

MERLIC RTE for Linux

How to install and run MERLIC RTE for Linux
on PC or Arm[®]-based systems

Version 5.6.0

Unless otherwise stated, the complete contents of this documentation, including but not limited to all text, designs and images appearing herein, are owned and copyrighted by MVTec Software GmbH. No part of this documentation may be reproduced, distributed, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, recording, or otherwise, without prior written permission of the publisher.

"MVTec Software GmbH" and "MERLIC" are registered trademarks of MVTec Software GmbH.

Metal is a trademark of Apple Inc., registered in the U.S. and other countries and regions.

Microsoft, Windows, and Windows 10 are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries worldwide.

OpenGL is a trademark or registered trademark of Silicon Graphics, Inc. in the United States and/or other countries worldwide.

GigE Vision and USB3 Vision are trademarks of the AIA registered in the United States and/or other countries worldwide. GenICam is a trademark of the European Machine Vision Association (EMVA).

Hilscher is a registered trademark of Hilscher Gesellschaft für Systemautomation mbH.

Arm is a registered trademark of Arm Limited.

NVIDIA and Jetson are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and/or other countries.

Raspberry Pi is a trademark of Raspberry Pi Ltd.

Vulkan® is a registered trademark of the Khronos Group Inc.

All other brand names, designs, service marks and trademarks (whether or not registered) referenced or used herein are the property of their respective owners.

Contents

1 About	3
2 System Requirements	3
3 License Activation	4
4 Required Files	5
5 Installing MERLIC RTE for Linux	5
5.1 Providing the MERLIC5.ini File	6
5.2 Checking the Firewall Settings	7
6 Setting Up the MVApp for Linux	8
6.1 Setting Up the Image Source	8
6.2 Adjusting the Paths in the Recipe	9
7 Starting MERLIC RTE for Linux	9
7.1 Starting MERLIC RTE for Linux without Recipe	10
7.2 Starting MERLIC RTE for Linux with a Specific .ini File	10
7.3 Starting MERLIC RTE with Communicator Plug-ins	10
7.4 Starting a Remote MERLIC Frontend	11
8 Uninstalling MERLIC RTE for Linux	11

1 About

The following documentation provides information on the MERLIC version for Linux. It can be used to run MERLIC Runtime Environment (MERLIC RTE) for Linux on PC and on Arm-based systems, for example, embedded devices.

This version only contains selected components of MERLIC including MERLIC RTE, the MERLIC Communicator, as well as the MVApp examples and Communicator plug-ins.

To run an application on Linux via PC or an Arm-based system, you need an MVApp (including the corresponding recipes) that was created on a Windows system using MERLIC Creator. This could be an MVApp that is already in use on your Windows system, an MVApp example, or a new MVApp that you created especially for this purpose. You can then use the MVApp on Linux via PC or an Arm-based system with MERLIC RTE. Image sources can be set up remotely using the MERLIC RTE Setup.

2 System Requirements

To use MERLIC RTE on Linux systems, the following system requirements must be fulfilled:

PC-based Systems

Component	PC-based systems
Operating system	Linux 5.0
Processor	Intel 64 or AMD 64
Libraries	glibc 2.27 (or higher) libX11 libxcbcommon libXext libGL libEGL libfontconfig libfreetype
Graphics	1920x1080, 32-bit color In addition, one of the following graphics API must be provided on the system: <ul style="list-style-type: none"> • Direct3D 11.1 or Direct3D 12 • Metal 1.2 or higher • OpenGL 3.0 or higher (OpenGL 3.3 or higher is strongly recommended) • OpenGL ES 2.0 or higher (OpenGL ES 3.0 or higher is strongly recommended) • Vulkan 1.0 or higher

Arm-based Platforms

Component	Arm-based platforms
Operating system	Linux 5.0
Processor	Armv8-A-64bit compliant with support for NEON instruction set
Libraries	glibc 2.27 (or higher) libX11 libxkbcommon libXext libGL libEGL libfontconfig libfreetype
Graphics	1920x1080, 32-bit color In addition, one of the following graphics API must be provided on the system: <ul style="list-style-type: none"> • Direct3D 11.1 or Direct3D 12 • Metal 1.2 or higher • OpenGL 3.0 or higher (OpenGL 3.3 or higher is strongly recommended) • OpenGL ES 2.0 or higher (OpenGL ES 3.0 or higher is strongly recommended) • Vulkan 1.0 or higher

This version was tested specifically on the following Arm-based systems:

- Raspberry PI 64-bit
- NVIDIA Jetson Xavier NX Developer Kit

This version might also work on different Arm-based systems. However, none but the above mentioned systems have been specifically tested.

This version requires significantly less disk or flash memory when not all functionality, e.g., Deep Learning, is needed. The installation can be manually reduced as far as 400 MB.

3 License Activation

To use MERLIC RTE on Linux, you need to have an activated license.

- Make sure that you have downloaded and installed CodeMeter Runtime, the licensing software used by MERLIC. If no CodeMeter Runtime is installed yet, go to the [CodeMeter download page](#) and select the required package in the section "**CodeMeter User Runtime for Linux**". We recommend installing the latest version of the standard package for the used architecture and distribution, that is, the package without any further name tag after the version number. Alternatively, the "Driver Only" can be used if no standard package is available.

- Make sure that you have a license dongle with an active MERLIC license that is plugged into your Linux system. For more information about the license activation, see the topics [How to Activate a MERLIC License](#) and [Troubleshooting for License Activation](#) in the MERLIC manual.
- If you do not yet have a license dongle, please get in touch with our [MERLIC distributors](#) or our [sales team](#).

4 Required Files

The following files are necessary after the installation to use MERLIC RTE for Linux:

- **MVApp (.mvapp)**: This can either be one of the provided MVApp examples from the **merlic-5.6.0-examples.tar.gz** archive, or an MVApp that you prepared with MERLIC Creator on a Windows system. Read more in the section [Setting Up the MVApp for Linux](#).
- **Recipe (.mrcp)**: At least one recipe file is required that references the MVApp that should be executed and that contains the desired values of the MVApp parameters. Read more in the section [Adjusting the Paths in the Recipe](#).
- **.ini file (.ini)**: The archives come without .ini file. You can either copy the .ini file from the MERLIC version that you used to create the MVApp or write your own .ini file. However, there are some important things to consider when handling the .ini file. Read more in the section [Providing the MERLIC5.ini File](#).

5 Installing MERLIC RTE for Linux

1. You can download the archives necessary for the installation of MERLIC RTE for Linux from the [MVTec download area](#). Use the filter for the operating system to show only the download links for Linux systems. You can also filter for the desired architecture.
2. Download the desired archives for MERLIC RTE and the MERLIC tools:

Runtime Version - MERLIC 5.6.0 Runtime for aarch64-linux	
Content	MERLIC RTE for Linux on Arm-based systems
Archive	merlic-5.6.0-aarch64-linux_merlic_rte-dl.tar.gz
Runtime Version - MERLIC 5.6.0 Runtime for x64-linux	
Content	MERLIC RTE for Linux on PC
Archive	merlic-5.6.0-x64-linux_merlic_rte-dl.tar.gz
Add-on to Full / Runtime Version - MERLIC 5.6.0 Tools for the Runtime version	
Content	This package contains all MERLIC tools. It is necessary to unpack this archive for MERLIC RTE to run on Linux via PC or an Arm-based system.
Archive	merlic-5.6.0-tools_all.tar.gz

- Download the archives for the examples:

Add-on to Full / Runtime Version - MERLIC 5.6.0 Examples for the Runtime version	
Content	This package contains MVApp examples. This archive is optional.
Archive	merlic-5.6.0-examples.tar.gz

- Unpack the downloaded archives in a directory of your choice. A new directory "merlic-5.6.0" will be created.

Linux on PC

```
tar xf merlic-5.6.0-x64-linux_merlic_rte-dl.tar.gz
```

Linux on Arm-based system

```
tar xf merlic-5.6.0-aarch64-linux_merlic_rte-dl.tar.gz
```

- Unpack the archive for the MERLIC tools into the new directory "merlic-5.6.0".

```
tar xf merlic-5.6.0-tools_all.tar.gz
```

- Optionally, unpack the archive with the example MVApps into the new directory "merlic-5.6.0":

```
tar xf merlic-5.6.0-examples.tar.gz
```

MERLIC is now installed on your Linux system. No further installation steps are necessary.

Example

If your directory of choice is "/opt/projects/MVTec", all of the unpacked archives should now be located in "/opt/projects/MVTec/merlic-5.6.0".

5.1 Providing the MERLIC5.ini File

By default, MERLIC RTE uses the configuration of a specific .ini file named "MERLIC5.ini". This file is expected in the directory "~/ .config/MVTec/".

However, the archives for the installation of MERLIC RTE on Linux or Arm-based systems come without any .ini file. Therefore, you have to add the file manually after the installation. You can do this by either creating a new .ini file or by copying an existing .ini file from a different installation and adjusting the configuration in the file.

For example, if you have a MERLIC installation on your Windows system, we recommend to use the .ini file that was used when creating your MVApps as basis for the .ini file on your Linux system.

Keep in mind that you have to make sure to adjust the paths in the .ini file to the new paths on your Linux system as described in the following section.

5.1.1 Required Adjustments

- Adjust the path of the recipe files, e.g., "Recipe0". It must represent the path to the recipe file on your Linux system.
- Adjust any other paths that might be defined in the .ini file, e.g., the path for the log files in "LogFilePath".

5.1.2 Location of the .ini File

If no .ini file is specified when starting MERLIC RTE, a default .ini file is used, called MERLIC5.ini. This .ini file must be available in the directory "~/.config/MVTec". Thus, you have to copy your .ini file to this directory with the name "MERLIC5.ini" if you want to use your .ini file by default.

You can also store your .ini file in any other directory. However, in this case, you have to manually specify the .ini file to be used with the command line option "--ini" when starting MERLIC RTE for Linux. The .ini file must be given with the relative path to the current directory in which MERLIC RTE is started.

Example

If your recipe files are located in the directory "/opt/projects/MVTec/merlic/mvapp", an example MERLIC5.ini could look like this:

```
[Logging]
LogFilePath=/opt/projects/MVTec/merlic/mvapp/logs/
LogFileSizeInBytes=5242880
LogFilesCount=10
Threshold=info
UseShortFilenames=true

[ProcessIntegration]
Recipe0=/opt/projects/MVTec/merlic/mvapp/merlic_rte.mrcp
DefaultRecipe=0

[Frontend]
GuiLanguage=en_US
```

5.2 Checking the Firewall Settings

If you are using a firewall, you have to make sure that your firewall settings allow the remote connection between your Linux system and the respective MERLIC components on the Windows system that is used for the remote configuration.

For the remote connection, the following ports must be allowed.

MERLIC Directory Agent

The Directory Agent is the MERLIC process that coordinates the availability of the configuration services on a host device. It provides the MERLIC RTE Setup with the ability to connect to it on port **9650** and query a list of available instances of MERLIC Creator, MERLIC RTE, and MERLIC Communicator.

Ports for the "Image Sources" and "Communication" Tabs of the MERLIC RTE Setup

The configuration services of the Image Source Manager (ISM) and the Communicator plug-ins are set to ephemeral ports by default. To change the firewall settings to allow the remote configuration of image sources and Communicator plug-ins, you have to define fixed ports instead and allow the communication via these ports.

You have to define the ports in the MERLIC5.ini for the property "ImageSourceConfigurationPorts" in section [General] and for the property "PluginConfigurationPorts" in section [Communicator]. Both properties require exactly three ports to ensure that the remote configuration works correctly as shown in the following example.

```
[General]
ImageSourceConfigurationPorts=10123;10124;10125

[Communicator]
PluginConfigurationPorts=10126;10127;10128
```

For more information about these port settings, see the topic [The MERLIC .ini File](#) in the MERLIC manual.

If these properties are used to defined fixed ports, only one instance of MERLIC RTE can be started. If you want to use a second instance of MERLIC RTE, you have to start the second instance with an additional .ini file in which different ports are set for these properties.

Ports for the MERLIC Frontend

To start and run the MERLIC Frontend on a remote system, you also have to allow the communication via the port of the MERLIC Frontend. By default, the ports **9090-9099** are used for the MERLIC Frontend.

6 Setting Up the MVApp for Linux

When creating an MVApp and the recipe files in the MERLIC Creator, all paths and settings are configured for the respective Windows system on which the application is created. Therefore, some adjustments are required before they can be used on a Linux system.

6.1 Setting Up the Image Source

To ensure that the desired images will be acquired when running MERLIC RTE on the Linux system with your MVApp, you have to adjust the configuration of the image source that is used in the MVApp. For this, you can use the remote connection of the MERLIC RTE Setup. However, at the start, you have to make sure to copy the MVApp to the Linux system.

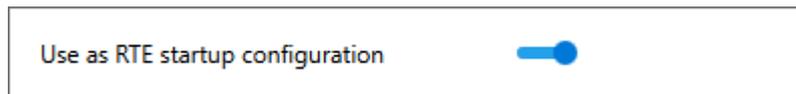
At first, you have to ensure that the configuration of image sources can be enabled for the MERLIC RTE Setup:

1. Start MERLIC RTE on the Linux system without a prepared recipe as described in the section [Starting MERLIC RTE for Linux](#).

```
./merlic_rte --no_recipe
```

In the next steps, you can open the MERLIC RTE Setup and start to configure the image source:

2. Change to your Windows system and start the MERLIC RTE Setup from the start menu or from the command line.
3. In the "Select system" dialog, connect the MERLIC RTE Setup to your Linux system.
 1. Select "Remote system" and enter the hostname or the IP address of your Linux system.
 2. Click on "Connect". The MERLIC RTE Setup opens with a remote connection to the specified system.
4. Go to the "Image Sources" tab. If any configuration is available, i.e., if the respective configuration directory is located in the same directory as the used .ini file, it will be shown in the overview of the image source configurations.
5. Adjust the image sources of the existing configuration or create a new configuration with an image source if none are available. For more information on the creation of configurations and image sources, see the topic [MERLIC Image Source Manager \(ISM\)](#) in the MERLIC manual.
6. Select an image source configuration as default for MERLIC RTE. This "RTE startup configuration" will be automatically activated when MERLIC RTE starts.
 1. Select the desired configuration that contains the image sources from which you want to acquire the images by default.
 2. Set the selected configuration as "RTE startup configuration" by clicking on the respective handle.



7. When you finished setting up the image source, you can close the MERLIC RTE Setup and change back to the Linux system.

6.2 Adjusting the Paths in the Recipe

Check the path to the MVApp in your recipe files. The path must correspond to the relative path in which the referenced MVApp is located. If your relative path from the recipe file to the MVApp is still the name on the Linux system, no changes will be required.

7 Starting MERLIC RTE for Linux

1. Change to the directory "bin/<ARCH>" in your installation directory.

Linux on PC

```
cd <MERLIC_INSTALLATION>/bin/x64-linux
```

Linux on Arm-based system

```
cd <MERLIC_INSTALLATION>/bin/aarch64-linux
```

2. Enter the following command to start MERLIC RTE:

```
./merlic_rte
```

If no .ini file is specified, MERLIC RTE uses the configuration defined in the "MERLIC5.ini" file which is expected in the directory "~/.config/MVTec/". If this file is not available, see the section [Providing the MERLIC5.ini File](#) for more information on how to provide the .ini file. If you want to use a different .ini file, you can use the command line option "--ini".

MERLIC also provides several command line options for "merlic_rte". For a list of these options, see the topic [Starting the Process Integration Mode](#) of the MERLIC manual.

7.1 Starting MERLIC RTE for Linux without Recipe

In some cases, it might be required to run MERLIC RTE without a prepared recipe., e.g., if you want to set up a new image source.

1. Change to the directory "bin/<ARCH>" in your installation directory as described in the previous section.
2. Start MERLIC RTE for Linux with the command line option "--no_recipe" or "-R":

```
./merlic_rte --no_recipe
```

7.2 Starting MERLIC RTE for Linux with a Specific .ini File

1. Change to the directory "bin/<ARCH>" in your installation directory as described in the previous section.
2. Start MERLIC RTE for Linux with the command line option "--ini":

```
./merlic_rte --ini <PATH_TO_INI_FILE>/<FILE>.ini
```

7.3 Starting MERLIC RTE with Communicator Plug-ins

1. Change to the directory "bin/<ARCH>" in your installation directory.
2. Start MERLIC RTE and the MERLIC Communicator on your Linux system at the same time:

```
./merlic_rte --communicator
```

3. Change to your Windows system and start the MERLIC RTE Setup from the start menu or from the command line.
4. In the "Select system" dialog, connect the MERLIC RTE Setup to your Linux system.
 1. Select "Remote system" and enter the hostname or the IP address of your Linux system.
 2. Click on "Connect". The MERLIC RTE Setup opens with a remote connection to the specified system.
5. Open the "Communication" tab of the MERLIC RTE Setup, add the plug-in of your choice and configure it if necessary.

6. Start the plug-in via the "Start plug-in" button. Because the MERLIC RTE is remotely connected to your Linux system, the plug-in will now be running on Linux.

You can find more information on Communicator plug-ins and their configuration in the [MERLIC Communicator manual](#).

7.4 Starting a Remote MERLIC Frontend

1. Change to your Windows system and open a command prompt.
2. Change to the directory "bin\x64-win64" within the MERLIC installation path, e.g., "%PROGRAMFILES%\MVTec\MERLIC-5.6\bin\x64-win64".
3. Start a Frontend with a remote connection to the Linux system. For this, use "merlic_frontend.exe" and the command line option "--tcp" to connect the Frontend to the MERLIC server at <host:port>. "host" defines the name or IP of the host machine and refers to the PC or Arm-based system you are using to start MERLIC RTE for Linux. "port" defines the port to which the server listens. The default port is 9090.

```
merlic_frontend.exe --tcp <HOST:PORT>
```

8 Uninstalling MERLIC RTE for Linux

MERLIC provides no uninstallation script for Linux systems, therefore you must perform the uninstallation manually:

```
rm -rf <MERLIC_INSTALLATION>
```

Uninstalling CodeMeter Runtime

Use the system-specific option of your Linux distribution to uninstall CodeMeter Runtime.