

Connecting Boot2Qt Devices

You can connect **Boot2Qt** devices to the development PC to run, debug, and analyze applications built for them from Qt Creator.

If you have a tool chain for building applications for Boot2Qt devices installed on the development PC, you can add it to Qt Creator. You can then select a **kit** with the **Boot2Qt** device type to build applications for and run them on the devices.

To be able to run and debug applications on Boot2Qt devices, you must add devices and select them in the Qt Creator **kit**.

Enabling the Boot2Qt Plugin

To enable the Boot2Qt plugin:

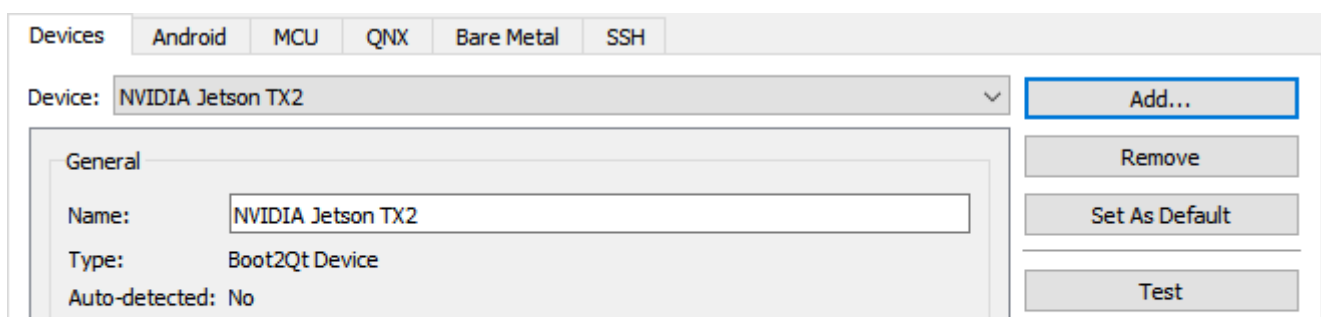
1. Select **Help > About Plugins > Device Support > Boot2Qt** to enable the plugin.
2. Select **Restart Now** to restart Qt Creator and load the plugin.

Adding Boot2Qt Devices

You use a wizard to create the connections. You can use either a network connection or a USB connection. If Qt Creator does not automatically detect a device you connected with USB, you can use a wizard to create a network connection to the device.

Note: On Ubuntu Linux, the development user account must have access to plugged in devices. To allow the development user access to the device via USB, create a new udev rule, as described in [Boot2Qt: Setting Up USB Access to Embedded Devices](#).

You can edit the settings later in **Edit > Preferences > Devices > Devices**.



The screenshot shows the 'Machine type' configuration window in Qt Creator. The 'Machine type' is set to 'Physical Device'. The 'Authentication type' has two options: 'Default' (selected) and 'Specific key'. The 'Host name' is '10.10.10.10', 'SSH port' is '22', and 'Check host key' is checked. 'Free ports' are '10000-10100' and 'Timeout' is '10s'. The 'Username' is 'root'. The 'Private key file' is empty, with 'Browse...' and 'Create New...' buttons. The 'GDB server executable' is 'Leave empty to look up ...'. On the right, there are two buttons: 'Reboot Device' and 'Restore Default App'.

You can protect the connections between Qt Creator and a device by using an [OpenSSH](#) connection. OpenSSH is a connectivity tool for remote login using the SSH protocol. The OpenSSH suite is not delivered with Qt Creator, so you must download it and install it on the development PC. Then, you must configure the paths to the tools in Qt Creator. For more information, see [Configuring SSH Connections](#).

You need either a password or an SSH public and private key pair for authentication. If you do not have an SSH key, you can use the ssh-keygen tool to create it in Qt Creator. For more information, see [Generating SSH Keys](#).

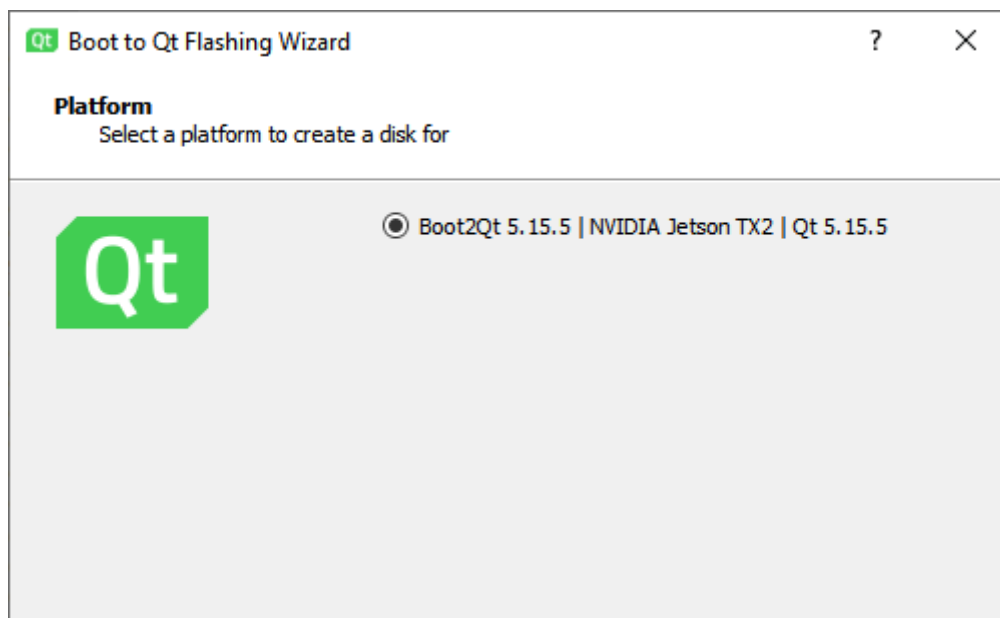
Note: Qt Creator does not store passwords. If you use password authentication, you may need to enter the password on every connection to the device, or, if caching is enabled, at every Qt Creator restart.

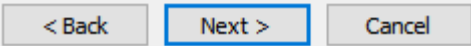
To reboot the selected device, select **Reboot Device**.

To restore the default application to the device, select **Restore Default App**.

Flashing Boot2Qt Devices

To flash the Boot2Qt image to an SD card with Flashing Wizard, select **Tools > Flash Boot to Qt Device** and follow the instructions of the wizard.

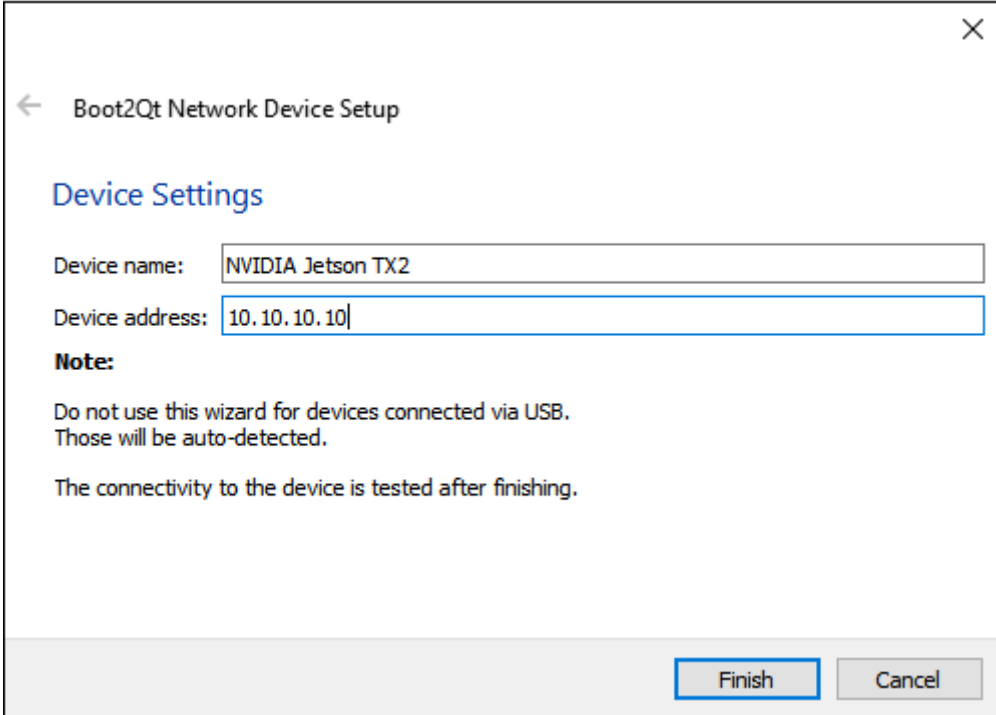




Configuring Connections

To configure connections between Qt Creator and a Boot2Qt device and to specify build and run settings for the device:

1. Make sure that your device can be reached via an IP address or connect it with a USB connection.
2. Select **Edit > Preferences > Kits > Qt Versions > Add** to add the Qt version for Boot2Qt.
3. Select **Edit > Preferences > Kits > Compilers > Add** to add the compiler for building the applications.
4. Select **Tools > Flash Boot to Qt Device** to flash the Boot2Qt image to an SD card with Flashing Wizard.
5. To deploy applications and run them remotely on devices, specify parameters for connecting to the devices over the network (Qt Creator automatically detects devices connected with USB):
 1. Select **Edit > Preferences > Devices > Devices > Add > Boot2Qt > Finish**.



2. In the **Device name** field, enter a name for the connection.
3. In the **Device address** field, enter the host name or IP address of the device. This value will be available in the `%{Device:HostAddress}` variable.
4. Click **Next** to create the connection.

You can edit the connection parameters in the **Devices** tab. The wizard does not show parameters that have sensible default values. One of these is the SSH port number, which is available in the variable `%{Device:SshPort}`.

6. Select **Edit > Preferences > Kits > Add** to add a kit for building applications for the device. Select the Qt version, compiler, and device that you added above, and choose **Boot2Qt** as the device type.



2. Select **Projects** > **Build & Run** to enable the kit that you specified above.

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

When you run the project, Qt Creator deploys the application as specified by the deploy steps. By default, Qt Creator copies the application files to the device. For more information, see [Specifying Run Settings for Boot2Qt Devices](#).

[< Connecting Bare Metal Devices](#)

[Adding Docker 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
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

Contribute to Qt
Forum
Wiki

