

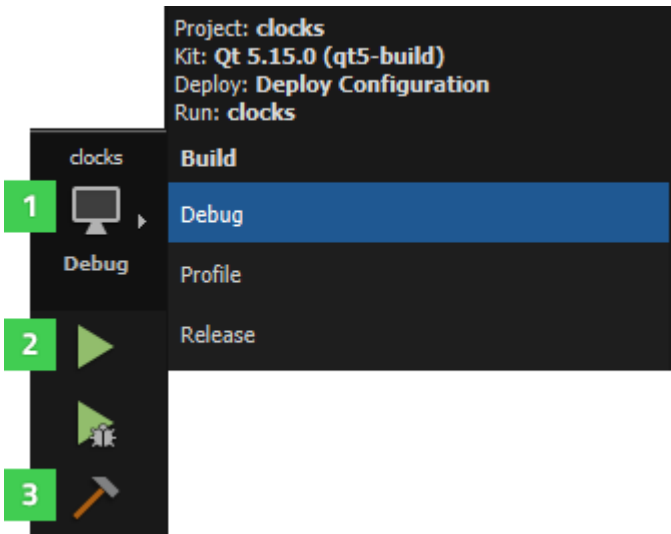
Running on Multiple Platforms

By default, running an application also builds it and deploys it to a location from where it can be run on the desktop, on a device emulator or simulator, or on a **device** that is connected to the development PC.

To run executable files without deploying them first, select **Build > Run Without Deployment**. To make this the default option, deselect the **Edit > Preferences > Build & Run > General > Always deploy project before running it** check box.

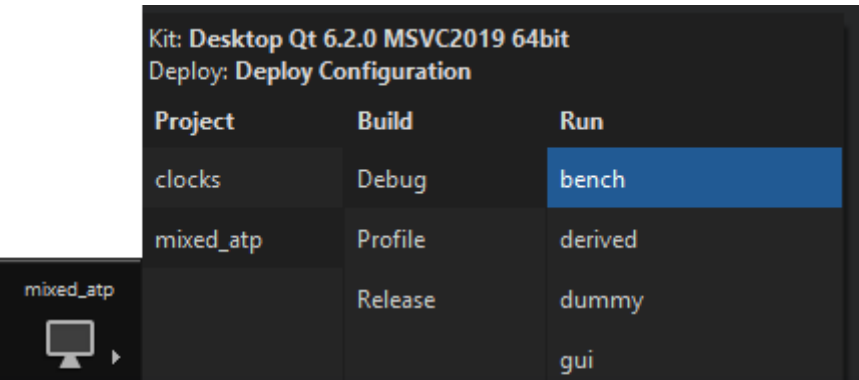
To run applications:

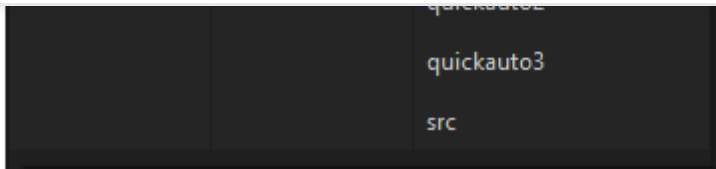
1. Click the **Build and Run Kit Selector** icon (1) or select **Build > Open Build and Run Kit Selector** to select the build and run **kit**.



2. Click the **Run** button (2).

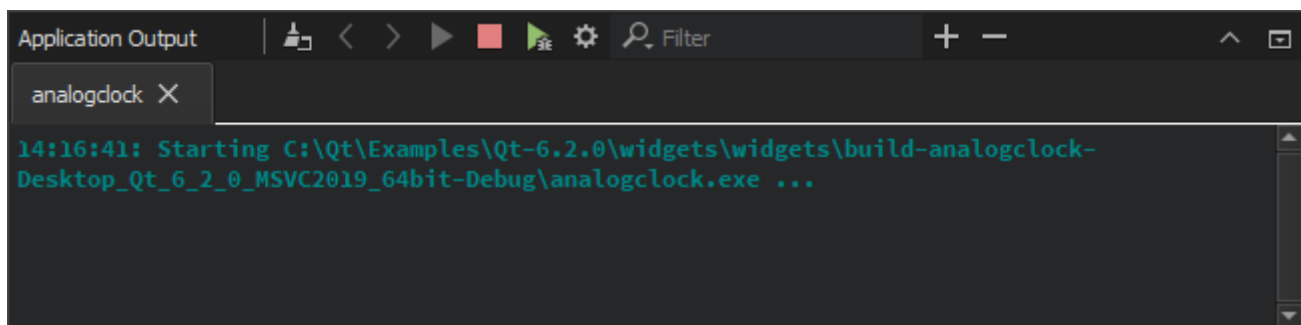
If your project has several run targets defined, such as **tests**, you can select them in the kit selector.





If you have connected [mobile devices](#) or [embedded devices](#) to the development PC or added virtual devices, such as [Android Virtual Devices \(AVD\)](#), you can select them in the kit selector. Select **Manage** to manage device settings. For example, you can add AVDs or manually start disconnected AVDs.

Application Output displays the status of the application while it is running. You can select the **Run** button to re-run applications without building them first. This is useful when developing Qt Quick applications because the QML files are interpreted at runtime. Therefore, the application does not need to be built again if you edited only QML files. This saves time especially if the application contains large image files that would need to be bundled into the resource file before running the application.



For more information on the options you have, see [Specifying Run Settings](#).

Running on Generic Remote Linux Devices

To build the application and run it on a device:

1. Specify a connection to the device. For more information, see [Connecting Generic Remote Linux Devices](#).
2. Click the **Run** button.

Qt Creator uses the compiler specified in the project build settings (tool chain) to build the application.

Qt Creator copies the application files to the connected device and runs the application. The application views are displayed on the device. Command-line output is visible in the Qt Creator **Application Output** view.

Choose **Projects > Embedded Linux > Run** to view the settings for deploying the application to the connected device. For more information, see [Specifying Run Settings for Linux-Based Devices](#). Debugging works transparently if GDB server is installed on the device and it is compatible with the GDB on the host.

Running on QNX Devices

1. Connect the device to the development PC or to the Wi-Fi network.
2. Configure the device and specify a connection to it. For more information, see [Connecting QNX Devices](#).
3. Make sure that your kit has your QNX device set.
4. Click the **Run** button.

Qt Creator uses the compiler specified in the QNX tool chain to build the application.

Troubleshooting Errors

To support running, debugging, and stopping applications from Qt Creator, the QNX Neutrino RTOS should provide a few additional command line tools and services, as described in [Qt for QNX](#).

Debug Output Cannot Be Shown

For the command-line output to show up in the **Application Output**, Qt Creator needs to be able to establish an SSH connection to the device. This is only possible if QNX Momentics is not running, and the SSH key configured for the device is a 4096-bit key.

If these conditions are not met, you will get an error message saying debug output cannot be shown.

Cannot Run, Debug, or Stop Applications

The board support package (BSP) for the QNX device might be missing some of the following applications that Qt Creator needs to run, debug, and stop applications on QNX devices: `awk`, `grep`, `kill`, `netstat`, `print`, `printf`, `ps`, `read`, `sed`, `sleep`, `uname`, `slog2info`, and `cat`.

Running Python Projects

You can execute Qt for Python applications directly from Qt Creator. If you used the [new project wizard](#) to create the application project, the `main.py` file is automatically executed when you select the **Run** button.

You can specify another file to execute in the [run settings](#) of the project.

[< Building for Multiple Platforms](#)

[Deploying to Devices >](#)

© 2022 The Qt Company Ltd. Documentation contributions included herein are the copyrights of their respective owners. The documentation provided herein is licensed under the terms of the [GNU Free Documentation License version 1.3](#) as published by the Free Software Foundation. Qt and respective logos are trademarks of The Qt Company Ltd in Finland and/or other countries worldwide. All other trademarks are property of their respective owners.



Contact Us

Company

[About Us](#)

[Investors](#)

[Newsroom](#)

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

- Contribute to Qt
- Forum
- Wiki
- Downloads
- Marketplace