Qt DOCUMENTATION

≡

🔍 Search

Topics ❯

Qt Creator Manual > Adding Docker Devices

Adding Docker Devices

You can add Docker images as *devices* to run, debug, and analyze applications built for the Docker container from Qt Creator. A Docker device operates like a virtual machine but uses less system resources at the cost of being less flexible.

Docker support is experimental. While Linux, macOS, and Windows hosts are supported in principle, Linux is the recommended platform.

Currently, only CMake is supported for building applications in the Docker container.

You can run applications locally or in a compatible Docker container. To be able to run and debug applications on Docker devices, you must install Docker as well as add Docker devices and select them in the kit. Qt Creator automatically detects kits in the shared Docker directories, but you need to check that they point to the correct kit items.
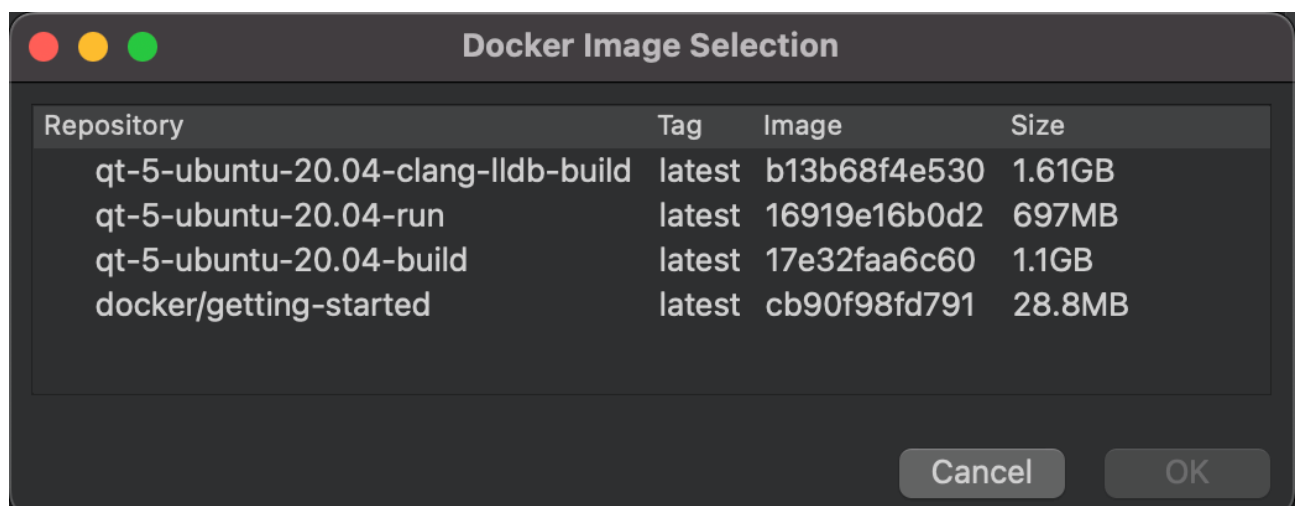
Use a wizard to search for Docker images available on your local Docker installation and add them as devices. To access images from Docker hub or other registries, you first need to pull the images using the docker pull command. You can edit the Docker device preferences later in **Edit** > **Preferences** > **Devices**.

To enable the experimental Docker plugin:

1. In Qt Creator, select **Help** > **About Plugins** > **Utilities** > **Docker (experimental)**.
2. Select **Restart Now** to restart Qt Creator and load the plugin.

To add a Docker image as a device:

1. Select **Edit** > **Preferences** > **Devices** > **Devices** > **Add** > **Docker Device** > **Start Wizard** to search for images in your local Docker installation.

Repository	Tag	Image	Size
qt-5-ubuntu-20.04-clang-lldb-build	latest	b13b68f4e530	1.61GB
qt-5-ubuntu-20.04-run	latest	16919e16b0d2	697MB
qt-5-ubuntu-20.04-build	latest	17e32faa6c60	1.1GB
docker/getting-started	latest	cb90f98fd791	28.8MB

Cancel   OK

Qt DOCUMENTATION

Note: If the Docker process is not found, make sure that Docker is running and the Docker CLI executable is set in PATH.

3. In **Devices**, check and modify Docker device preferences.
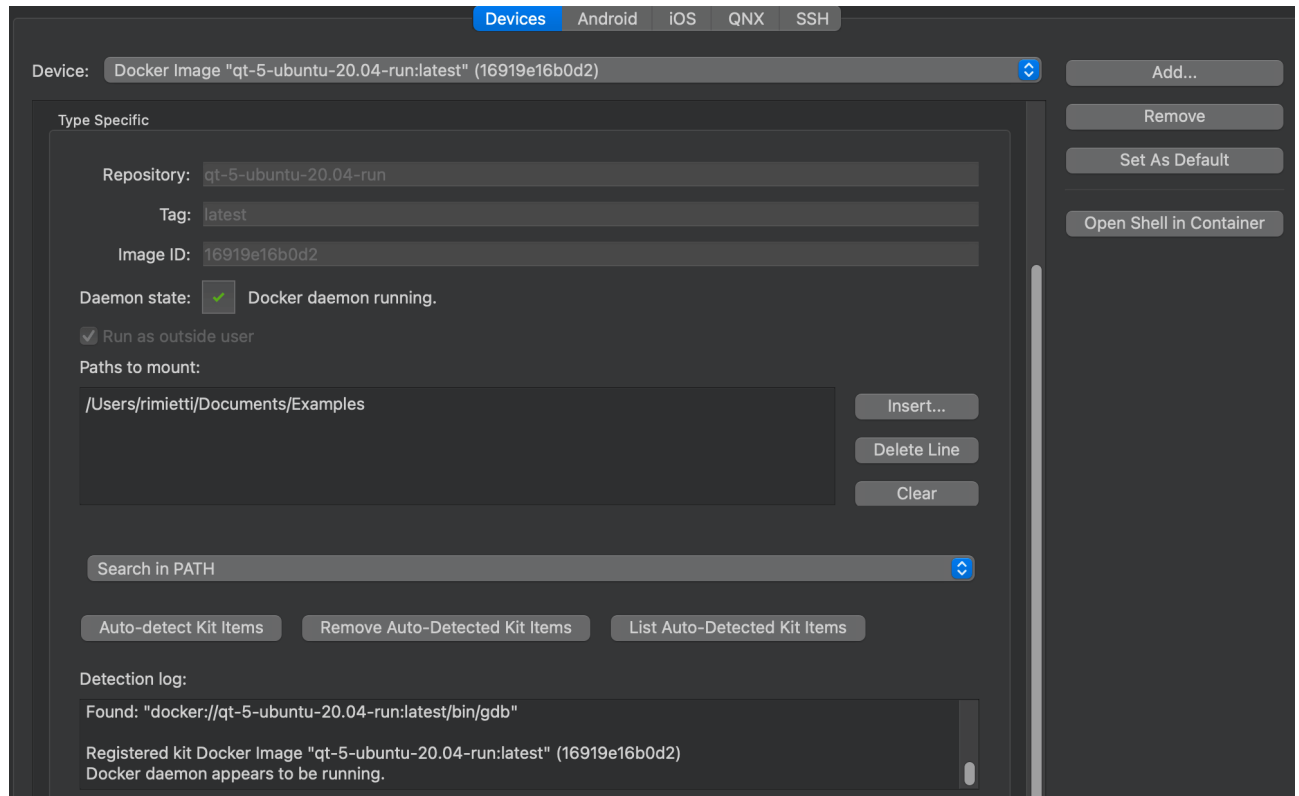
4. Select **Run as outside user** to use the user ID and group ID of the user running Qt Creator in the Docker container.

5. In **Paths to mount**, specify host directories to mount into the container, such as the project directory.

6. Select **Auto-detect Kit Items** to generate an initial build and run kit for the Docker device.

Specifying Paths to Mount

You can either copy your project files into the Docker container or specify paths to them in **Paths to mount**. Shared mounts are restricted to locations in the host system that can end up in the same absolute location in the Docker container. On Windows, mounted drives cannot be used as shared mounts.

The paths in **Paths to mount** are mapped one-to-one to the Docker container. Select **Insert** to browse directories to add. Select **Delete Line** to delete the selected path or **Clear** to delete all paths.

Auto-detecting Kit Items

Select **Auto-detect Kit Items** to generate an initial build and run kit for the Docker device. You can either set the kit items, such as debuggers and Qt version, in PATH or install them in the Docker container.

Select **Search in PATH** to detect kit items that are set in PATH.

Select **Search in Selected Directories** to detect kit items in the selected directories.

To view the automatically detected kit items, select **List Auto-Detected Kit Items**. To remove them, select **Remove Auto-Detected Kit Items**.

Qt DOCUMENTATION

Editing Docker Device Kits

Select **Edit** > **Preferences** > **Kits** to check that the automatically generated kit points to the appropriate kit items.

To specify build settings:

1. Open a project for an application you want to develop for the device.
2. Select **Projects** > **Build & Run** to enable the kit that you specified above.

Select **Run** to specify run settings. Usually, you can use the default settings.

‹ Connecting Boot2Qt Devices   Connecting Generic Remote Linux Devices ›

Qt The Qt Company

Contact Us

Company

About Us

Investors

Newsroom

Careers

Office Locations

Licensing

Terms & Conditions

Open Source

FAQ

Support

Support Services

Professional Services

Partners

Training

For Customers

Support Center

Downloads

Qt Login

Contact Us

Customer Success

Community

Qt DOCUMENTATION

Wiki

Downloads

Marketplace

Feedback   Sign In