添加泊坞设备

您可以将 Docker 映像添加为*设备*，以运行、调试和分析为 Qt Creator 的 Docker 容器构建的应用程序。Docker 设备像虚拟机一样运行，但使用较少的系统资源，但代价是灵活性较低。

泊坞程序支持是实验性的。虽然原则上支持 Linux、macOS 和 Windows 主机，但建议使用 Linux。

目前，只有 CMake 支持在 Docker 容器中构建应用程序。

您可以在本地运行应用程序，也可以在兼容的 Docker 容器中运行应用程序。为了能够在 Docker 设备上运行和调试应用程序，您必须安装 Docker 以及添加 Docker 设备并在工具包中选择它们。Qt Creator 会自动检测共享 Docker 目录中的工具包，但您需要检查它们是否指向正确的工具包项目。

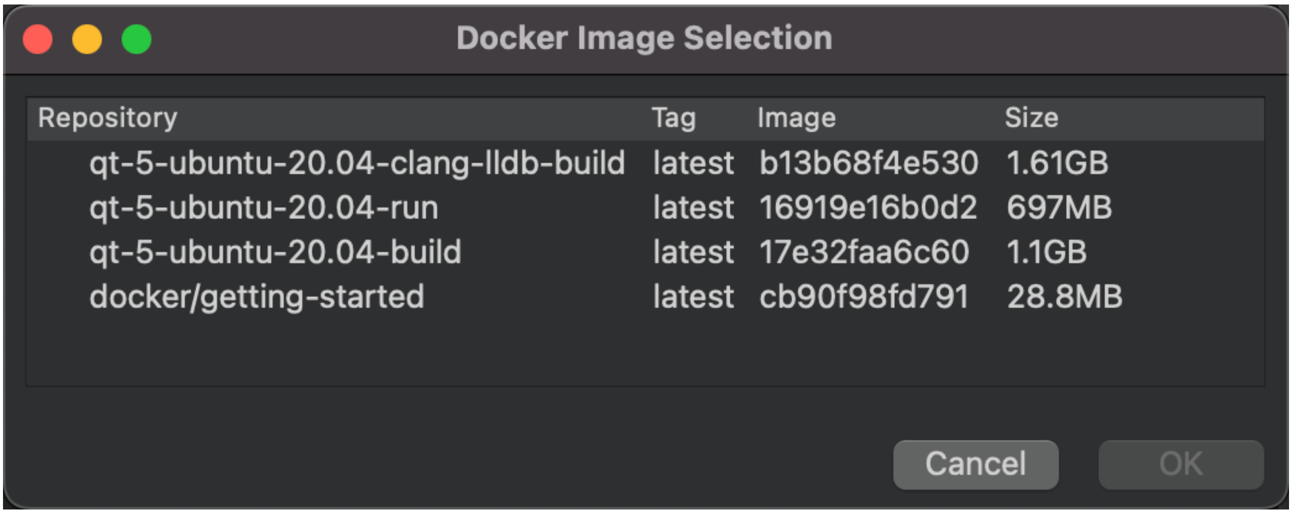
使用向导搜索本地 Docker 安装中可用的 Docker 映像，并将它们添加为设备。若要从 Docker 中心或其他注册表访问映像，首先需要使用 Docker 拉取命令拉取映像。您可以稍后在"编辑>首选项>**设备"中编辑** Docker **设备首选项**。

要启用实验性的 Docker 插件，请执行以下操作：

1. 在 Qt Creator 中，选择"**帮助**">"**关于插件**>**实用程序**> **Docker（实验性）**"。
2. 选择**立即重新启动**以重新启动Qt创建器并加载插件。

要将 Docker 映像添加为设备："

1. 选择">**设备**>**设备编辑**>**首选项**">**添加**> **Docker 设备**>**启动向导**"以在本地 Docker 安装中搜索图像。

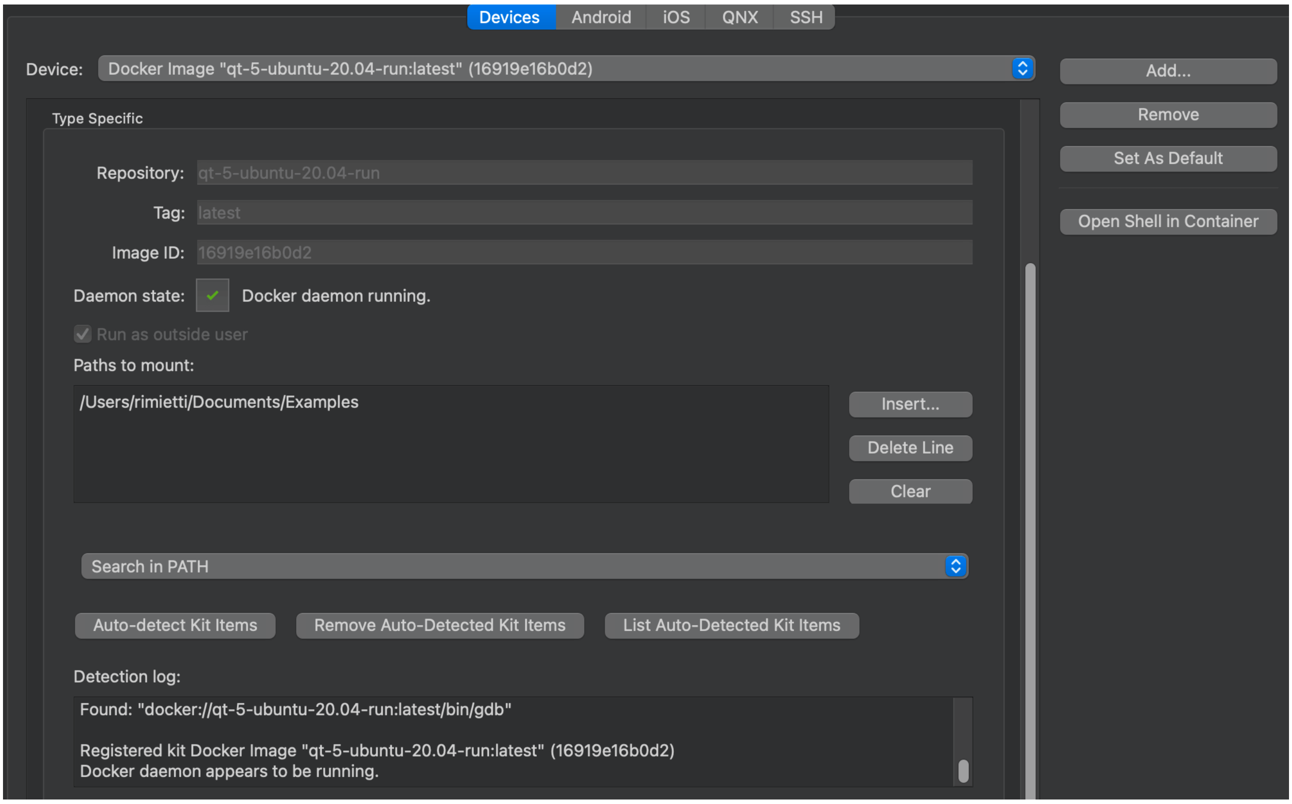


Repository	Tag	Image	Size
qt-5-ubuntu-20.04-clang-lldb-build	latest	b13b68f4e530	1.61GB
qt-5-ubuntu-20.04-run	latest	16919e16b0d2	697MB
qt-5-ubuntu-20.04-build	latest	17e32faa6c60	1.1GB
docker/getting-started	latest	cb90f98fd791	28.8MB

2. 选择要使用的 Docker 映像，然后选择"**确定**"。

> **注意：** 如果未找到泊坞程序进程，请确保泊坞程序正在运行，并且在 PATH 中设置了泊坞窗 CLI 可执行文件

3. 在 **设备** 中，检查并修改 Docker 设备首选项。



Device: Docker Image "qt-5-ubuntu-20.04-run:latest" (16919e16b0d2)

Repository: qt-5-ubuntu-20.04-run

Tag: latest

Image ID: 16919e16b0d2

Daemon state: ✓ Docker daemon running.

☑ Run as outside user

Paths to mount:

/Users/rimietti/Documents/Examples

Search in PATH

Detection log:

```
Found: "docker://qt-5-ubuntu-20.04-run:latest/bin/gdb"

Registered kit Docker Image "qt-5-ubuntu-20.04-run:latest" (16919e16b0d2)
Docker daemon appears to be running.
```

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 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**.

Editing Docker Device Kits

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 DOCUMENTATION

Feedback Sign In