Installation Guide

This document describes how to install MeVisLab on the supported platforms.

System Requirements

Table 1. MeVisLab System Requirements

	Recommended Hardware	Minimum Requirements
Windows	• Windows 10 or 11 • Intel Core-i 6000 or above / AMD Ryzen processor • 16 GB RAM • AMD Radeon or NVIDIA higher-end graphics card 4 GB VRAM	• Windows 7 • Intel Core-i 2000 or above / AMD Ryzen processor • 4 GB RAM • an OpenGL compliant graphics board
Linux	• Ubuntu (latest) • Intel Core-i 6000 or above / AMD Ryzen processor • 16 GB RAM • AMD Radeon or NVIDIA higher-end graphics card 1 GB VRAM	• Ubuntu 22.04 • Intel Core-i 2000 or above / AMD Ryzen processor • 4 GB RAM • an OpenGL compliant graphics board

Note

The MeVisLab volume rendering and other advanced visualization features are not supported on onboard Intel graphics (e.g., Intel HD family boards). For the PathTracer framework CUDA support is needed, which is only offered by NVIDIA graphics cards. A screen resolution of 1920 × 1080 or higher is recommended.

The following chapter gives information on some operating system (OS) specific settings and things to keep in mind, especially when working on module development. Read the respective OS sections in case of compilation and compiler problems and when in need of troubleshooting.

- the section called "Windows Guide"

- the section called "Linux Guide"

Windows Guide

Installation on Windows

For Windows, a typical installation of the SDK results in the following programs being installed:

- **MeVisLab** in release and debug version. This versions will scan the installed packages at start-up to update the internal module database.

- **QuickStart MeVisLab** in release and debug version. This will start MeVisLab without scanning the installed packages; while being much faster, new or removed modules will not cause a change of the

internal database. However, if no modules are being developed, this programs are what you want to start.

- **MeVisLab MATE**, this is an integrated text editor.

- **MeVisLab ToolRunner**, this is a meta-tool.

Figure 1. MeVisLab Icons

> **Note**
>
> Debug versions can only be started if a development environment (Visual Studio) is installed on the computer.

Module Development on Windows

For C++ module development, you can usually use a later Visual Studio version with a MeVisLab that was created for an earlier Visual Studio version.

There may also be differences between Visual Studio Express versions and the full Visual Studio versions (standard or professional). Make sure that the compiler version that you intend to use is supported by your Visual Studio installation.

For compilation, CMake is used. See the CMake Manual for further information.

Linux Guide

MeVisLab on Linux

This section shows how to handle MeVisLab on Linux:

- the section called "Installing MeVisLab"

- the section called "Running MeVisLab"

- the section called "Module Development on Linux"

Installing MeVisLab

Get MeVisLab from https://www.mevislab.de/download/. Only a 64-bit version is available, so you must have a 64bit version of Ubuntu installed. After the download you must make the installer executable.

Run the command 'chmod u+x' and execute '*.bin' file. Select an installation directory. We will refer to it as `InstallDirectory` from now.

Running MeVisLab

You can start MeVisLab from the applications menu entry. Alternatively, you can run `InstallDirectory/bin/MeVisLab`. `InstallDirectory/bin/MeVisLab_d` will start the debug version of MeVisLab.

Module Development on Linux

MeVisLab offers "Module Wizards", which are accessible via the file menu. They create the skeleton of ML and Inventor modules. The sources are created in a folder named by your module in the "Source" subdirectory of a MeVisLab package.

Note

Environment variable `MLAB_ROOT` is required to build C++ modules. `MLAB_ROOT` must point to `InstallDirectory/Packages`.

For compilation, CMake is used. See the CMake Manual for further information.

Docker Guide

Installation on Docker

The MeVisLab SDK for Linux can also be installed and used within a Docker image.

Since MeVisLab for Linux was built for Ubuntu, it is best to use Ubuntu as the base image.

In addition MeVisLab always needs a running X server. Therefore we use xvfb for the installation. To be able to use MeVisLab after installation, you must also provide an X server. There are several ways to do this, including using xvfb.

We also use a multistage build to get the final image a bit smaller.

```
# ---------------------
# common base container
# ---------------------
FROM ubuntu:24.04 AS base

ARG DEBIAN_FRONTEND=noninteractive

ENV XDG_RUNTIME_DIR='/tmp/runtime-root'

RUN apt-get update \
 && apt-get upgrade -y \
 && apt-get install -y --no-install-recommends \
    apt-utils \
    fontconfig \
    libfontconfig1-dev \
    libgl1-mesa-dev \
    libglu1 \
    libglu1-mesa-dev \
    libgomp1 \
    libgstreamer-plugins-base1.0 \
    libgstreamer1.0-0 \
    libnss3 \
    libopengl0 \
    libxcb-cursor0 \
```

```
    libxcb-icccm4 \
    libxcb-image0 \
    libxcb-keysyms1 \
    libxcb-randr0 \
    libxcb-render-util0 \
    libxcb-xinerama0 \
    libxcb-xinput0 \
    libxcb-xkb1 \
    libxkbcommon-x11-0 \
    locales \
    mesa-common-dev \
    ttf-bitstream-vera \
    wget \
    xorg \
    xvfb \
    file \
 && sed -i -e 's/# en_US/en_US/' /etc/locale.gen \
 && locale-gen \
 && apt-get autoremove \
 && apt-get clean

ENV LANG='en_US.utf8'
ENV LC_ALL='en_US.utf8'
ENV LANGUAGE='en_US.UTF-8:en'

# ---------------------------------------------
# intermediate container for the installation
# ---------------------------------------------
FROM base AS installer

RUN wget -O /MeVisLabInstaller.bin https://mevislabdownloads.mevis.de/Download/MeVisLab4.1.2/Linu
RUN chmod +x /MeVisLabInstaller.bin
RUN xvfb-run /MeVisLabInstaller.bin --mode silent --prefix /opt/mevislab --no-desktop-files

# ---------------------------------------------
# final image
# ---------------------------------------------

FROM base AS sdk

ENV MLAB_ROOT=/opt/mevislab/Packages
COPY --from=installer /opt/mevislab /opt/mevislab
```