.

If you prefer to use a local terminal to work with the guest console, you must de-activate the console **before** starting the guest.

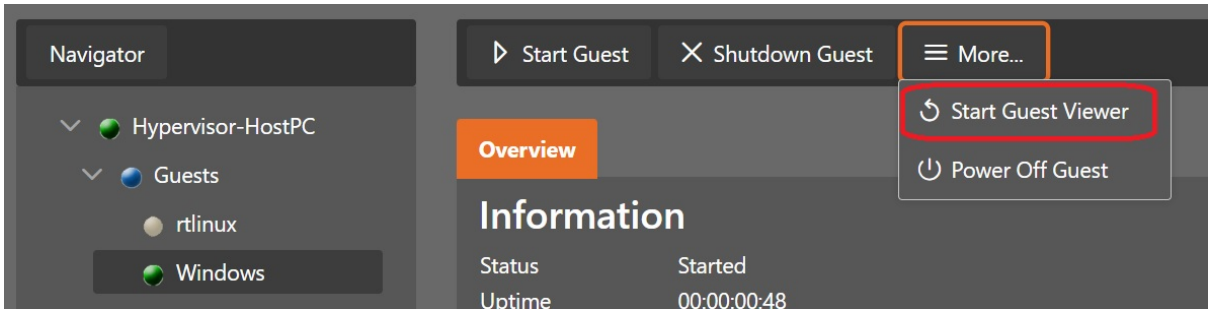


5.4 Start a General Purpose guest

Select the guest and press the *Start Guest* button.

After you started the guest, you need to wait for some time until finally the status indicators should have switched to green.

Select the guest in the tree view and then start the guest viewer via the *More...* combo-box.



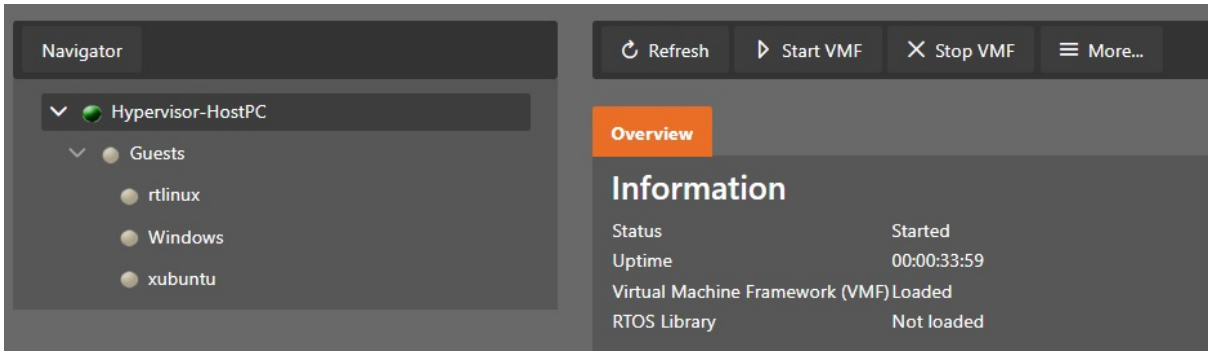
In case you get an error *No Display device defined, please login via the graphical desktop and run the command: hv_sysmgr restart*, then you need to start a terminal in the Hypervisor Host (**not** in a remote logged in shell) and run the *hv_sysmgr restart* command. After running this command you need to refresh the browser and reconnect again.

Hint: You can launch the viewer also from a shell console in the Hypervisor Host. You need to switch into the respective guest folder, in the example below it is `/hv/guests/guest0002`.

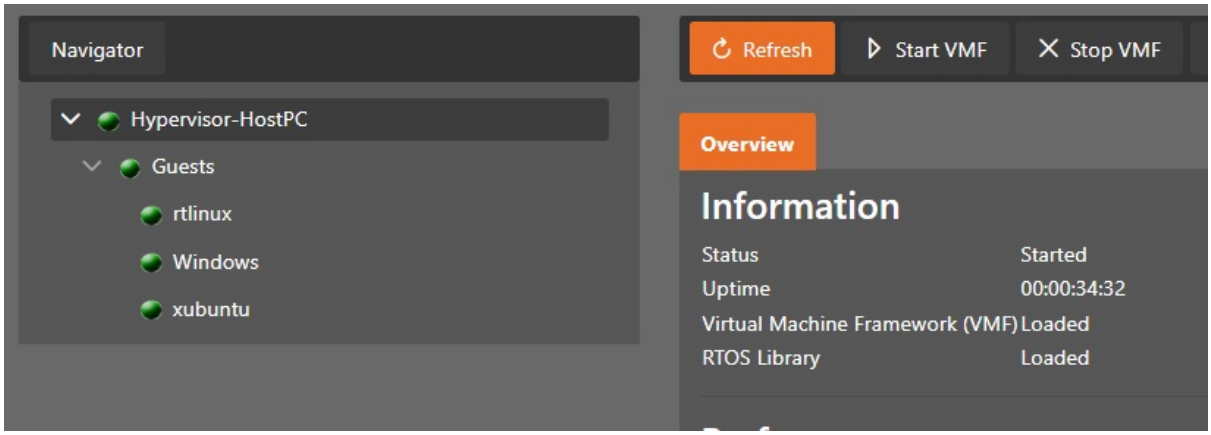
```
cd /hv/guests/guest0002
hv_guest_console
```

5.5 Refresh Guest Status

The System Manager will not always correctly show the real status of the guests.



If you want to be sure about the current guest status, press the *Refresh* button.



5.6 Shutdown an RTOS guest

Select the guest in the tree view and then press the *Shutdown Guest* button. You need to wait for some time until finally the status indicators should have switched to gray.

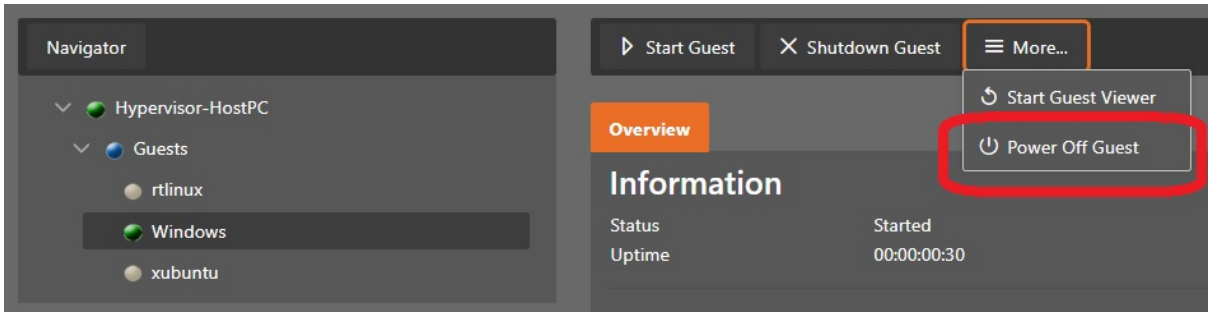
Caution: The System Manager will not accept user input until the guest finally has shutdown.

5.7 Shutdown a General Purpose guest

Select the guest in the tree view and then press the *Shutdown Guest* button.

The System Manager will then initiate the guest shutdown process. Depending on the guest or guest status, this may not work (for example, if the guest does not support ACPI shutdown messages or if the guest crashed). You may take a look into the guest console to see if the shutdown process is effective.

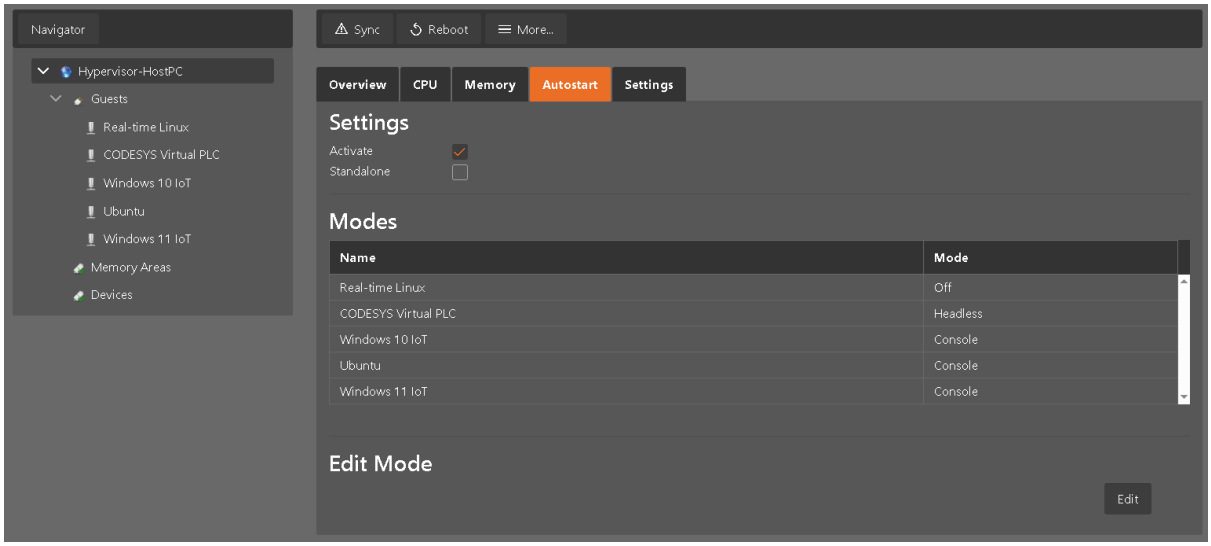
In case shutdown does not work, you may shutdown the guest manually (within the guest) or you may forcibly power off the guest.



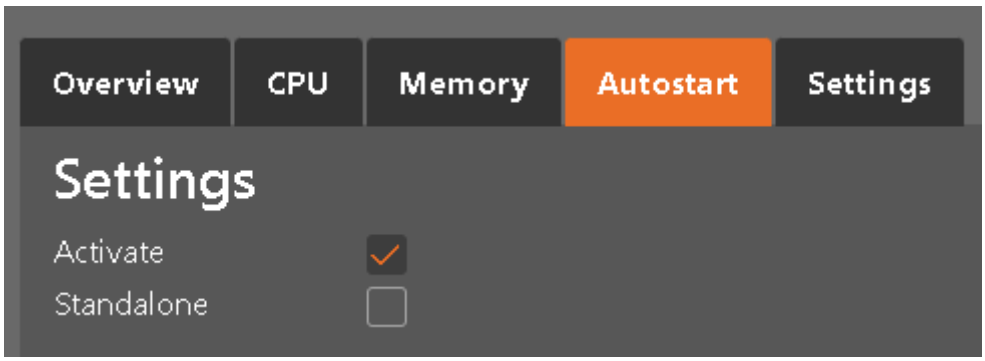
As the System Manager does not wait until the shutdown finished, you may not see the real status of the guest. You need to *refresh* to update the guest status manually.

6 Guest Autostart

You can use the System Manager to configure the Hypervisor to automatically start guests after powering on the system. Select the Hypervisor PC and switch to the *Autostart* tab:



By default, no guest will be started automatically and autostart is disabled. You need to check the *Activate* checkbox to activate autostart.



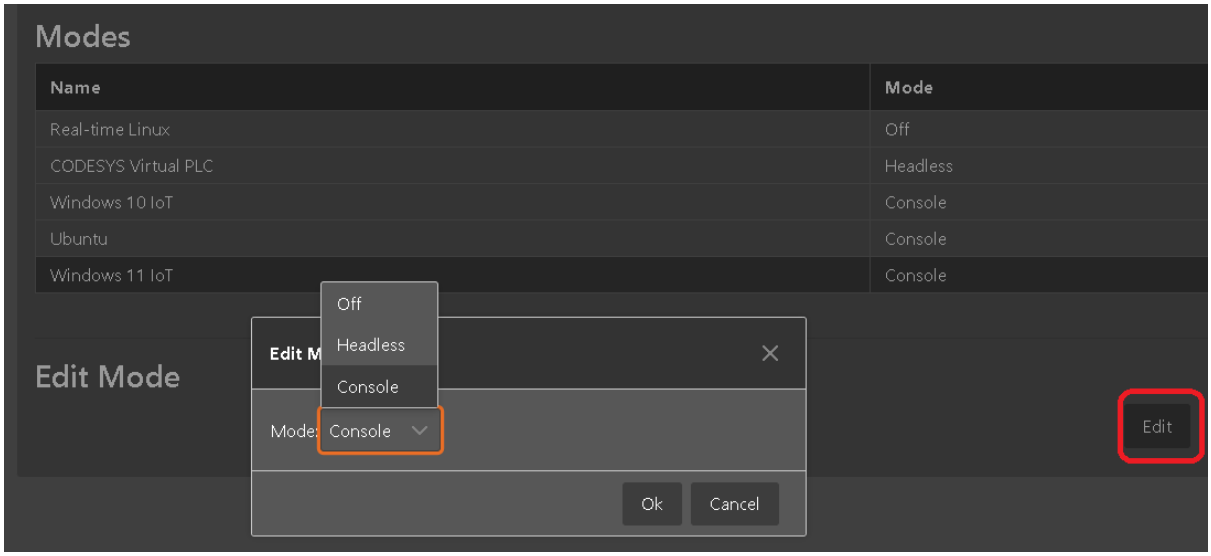
6.1 Multiple Consoles

If the Hypervisor has enabled its graphical user interface (usually for in house and debugging scenarios only), you can use a viewer/console for each of the guests that is started.

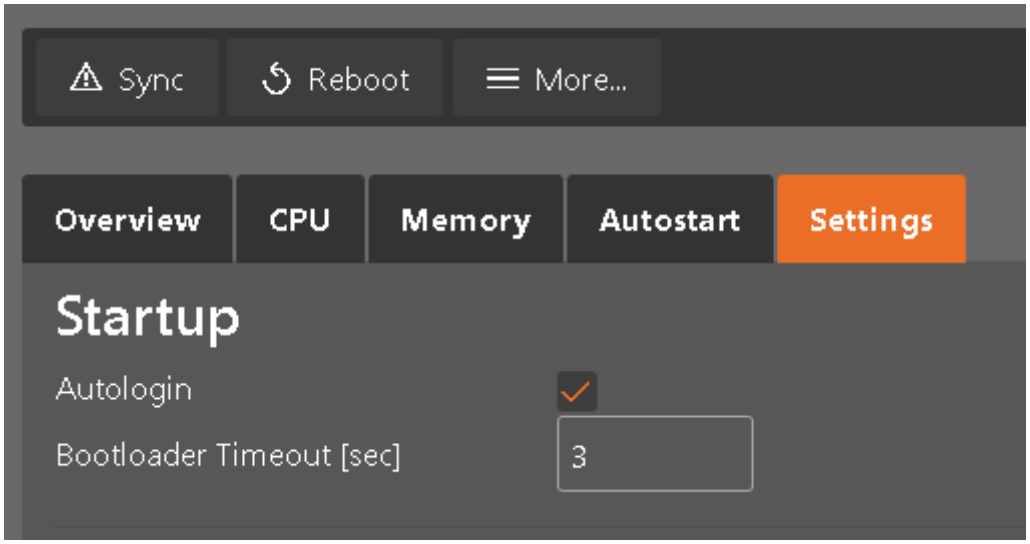
For each guests, there are 3 modes:

- *Off*: do not start guest automatically
- *Headless*: start guest without showing its viewer/console
- *Console*: start guest and show its viewer/console (e.g. the Windows desktop)

Select the appropriate guest, press the *Edit* button and change its mode.

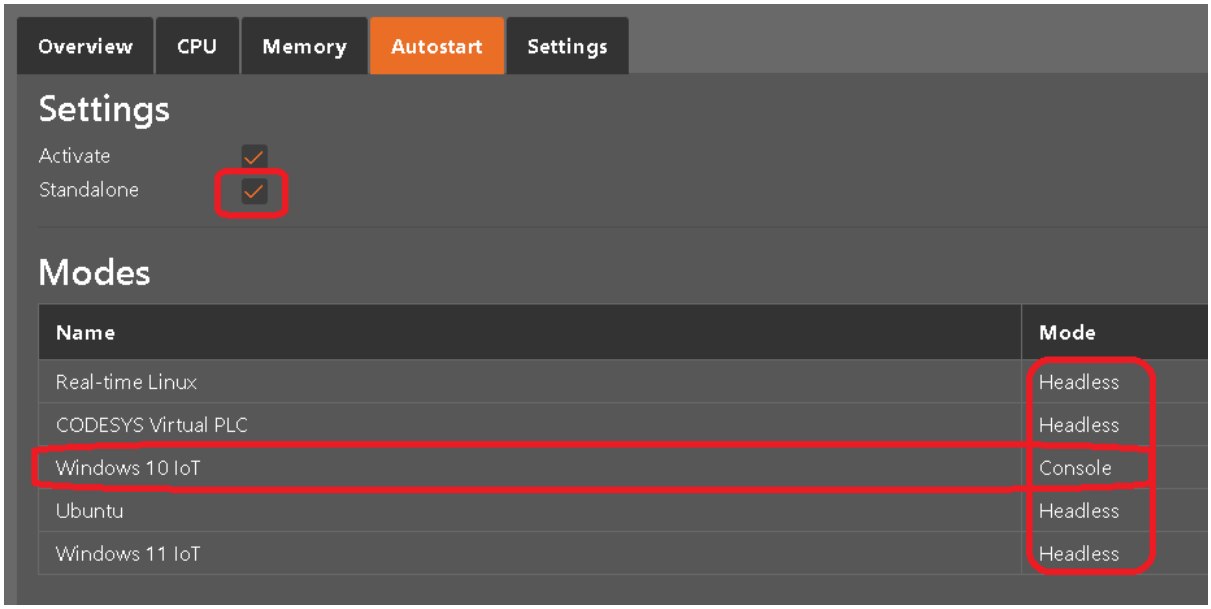


If Autostart is activated, you need to assure that *Autologin* is selected in the *Settings* tab.



6.2 Standalone Mode

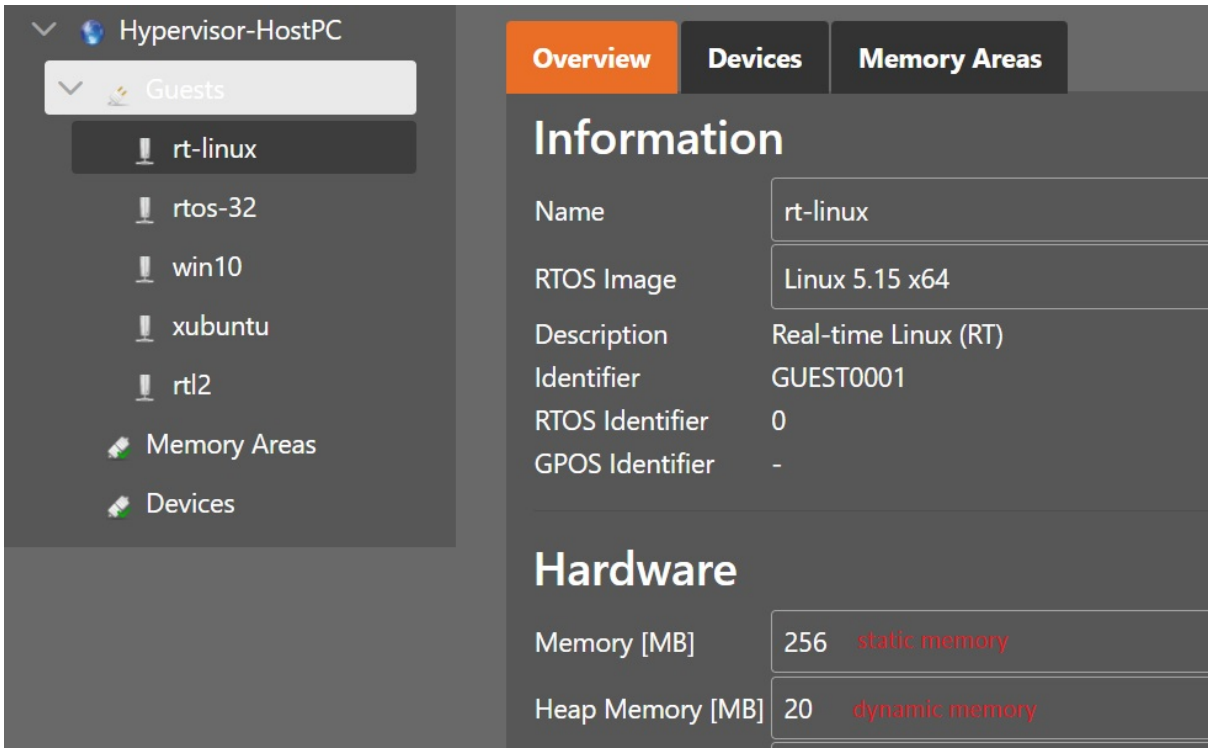
If you want one of the KVM guests to exclusively use the display (typically in production environments), you need to check the *Standalone* checkbox. In this mode, only one of the KVM guests is allowed to set to *Console* mode. This guest will then control the monitor exclusively. In Standalone Mode, the Hypervisor host GUI is also disabled.


Overview
CPU
Memory
Autostart
Settings
Settings
Activate
Standalone
Modes
Name
Mode
Real-time Linux
Headless
CODESYS Virtual PLC
Headless
Windows 10 IoT
Console
Ubuntu
Headless
Windows 11 IoT
Headless

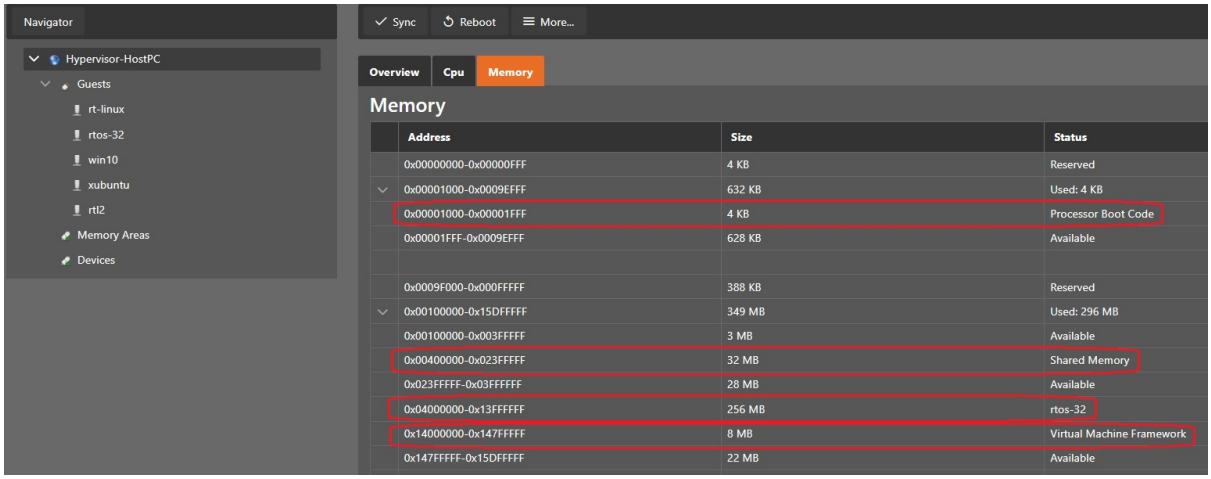
7 Guest Memory

7.1 Memory for RTOS guests

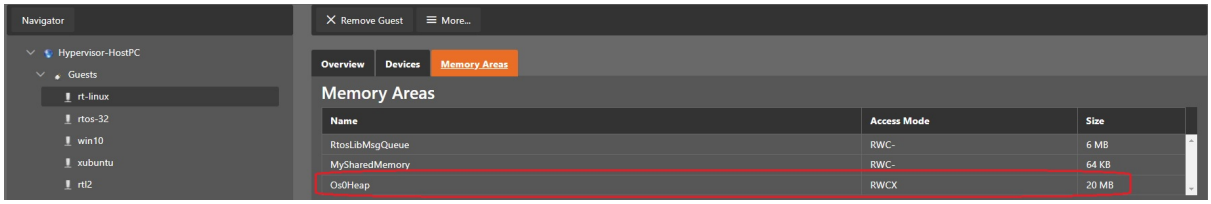
Each RTOS is assigned a specific amount of memory. Two memory sections can be configured. The first memory section is statically removed from the Hypervisor Host before booting. The second section (dynamic) is located in a shared memory area and added to the RTOS guest heap when the RTOS is booted.



How the System Manager organizes the RTOS guests (and other hypervisor components) in memory can be determined in the memory tab.



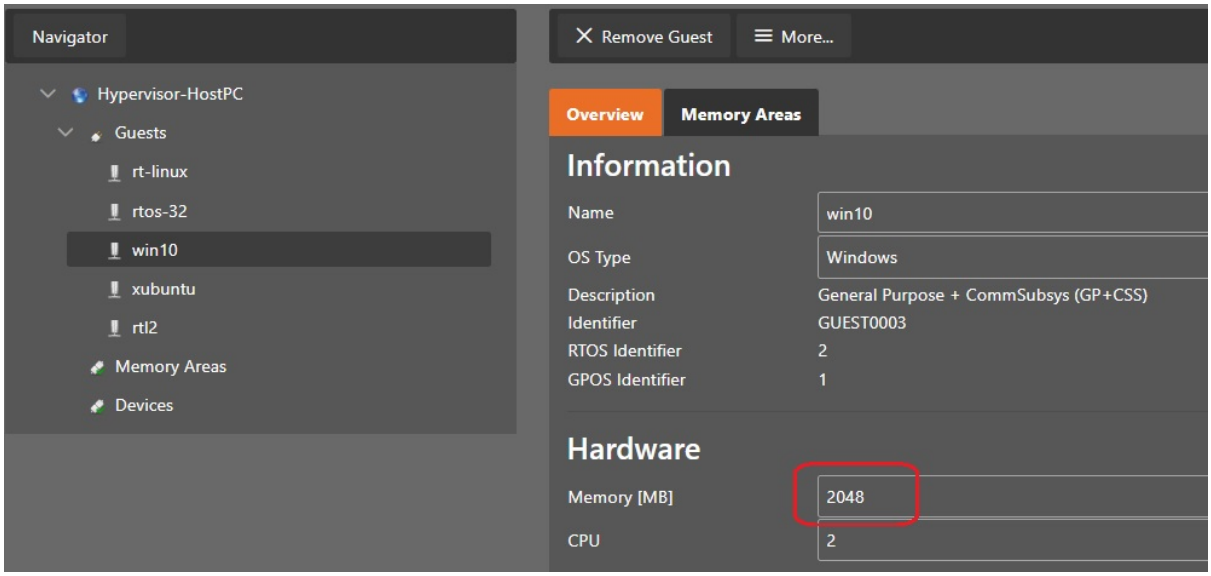
Heap memory for the RTOS is shown in the memory area tab of the respective RTOS guest.



Hint: Some RTOS guest images have restrictions, where they can be physically located in main memory. These restrictions are stored in `/hv/templates/images.xml`. The settings are explained within this xml file. For example, VxWorks or RTOS-32 guests are statically linked to fixed physical addresses. These addresses are set using the **BaseAddr** setting.

7.2 Memory for General Purpose (KVM) guests

The amount of memory which a KVM guest can use is configured in the overview section of the guest.

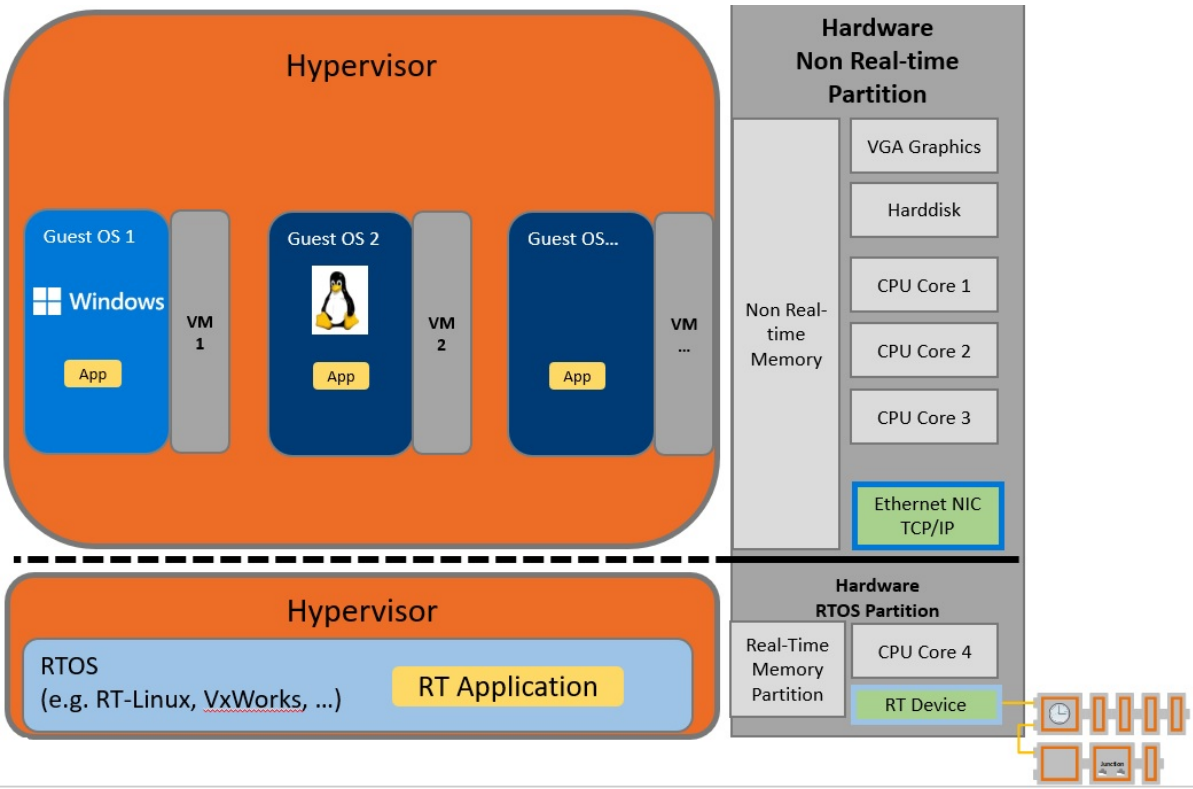


Memory for KVM guests is dynamically allocated at the time, when the guest is booted. The Hypervisor Host will require a minimum amount of memory to work correctly. To avoid unexpected behavior, the System Manager will assure that a specific amount of memory will be reserved for the Hypervisor Host. This is defined in the System Manager internal setting **RemainingHostMemory**, see also *here*.

8 Hardware Partitioning

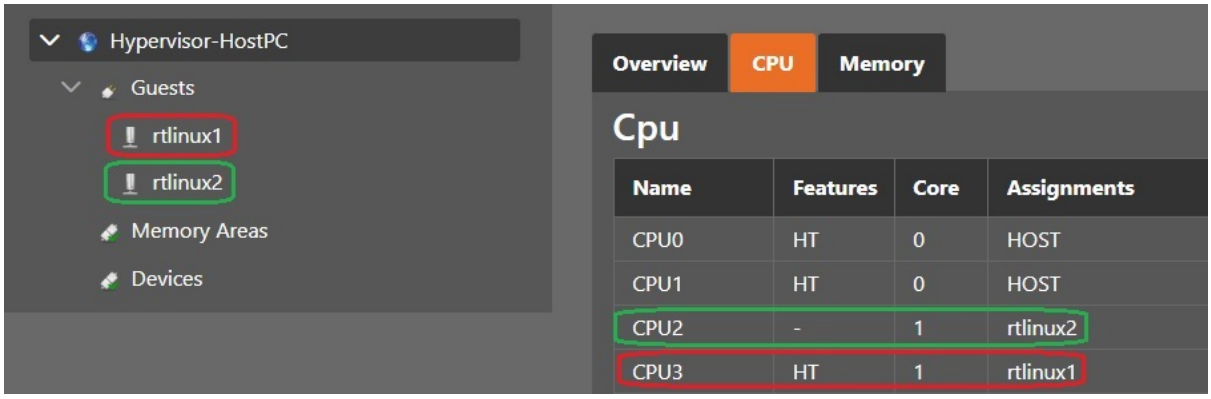
Each RTOS guest runs on a specific part of the hardware. In case one RTOS guest is running, the PC hardware is split into two partitions, one partition for all non Real-time guests and one partition for the RTOS guest. If additional RTOS guests shall be used, each of them needs to get its own partition. An RTOS partition consists of

- one of multiple dedicated CPU Cores where the RTOS is executed, no other guest is running on this CPU core.
- dedicated memory for the RTOS, see also *here*.
- dedicated devices to be exclusively used by the RTOS.



8.1 RTOS CPU Partitioning

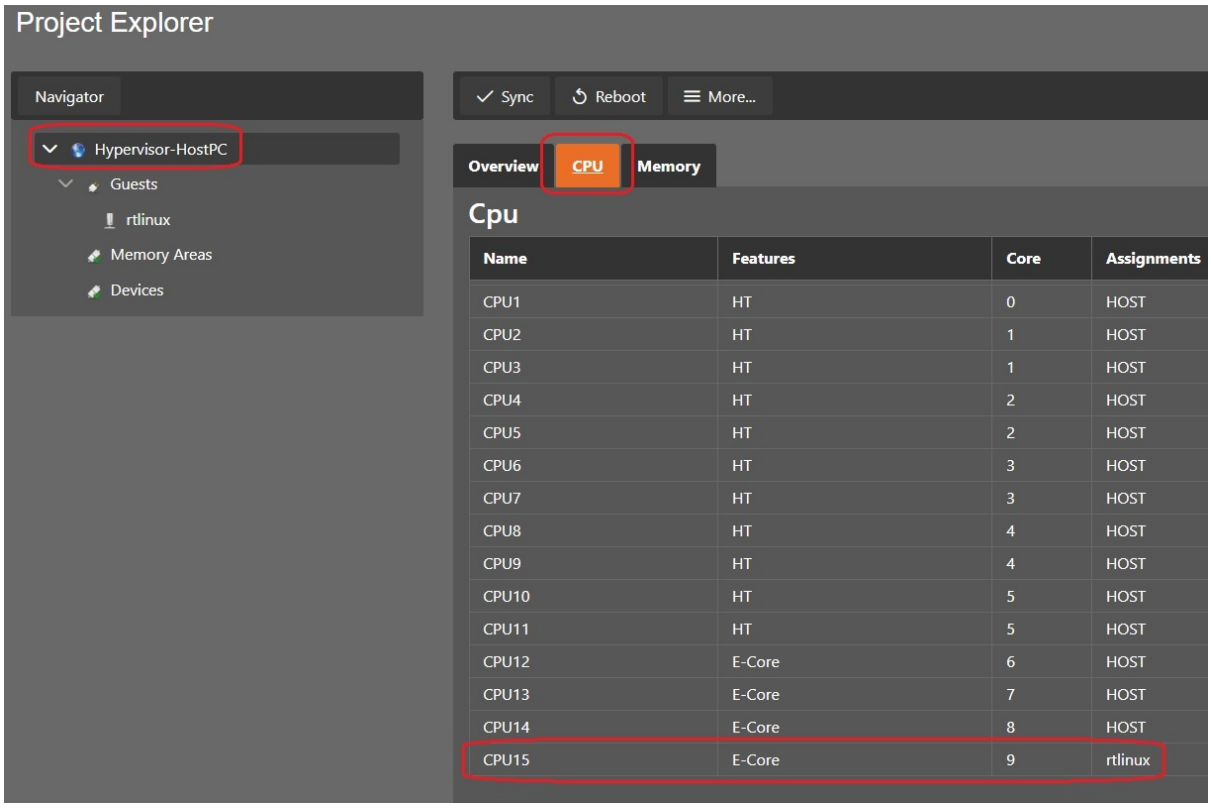
By default, the first RTOS which is added will use the last CPU core of the system, for example the fourth CPU Core (CPU3) in a quad-core CPU. The second RTOS then will use the next lower CPU, in a quad-core system it would be CPU2.



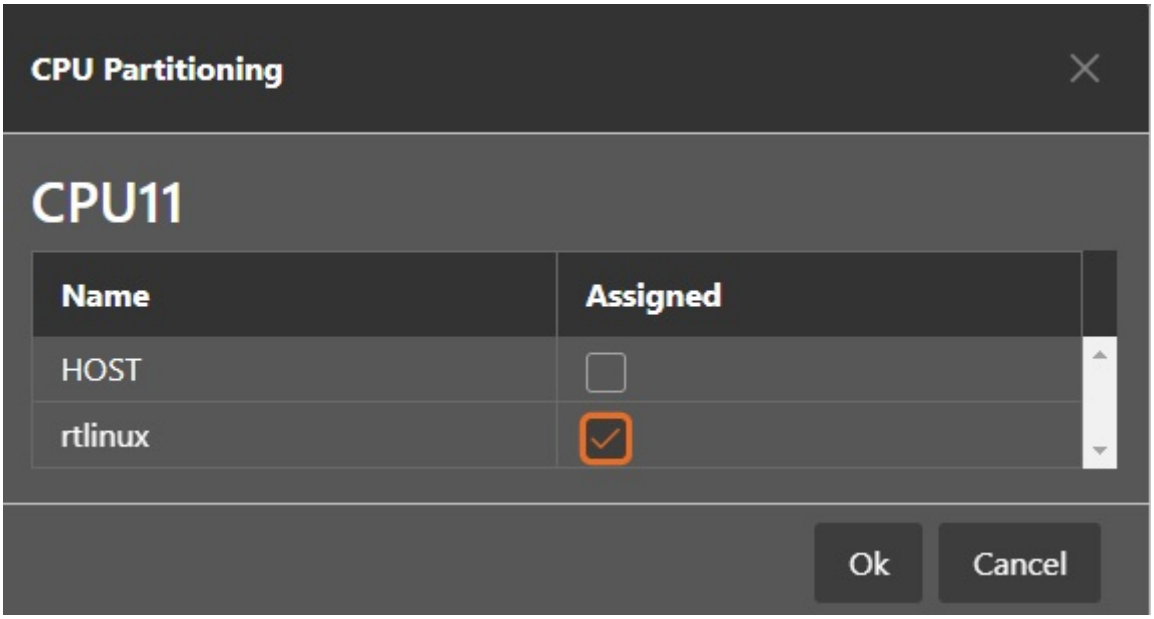
8.1.1 Dynamic RTOS CPU Partitioning

In some cases, you may need to run an RTOS guest on a different CPU Core than automatically set. For example, if the CPU provides E-Core and P-Core CPUs, you may want to run the RTOS on an E-Core CPU.

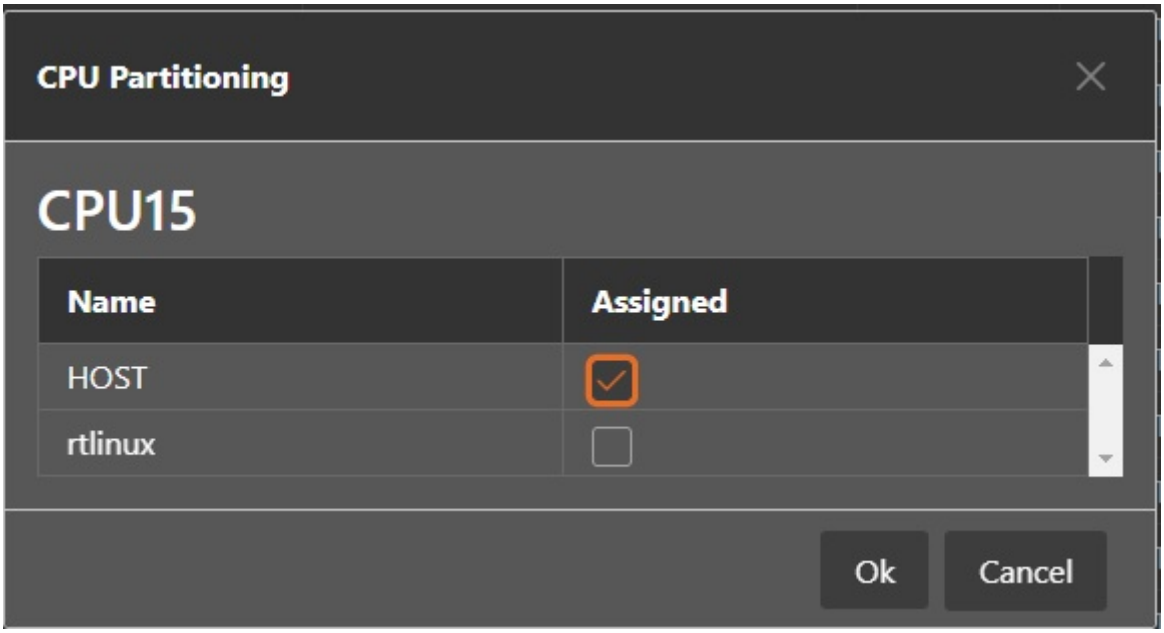
To dynamically adjust the CPU, you need to select the *CPU* tab in the Hypervisor configuration. The following screenshot shows the default configuration for a RT-Linux guest on the last CPU core of a 16 core CPU (CPU15).



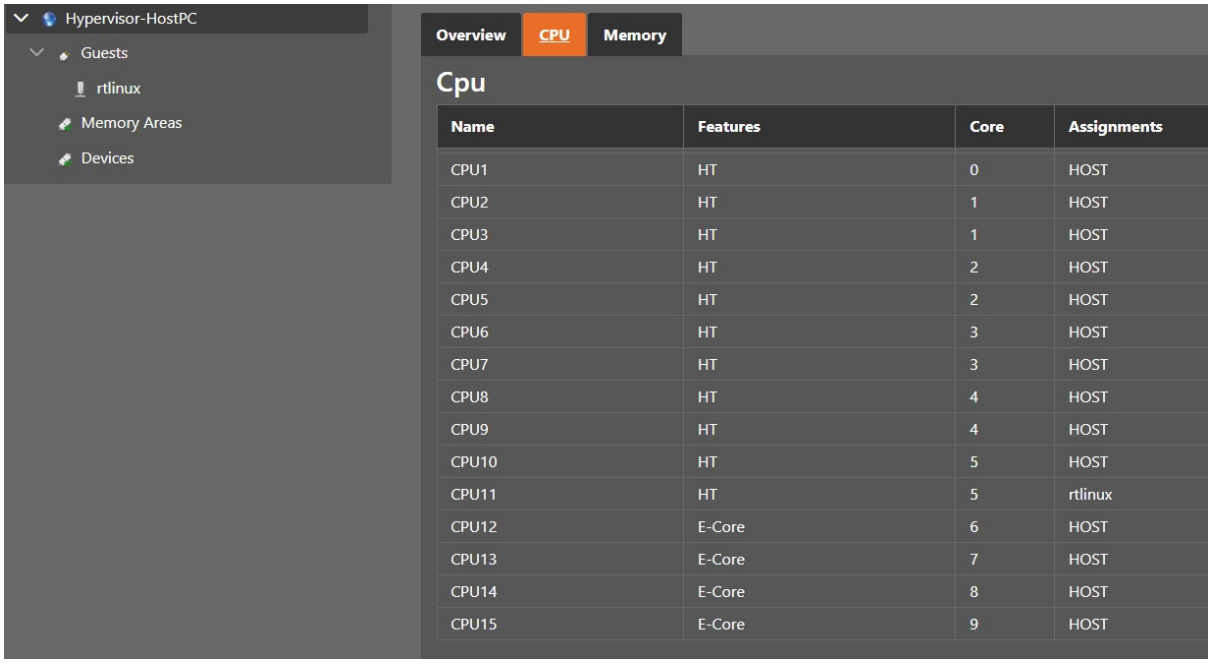
If you want to change the CPU core from CPU15 to CPU11 (a hyper threaded P-Core CPU), then select CPU11 in the CPU list and press the Edit button. Then unselect the HOST checkbox and select the rtlinux checkbox.



On CPU15 you also need to unselect the rtlinux checkbox and select the HOST checkbox.



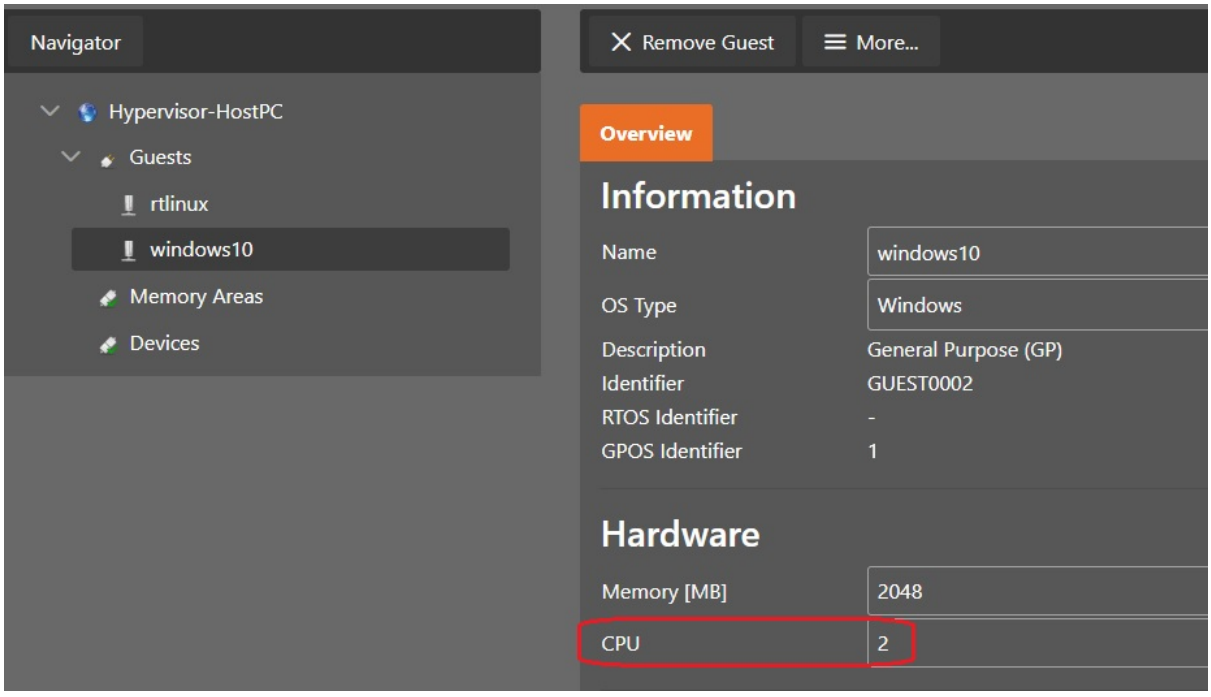
The result should then look as follows:



Caution: If additional RTOS guests are added, the System Manager may not automatically assign CPU cores to these RTOS guests, you need to manually assign CPU cores then.

8.2 KVM Guest CPU Usage

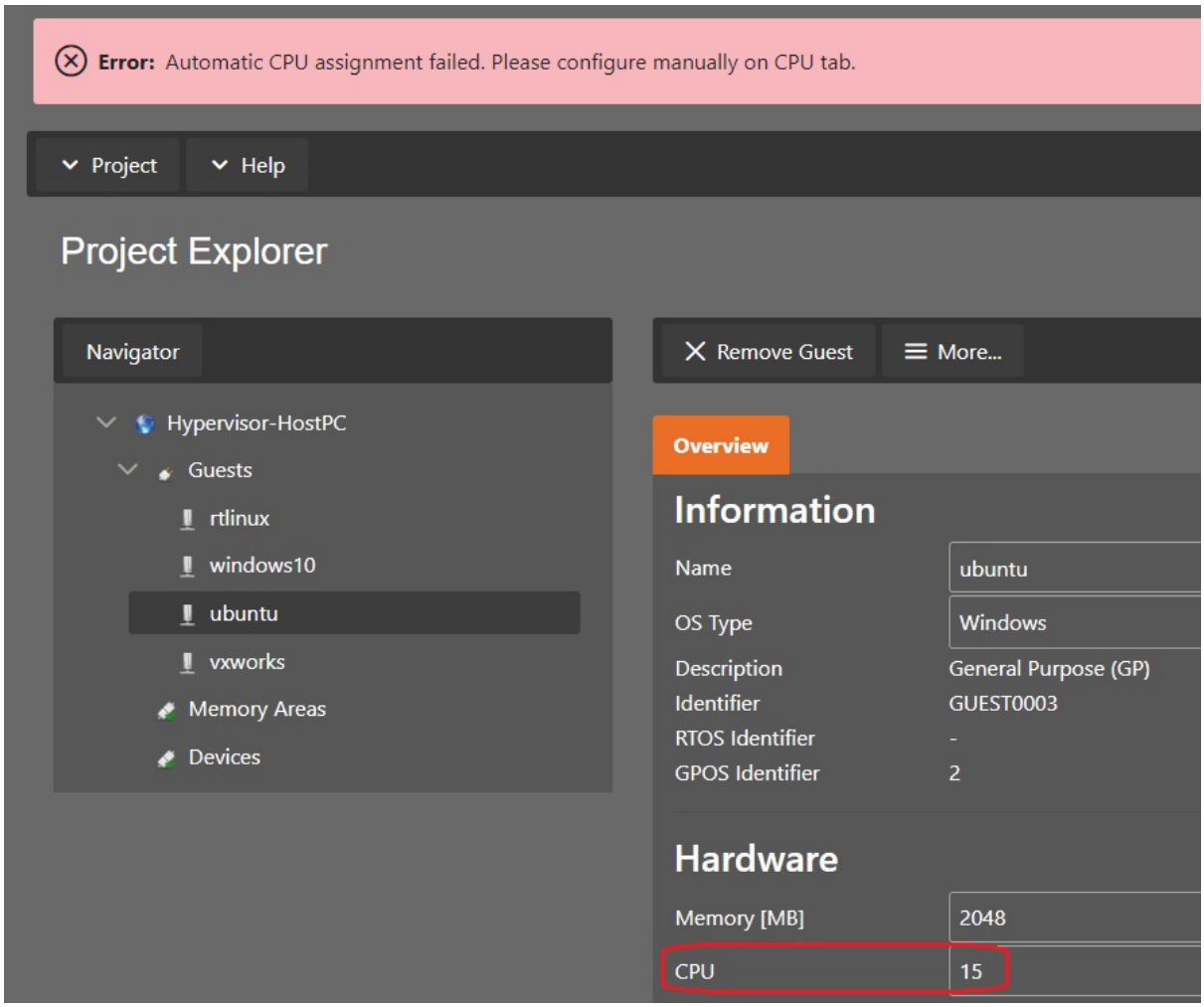
The number of CPUs that shall be used by a KVM guest, is determined in the Overview page for the guest.



The maximum number of CPUs that can be configured for a KVM guest is limited by the number of total

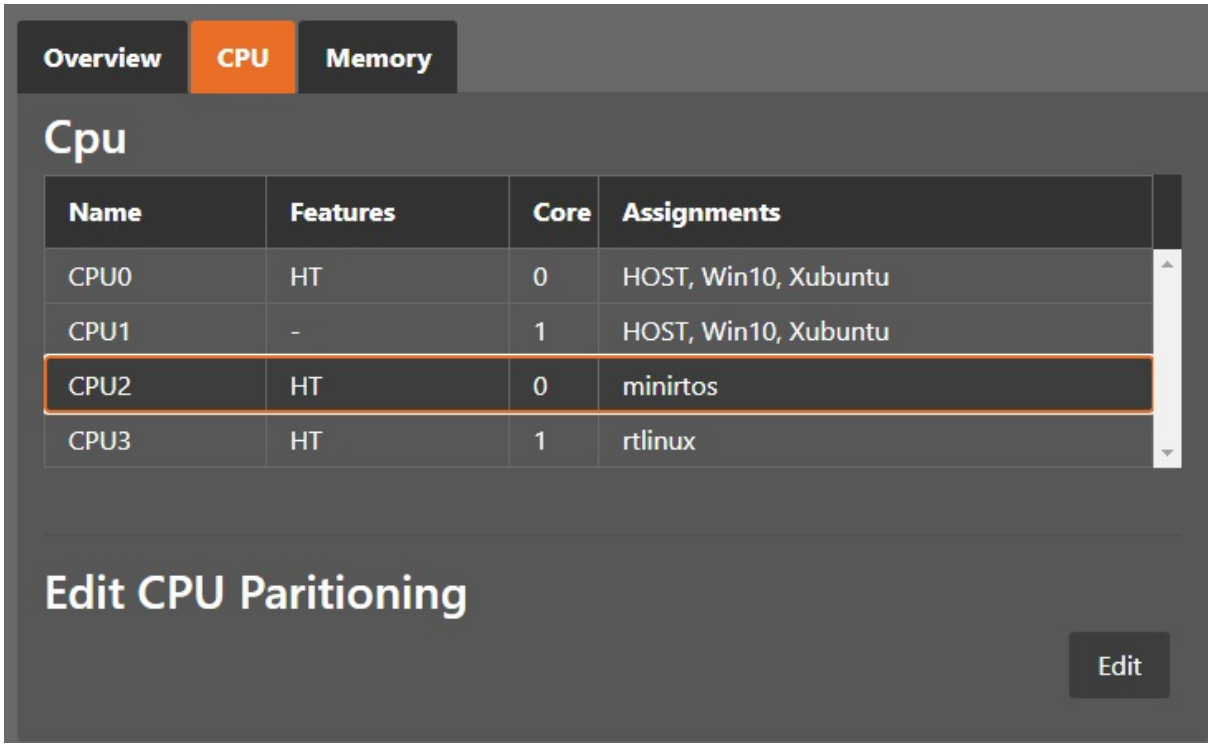
CPU cores available and the number of CPU cores configured for RTOS guests. For example, if the PC in total has 16 CPU cores and 2 RTOS guests with in total 2 CPU cores are configured, only 14 CPU cores can be used by the KVM guest.

If more CPU Cores are configured, then the System Manager will show an error message.

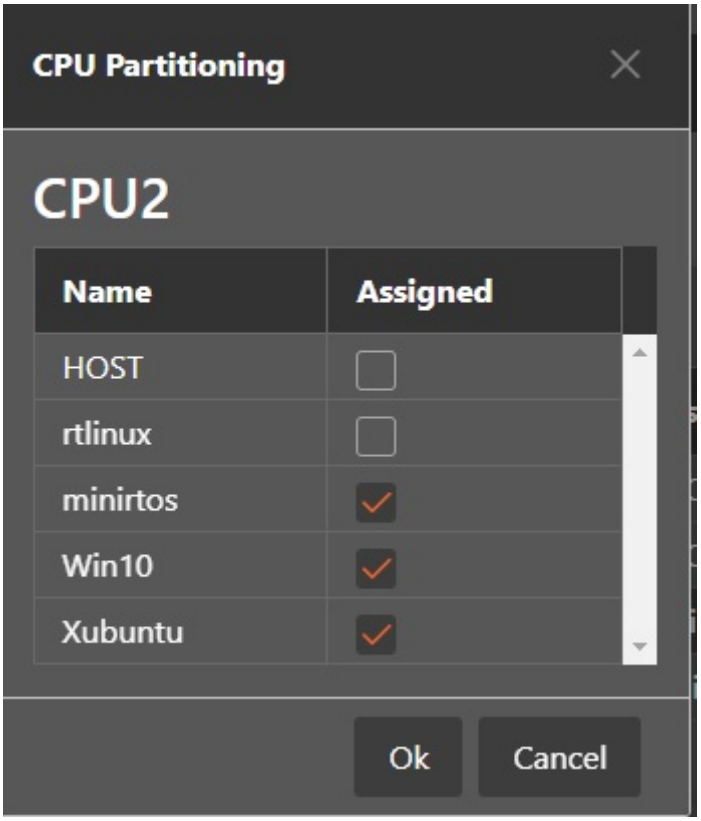


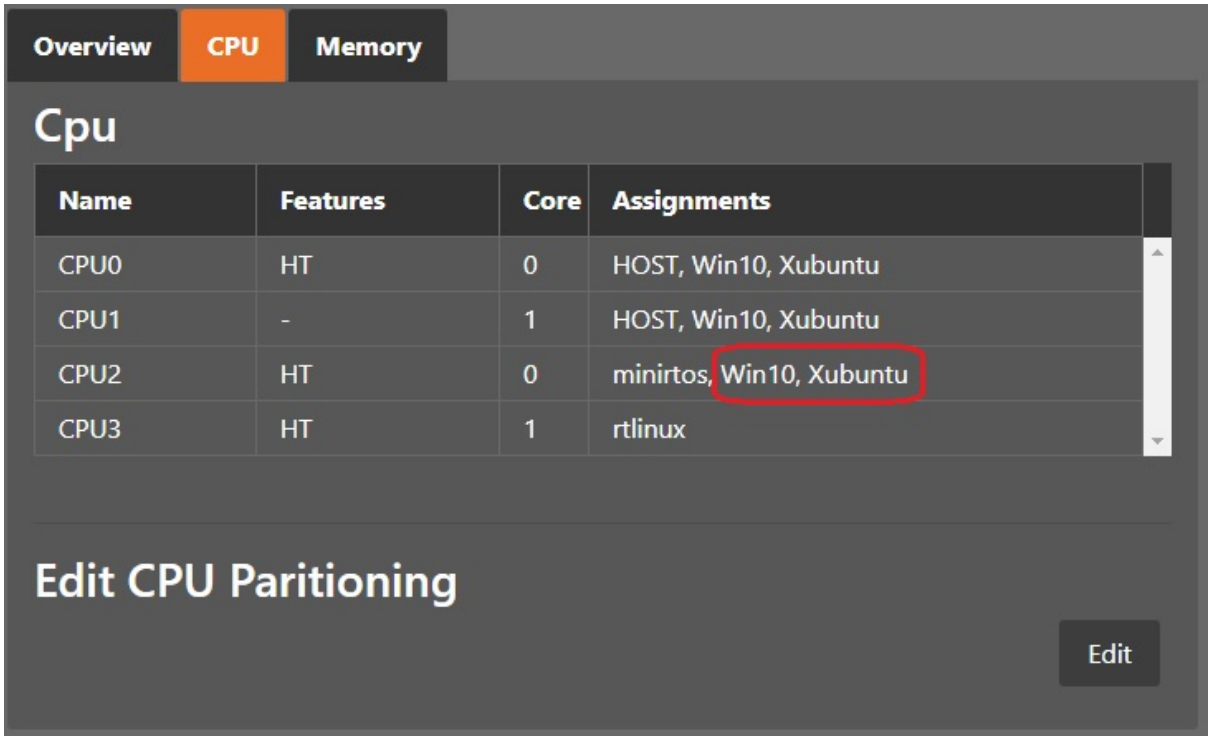
You may then use the CPU tab in the Hypervisor configuration and manually assign additional CPU cores to the KVM guest by virtually utilizing the RTOS guest CPUs for KVM guests as well.

In a first step, select the CPU on which the KVM guest currently is not configured to run:

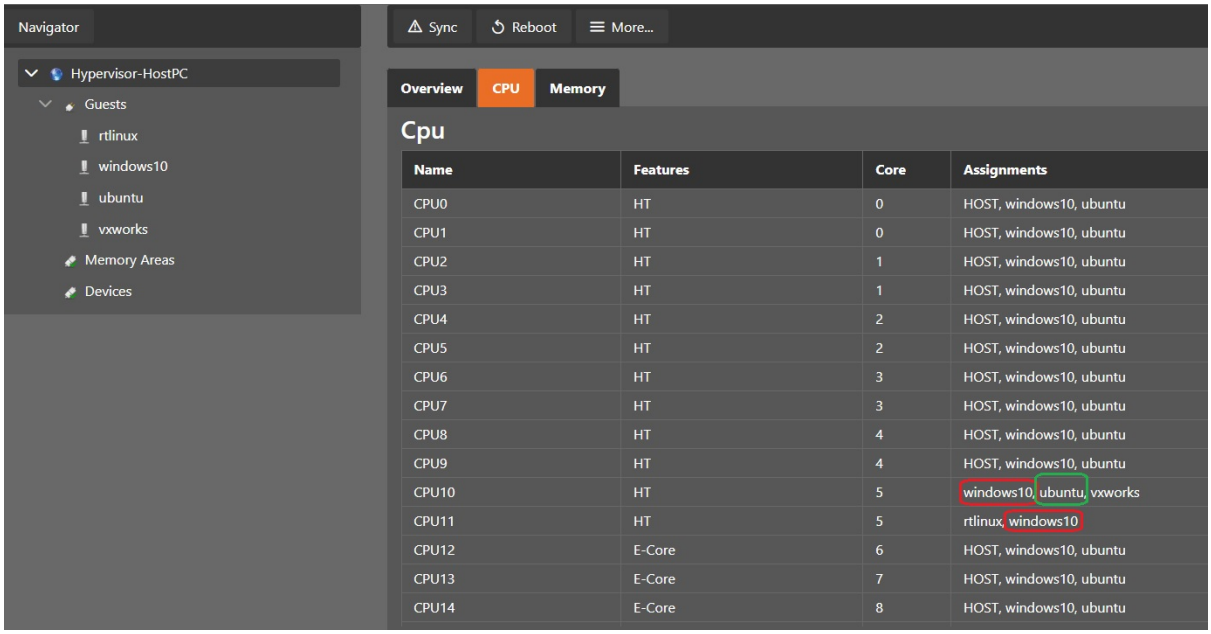


Then assign the desired guest(s) to the selected CPU:





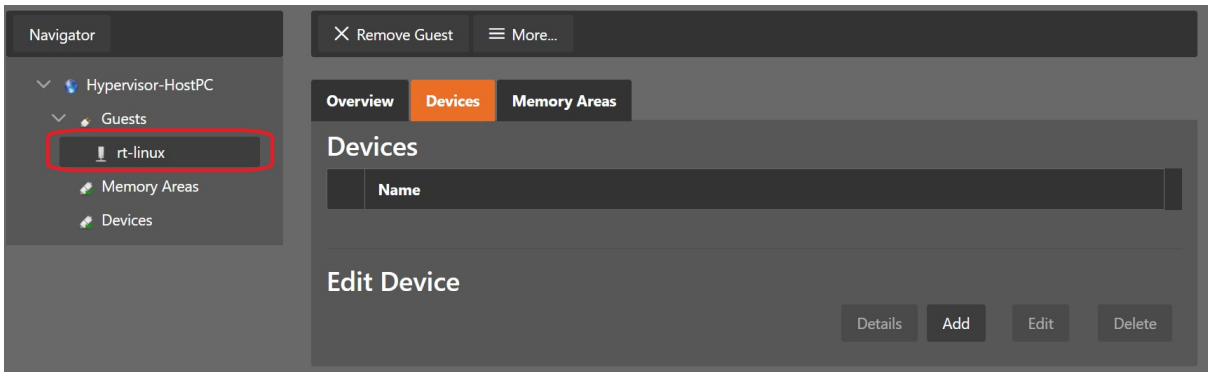
In the below example, 2 additional virtual CPU cores are added for a Windows 10 guest and one additional CPU core is added to an Ubuntu guest.



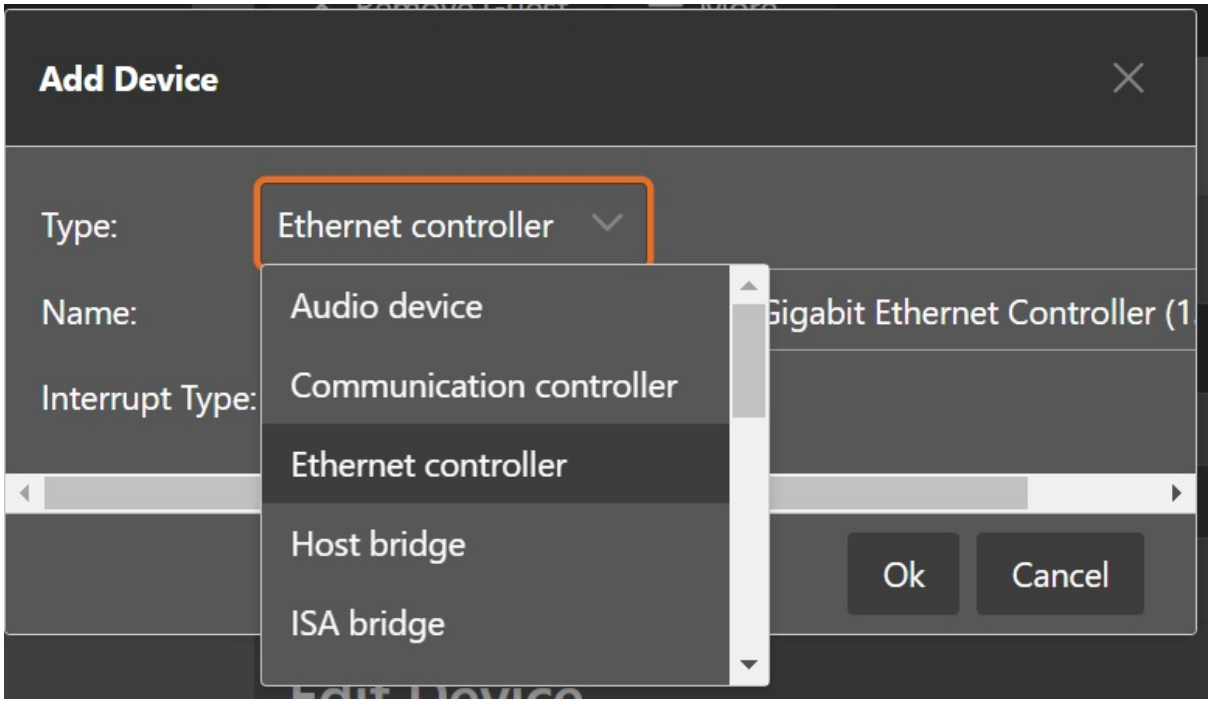
Caution: You may **configure** multiple KVM guests, each such guest may use CPU cores up to the total number of CPU cores available. The maximum number of CPU cores utilized by **running** KVM guests is limited to the total number of CPU cores as well. For example, you may configure 2 KVM guests with each 16 CPU cores on a 16 CPU core system but you only can run one of these guests.

8.3 Assign a Device to an RTOS guest

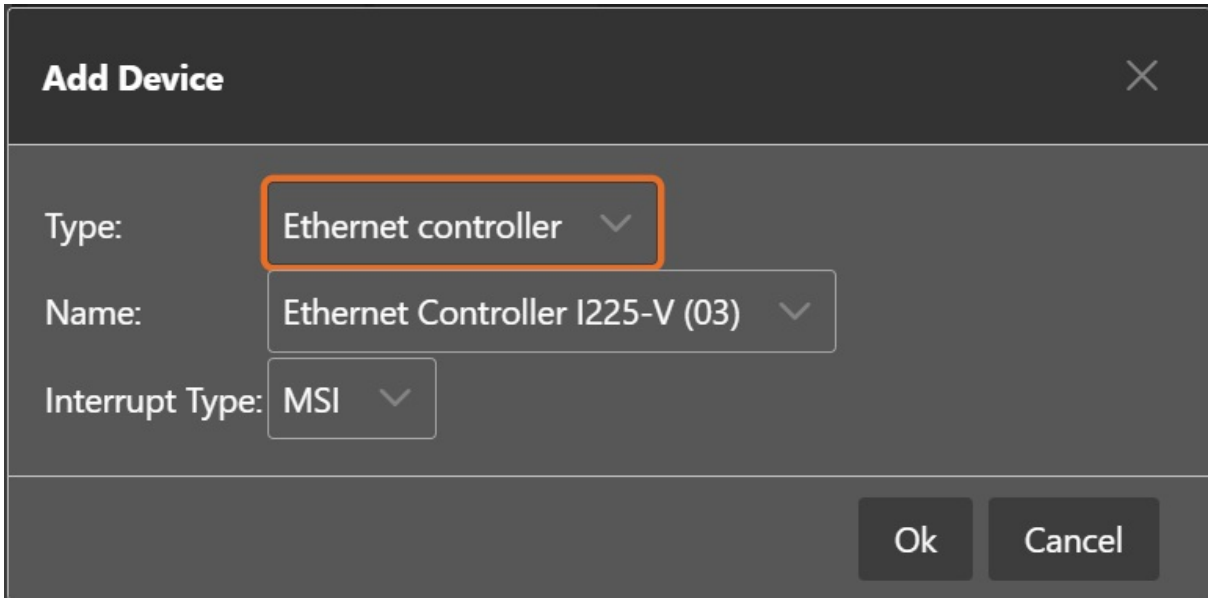
Select the RTOS where a specific device should be assigned and switch to the `Devices` tab.



Select `Add` and the device type you want to assign.



Then select the appropriate device, for example an Ethernet adapter.



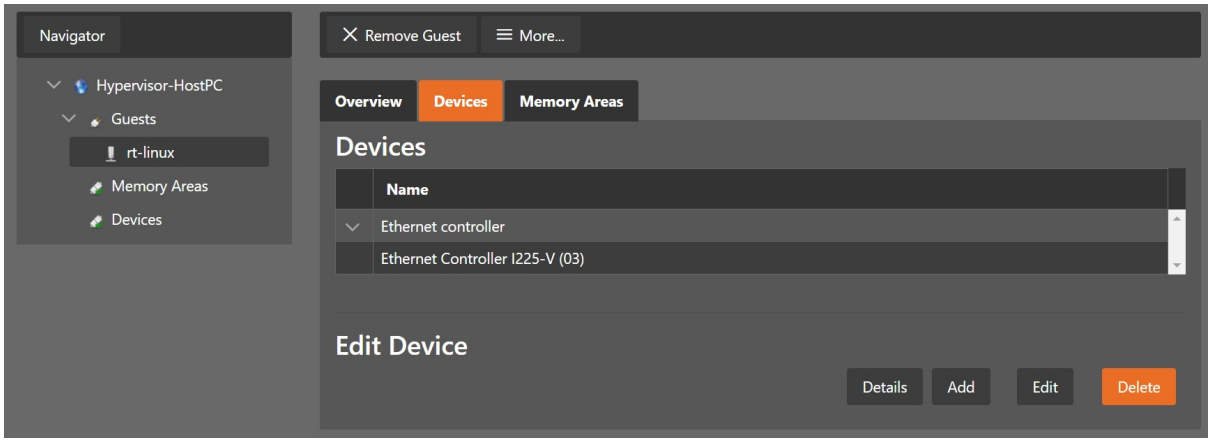
Caution: If you assign an Ethernet adapter to an RTOS guest, you need to assure not to accidentally assign the adapter you are remotely connected with the Browser!

In case your hardware has multiple Ethernet adapters, you may wonder how to find the appropriate one you want to use for the RTOS guest. You can find a detailed description of how to assign one of multiple Ethernet adapter *here*.

Hint: For each device which is assigned to an RTOS guest, two files for example `/hv/config/sysmgr_rtos_dev1.sh` (for assignment) and `/hv/config/sysmgr_rtos_dev1.config` (for configuration) are created. The assignment script file (`sysmgr_rtos_dev1.sh`) is executed within `/hv/bin/hvpart.sh` and `/hv/bin/sysmgr_hvpart.sh` when the system boots, it will assign the device to the guest. The configuration file `sysmgr_rtos_dev1.config` will be included in the respective guest configuration (e.g. in `/hv/guests/guest0001/guest.config`) which is part of the overall system configuration.

8.4 Remove a Device from an RTOS guest

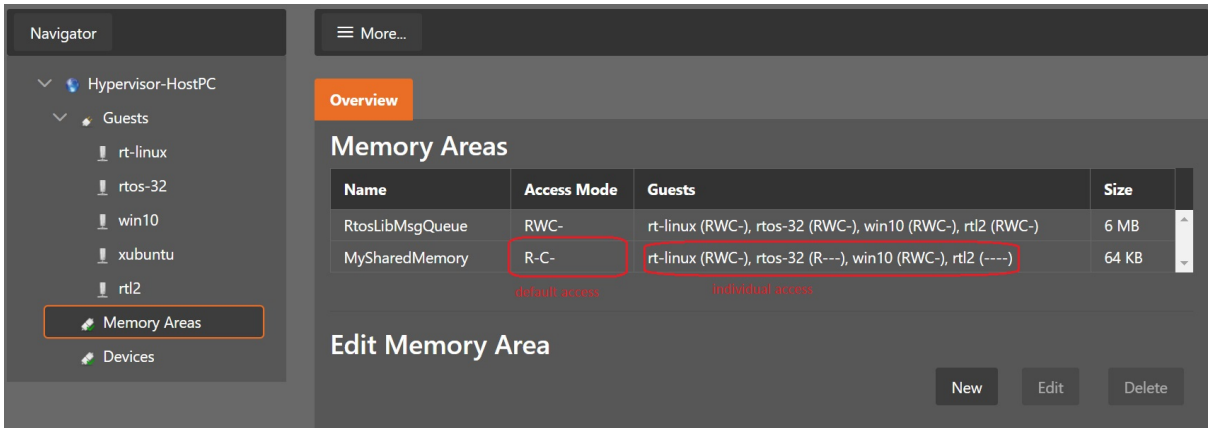
To remove the device, you may simply delete it in the guest Devices tab.

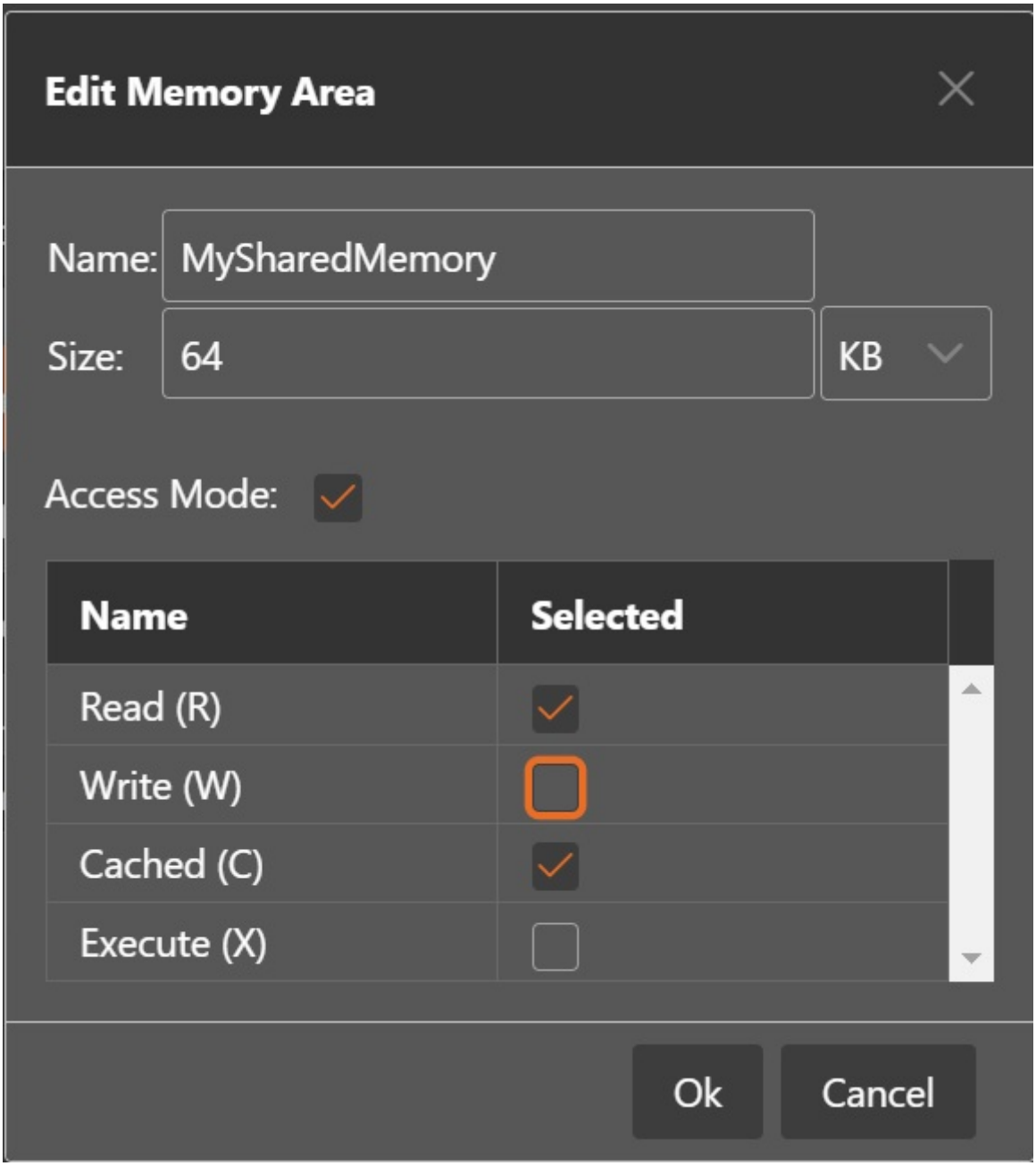


If a guest is removed from the configuration, its devices are automatically removed from this guest as well.

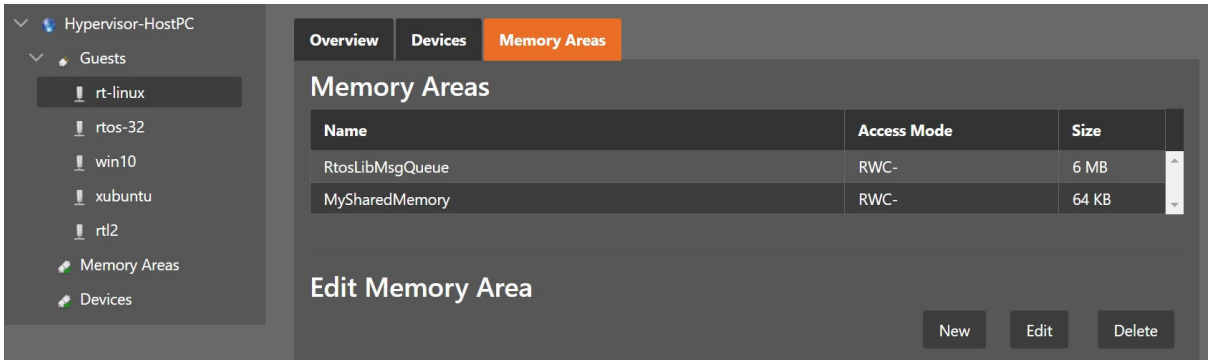
9 Memory Areas

Memory Areas (shared memory) can be used to exchange data between guests. One pre-defined memory area **RtosLibMsgQueue** is used internally for message queues. You must not delete this memory area, otherwise the system may not behave correctly. It is possible to restrict access for specific guests. In the global overview tab you can define the default access rights for all guests.



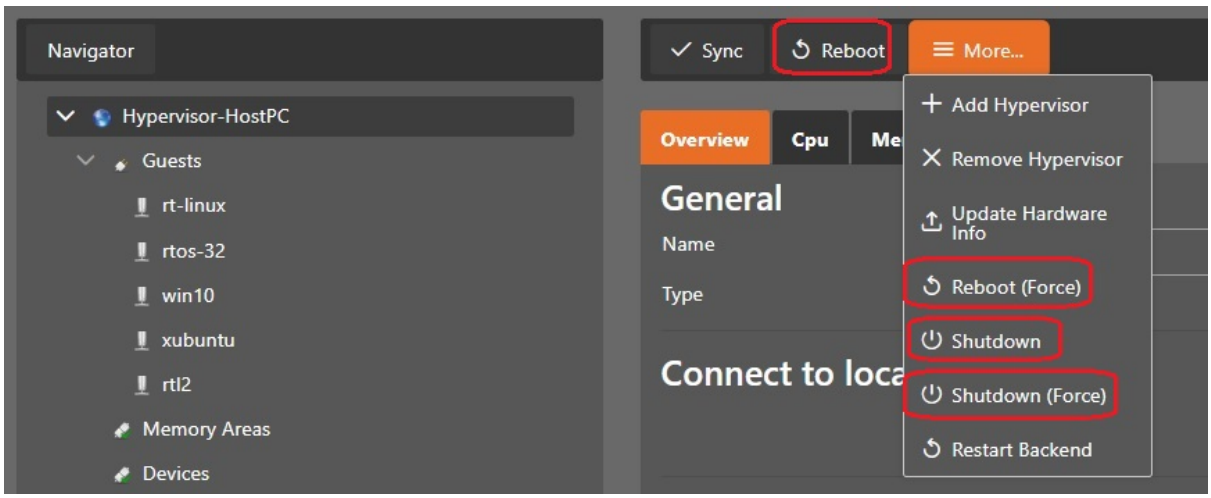


You can override the default access rights in the guest section.



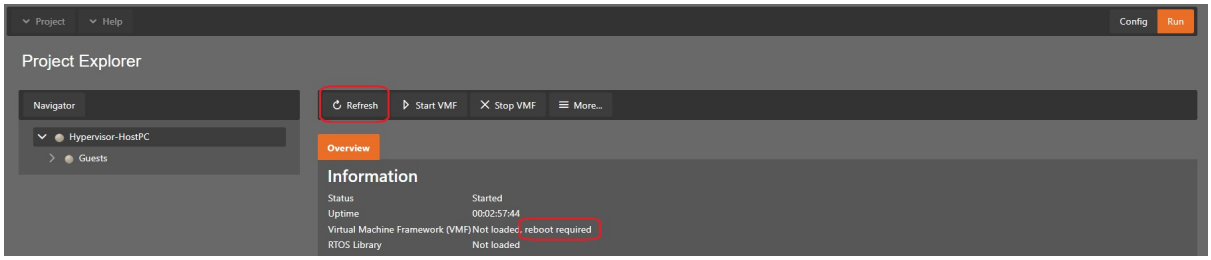
10 Hypervisor Host Control

It is possible to reboot or shutdown the Hypervisor Host from within the System Manager.

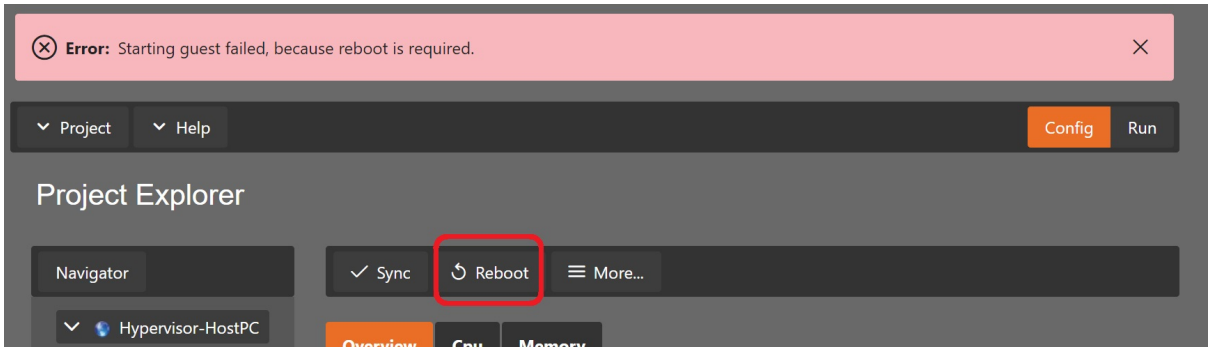


Rebooting is required in some cases, especially if the memory configuration is changed (when adding or removing RTOS guests). In some cases a regular reboot or shutdown is not possible (e.g. if guests are running), then you may use the **(Force)** option to ignore such restrictions.

You can determine whether a reboot is required by pressing the **Refresh** button in **Run** mode.



If you try to start a guest while a reboot is required, the System Manager will throw an error message. Then you should reboot the system.



11 Example Guests

Initially, the Hypervisor Host does not provide any example guest folders. To switch to an example guest, you must execute the corresponding initialization. For instructions on how to initialize the examples, refer to the chapter **RTOS Guests** in the Hypervisor Manual.

Follow these steps to initialize the RT-Linux example and run the guest:

```
$ hv_open_example rt-linux
$ hv_sync_example rt-linux
$ cd /hv/guests/guestrtlinux
$ hv_guest_start -view
```

Caution: When the example guests are initialized, the corresponding pre-configured System Manager projects will be loaded. This process will delete any existing guests, subject to your confirmation. You can preserve your current configuration by using the System Manager to save it before initializing an example. Ensure you also save any manually added content in the `/hv/guests/guestxxxx` folders, as these will be deleted as well.

Each example configuration is designed to operate a single RTOS. Both the Windows and Ubuntu example configurations are capable of running RT-Linux simultaneously.

12 System Manager for Windows

Caution: The System Manager for Windows is not yet available. This is preliminary documentation.

12.1 Installation

- Install LxWin to “C:\Program Files\acontis_technologies\LxWin\” (default)
- Unpack SystemManager HvWebApp_Windows-x64_V1.0.10.zip to “C:\Program Files\acontis_technologies\SysMgr\”
- Copy “C:\Program Files\acontis_technologies\SysMgr\Templates_patch\” to “C:\Program Files\acontis_technologies\”

12.2 Optional

- Copy VxWin to “C:\Program Files\acontis_technologies\VxWin\”
- Copy RTOS32Win to “C:\Program Files\acontis_technologies\RTOS32Win\”

12.3 Execution

- Start System Manager
  - C:\Program Files\acontis_technologies\SysMgr\HvSystemManager.exe
- Open in Browser
  - http://127.0.0.1:5000
- Select as Hypervisor
  - LxWin (can be used for LxWin, VxWin, RTOS32Win)
  - VxWin (for future use)
  - RTOSVisor (required for Hypvervisor remote configuration over TCP/MQTT)

12.4 Known Issues

- First start of RTLinux fails with following error code because the uploader changes the memory configuration, after reboot it will work:
  - Uploader return code: 0x00000037 (Visible, if all messages are enabled!)
- Install “Microsoft Visual C++ 2015 Redistributable Update 3 RC”
  - https://www.microsoft.com/en-US/download/details.aspx?id=52685 (vc_redist.x64.exe)

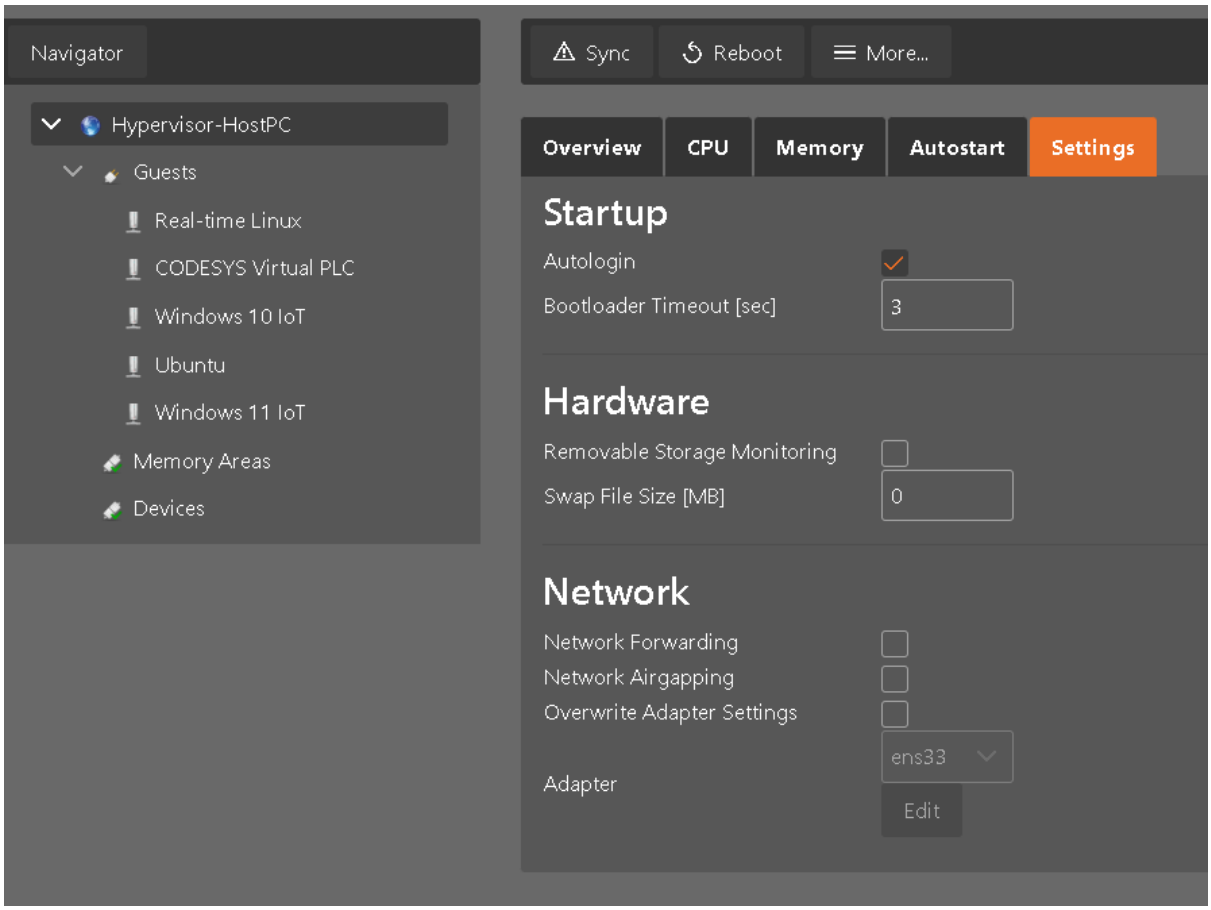
  - required by RtosLibDotNet.dll / RtosLibDotNetNative.dll

- All folders are in “C:\Program Files\acontis_technologies\” because SystemManager must be relative to LxWin, VxWin, … (like in Linux)
- Memory configuration is hard-coded (copied from trace)
  - Uploader can dump memory configuration into log with trace on
  - Uploader requires a cmdline option to dump memory configuration (This will remove the additional reboot after first start)

13 Settings

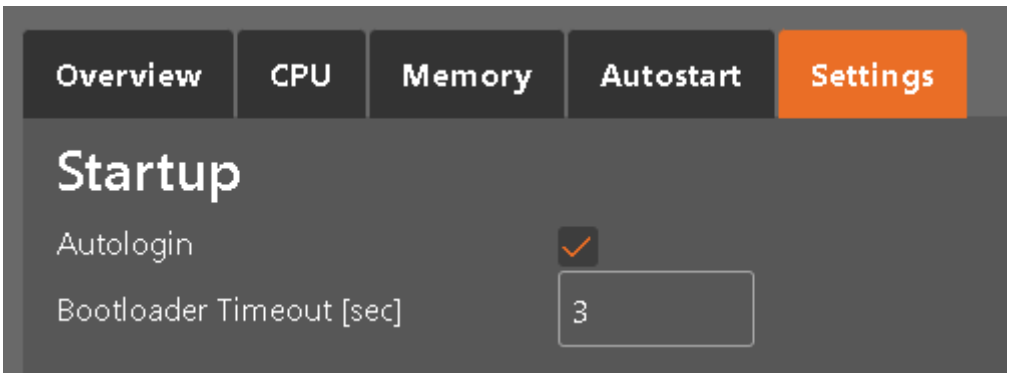
13.1 Hypervisor settings

Some of the many settings that are available in the Hypervisor can be controlled by the System Manager. You need to select the *Settings* tab to adjust such configuration settings.

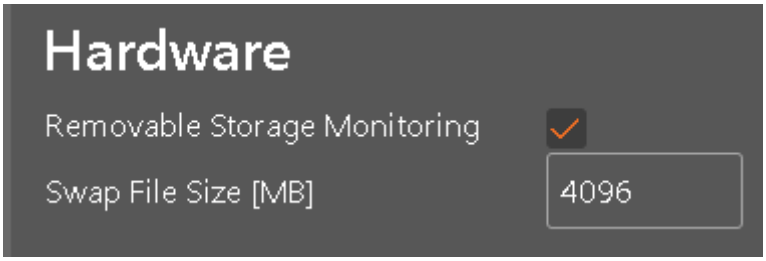


13.1.1 Hypervisor startup

The bootloader timeout as well as autologin can be controlled in the *Startup* section.



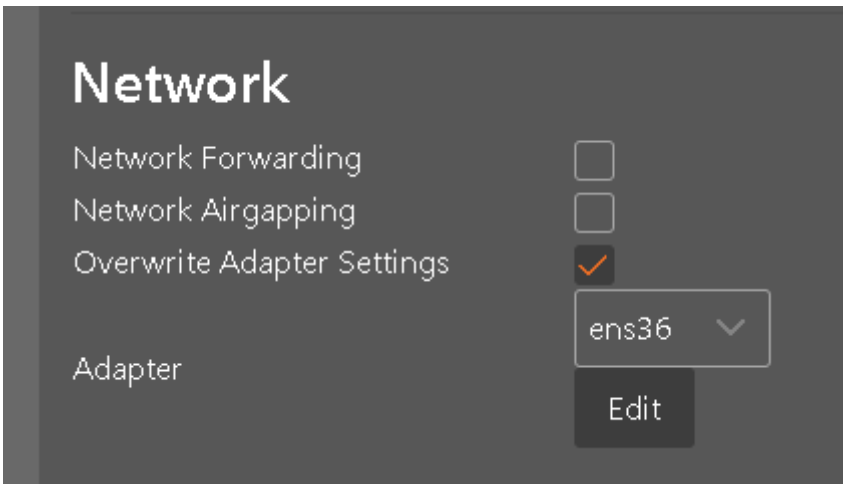
13.1.2 Hardware related settings



If *Removable Storage Monitoring* is enabled, there will be an automatic detection if storage media like USB memory sticks are connected to the PC. In such case, a share for KVM guests will be generated and guests like Windows or Linux can access such media. You may take a look into the Hypervisor manual (section *Sharing removable devices*) to get more detailled information about this feature.

The *Swap File Size* setting can be used to provide additional virtual memory for KVM guests. The default (and minimum) swap size is 1024 MByte.

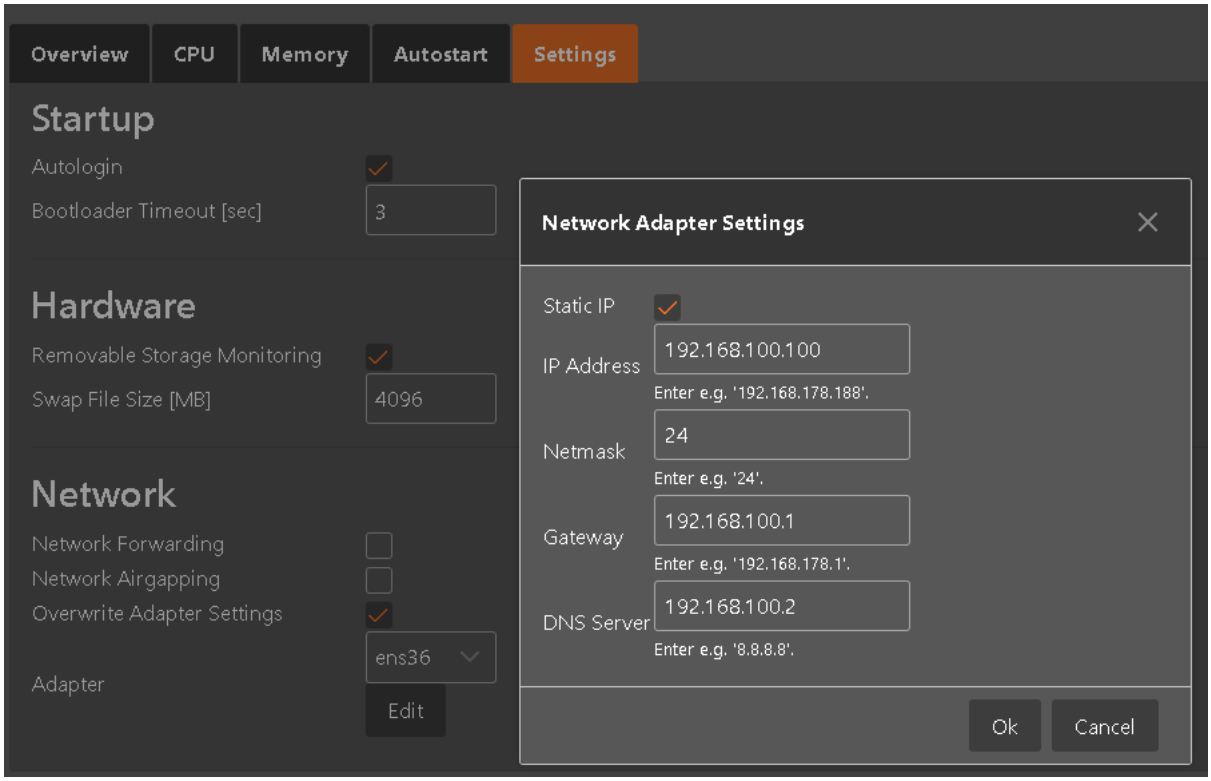
13.1.3 Network settings



If network traffic shall be forwarded between different network adapters, the *Network Forwarding* checkbox has to be enabled. Network forwarding is required if TCP/IP traffic shall be forwarded from externally connected systems to Real-time guests.

If you want to completely disconnect the Hypervisor host itself from the network (e.g. for security reasons) you may enable the *Network Airgapping* checkbox. See section *Network airgapping* in the Hypervisor manual for more details.

The setting for each network adapters that are available can be overwritten. By default, all such adapters are configured to using DHCP. You may select an adapter and then adjust the settings appropriately:



Caution: If you change network settings remotely, make sure you do not accidentally cut the network connection to the browser where the System Manager is running.

13.2 System Manager internal settings

The following settings (stored in `/hv/sysmgr/HvSystemManager.xml`) can be used to change the behavior of the System Manager itself. If you want to edit this file, you need to have root rights:

```
$ cd /hv/sysmgr
$ sudo mousepad HvSystemManager.xml
```

Hint: SystemManager has to be stopped before editing HvSystemManager.xml

Setting	Description
ServerPort	System Manager Listening port
RemainingHostMemory	Memory size in MB which should remain for Hypervisor Host
LogLevel	0 = off 1 = Debug (verbose) 2 = Info 3 = Warning 4 = Error
EnableSessionManager	True: only a single session is allowed. False: multiple parallel sessions are enabled (not recommended!)
EnableDynamicCpuConfig	True: arbitrary CPUs can be used by the RTOS. False: only the last n CPUs can be used by the RTOS.
TimerSleepTimeMs	Basic sleep timer delay
TimerNormalCount	Timer multiple for normal polling
TimerSlowerCount	Timer multiple for slow polling
TimerSlowestCount	Timer multiple for slowest polling
HardwareTcpTimeout	Timeout for TCP Gateway (msec)
HardwareMqttTimeout	Timeout for MQTT Gateway (msec)
PrintSysMgrMessages	True: print script messages for System Manager

14 Hypervisor Backend

The System Manager consists of several components based on the hypervisor backend. This is the interface to the Hypervisor. The backend interacts with the Hypervisor (running on the same Hypervisor Host), e.g. to add guests, adjust configuration files etc. The default installation of the hypervisor backend is in `/hv/sysmgr`.

The most common use of the hypervisor backend is through the system manager in the browser, which communicates with the hypervisor backend via the Web Application.

The hypervisor backend can also be used directly by calling the system manager core application.

14.1 System Manager Core Application

The System Manager Backend has a command line interface which supports helpful functions like loading an existing *project* file and applying the related configuration.

This may be useful for automated testing with different guest and configuration scenarios. Project files are stored in `/hv/sysmgr/WorkingDir`.

It is not possible to make multiple changes in one step. Multiple changes can be achieved step by step. To do this, the project is loaded with /loadConfig, the desired change is made (e.g. /CONFIGUREGUEST), and then saved with /saveConfig.

All available parameters of the HvCoreApp are shown with /help.

```
$ cd /hv/sysmgr
$ sudo ./HvCoreApp /help
```

14.1.1 Loading an existing configuration

```
$ cd /hv/sysmgr
$ sudo ./HvCoreApp /loadconfig='./WorkingDir/MyProject.hwc' /sync
```

This will apply an existing configuration stored in `/hv/sysmgr/WorkingDir/MyProject.hwc`.

> **Caution:** Existing guest folders will be removed!

14.1.2 Saving a configuration

```
$ cd /hv/sysmgr
$ sudo ./HvCoreApp /saveconfig='./WorkingDir/MyProject.hwc'
```

This will save the configuration in `/hv/sysmgr/WorkingDir/MyProject.hwc`. If the project file doesn't exist a new one will be created.

14.1.3 Assign CPU to guest

```
$ cd /hv/sysmgr
$ sudo ./HvCoreApp /CONFIGUREGUEST=GUEST0001:CpuMask:0x0001000
```

This will assign CPU 3 to GUEST0001

14.1.4 Adjust amount of CPUs assigned to guest

```
$ cd /hv/sysmgr
$ sudo ./HvCoreApp /CONFIGUREGUEST=GUEST0001:Cpu:2
```

This will assign 2 CPUs to GUEST0001

14.1.5 Adjust guest RAM size

```
$ cd /hv/sysmgr
$ sudo ./HvCoreApp /CONFIGUREGUEST=GUEST0001:Memory:256
```

This will assign 256MB RAM size to RTOS GUEST0001

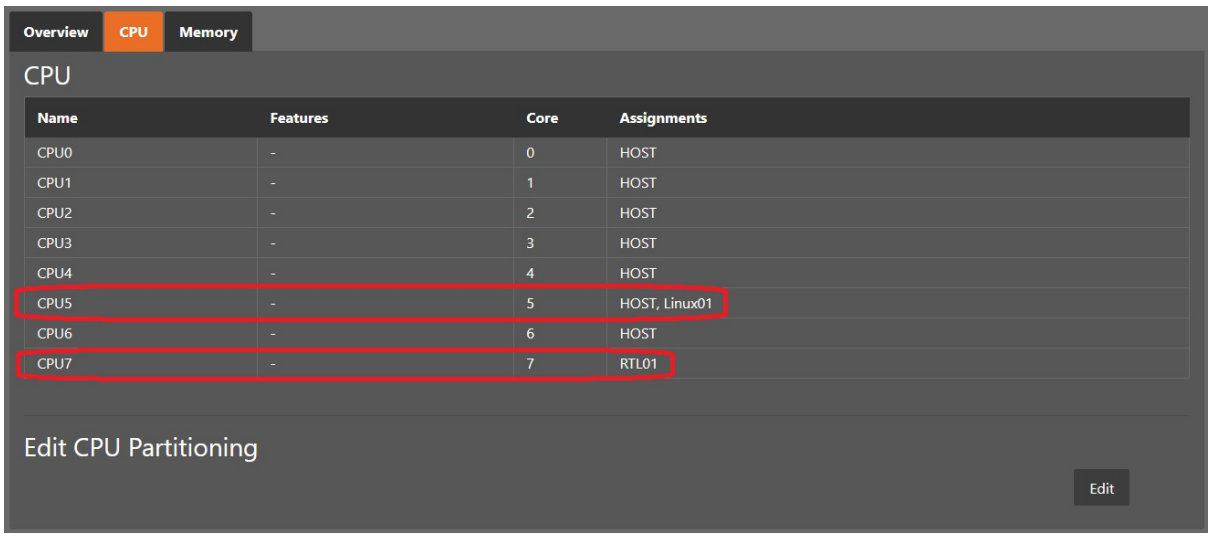
15 Miscellaneous

15.1 Changed Hardware

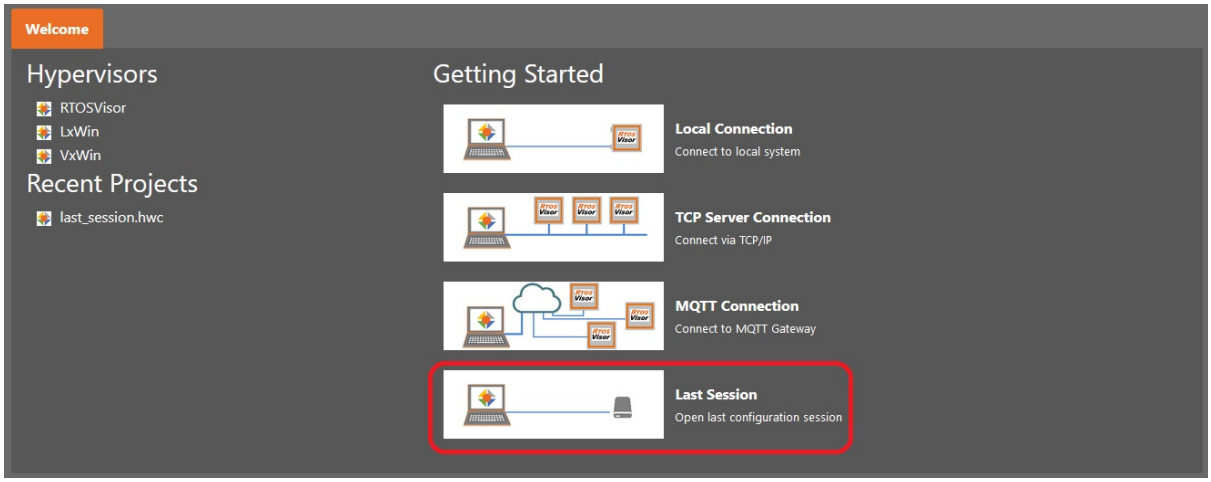
After changing the hardware and restarting the Hypervisor Host you should load your project in the System Manager and synchronize these changes with your project. There are two ways to achieve this:

- You can use the ‘Sync’ button to directly insert the changed hardware information into the project. The System Manager will automatically adjust the project settings based on the information about the hardware changes. Then the project will be automatically saved.
- Another option is to load the project and use the ‘Sync Hardware Info’ menu to read the changes. The System Manager will automatically adjust the project settings based on the information about the hardware changes but the modifications will not be saved to the project. Now you can modify the project settings and then insert them to the current project using the ‘Sync’ button.

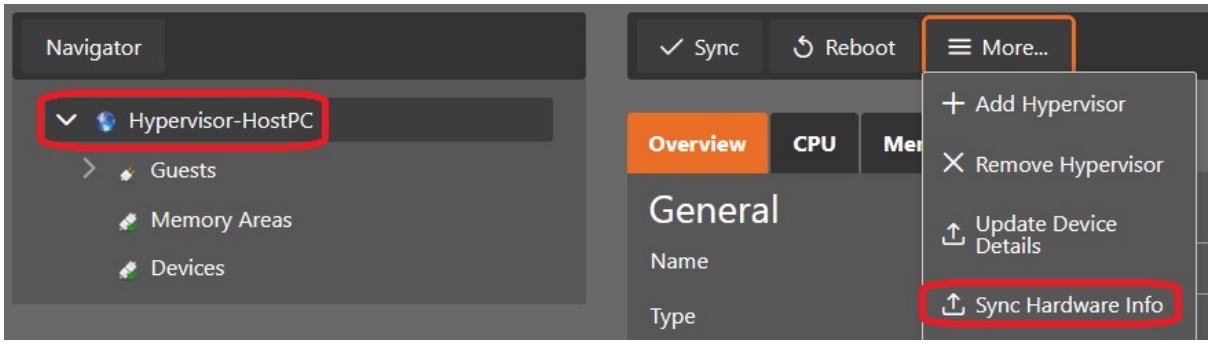
In the following example, we have a Hypervisor Host with activated hyperthreading and 8 logical CPUs. A real-time Linux was assigned to CPU 7 with the designation RTL01 and a general purpose Ubuntu was assigned to CPU 5 with the designation Linux01.



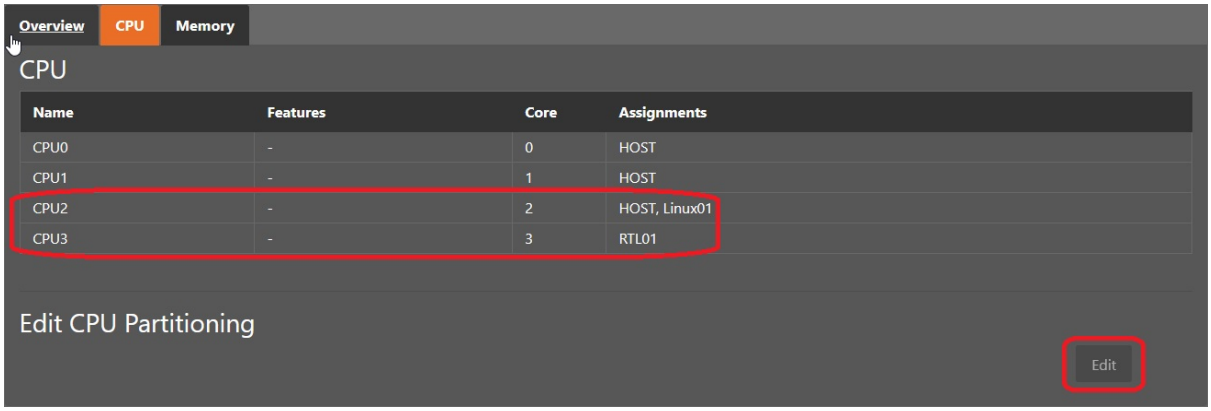
The Hypervisor Host was shut down and hyperthreading disabled in the BIOS settings which results in 4 CPUs. After booting the Hypervisor Host the saved project or the last Session is restored.



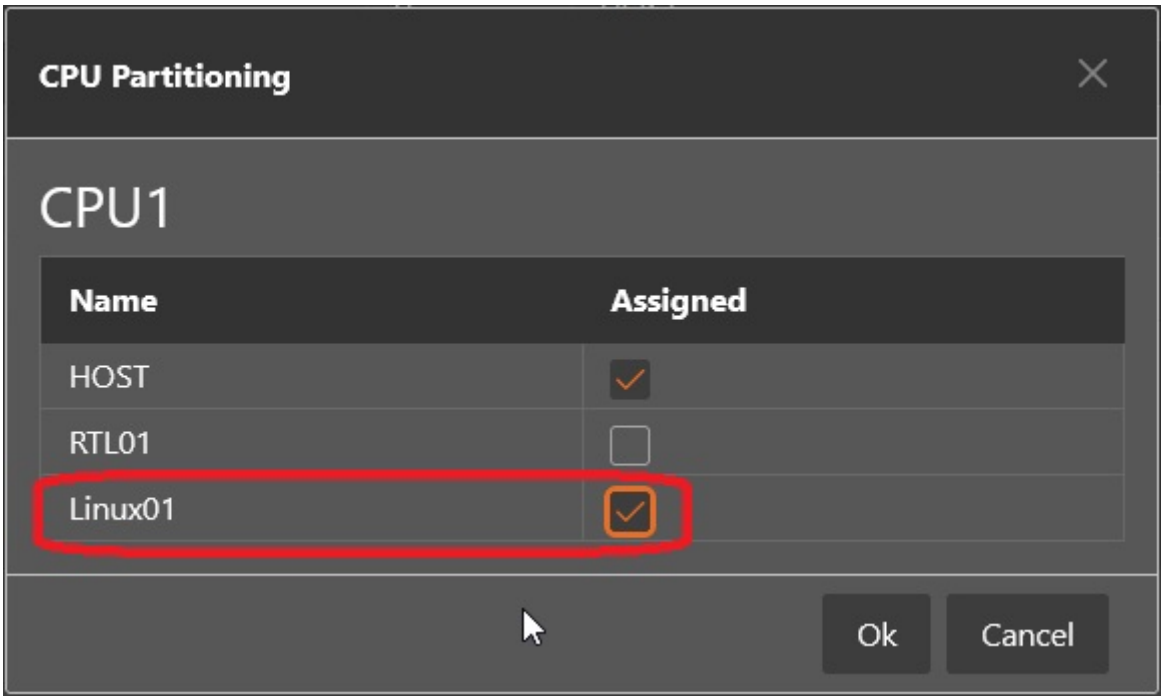
Now changed hardware info can be retrieved with `Sync Hardware Info`.



The new CPU Partitioning is adopted automatically to the changed CPUs.



Now you can edit the CPU Partitioning to fit your needs.



After adjusting the hardware settings in your project press the ‘Sync’ button.