

# Connecting Generic Remote Linux Devices

You can connect generic Linux 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 embedded Linux devices installed on the development PC, you can add it to Qt Creator. You can then select a [kit](#) with **Embedded Linux** device type to build applications for and run them on the devices.

To be able to run and debug applications on generic remote Linux devices, you must add devices and select them in the Qt Creator [kit](#).

You use a wizard to create the connections. You can edit the settings later in **Edit > Preferences > Devices > Devices**.

The screenshot shows the 'Devices' dialog box in Qt Creator. At the top, there are tabs for different device types: Devices, Android, MCU, WebAssembly, QNX, Bare Metal, and SSH. The 'Device' dropdown is set to 'Generic Linux Device (default for Generic Linux)'. To the right of the dropdown are buttons for 'Add...', 'Remove', 'Set As Default', 'Test', 'Show Running Processes...', and 'Deploy Public Key...'. The 'General' tab is active, showing fields for Name (Generic Linux Device), Type (Generic Linux), Auto-detected (No), and Current state (Unknown). The 'Type Specific' tab is also visible, showing fields for Machine type (Physical Device), Authentication type (Default), Host name (10.10.10.15), SSH port (22), Free ports (10000-10100), Timeout (10s), Username (jsmith), Private key file (Browse...), and GDB server executable (Leave empty t...). The 'Add...' button is highlighted with a blue border.

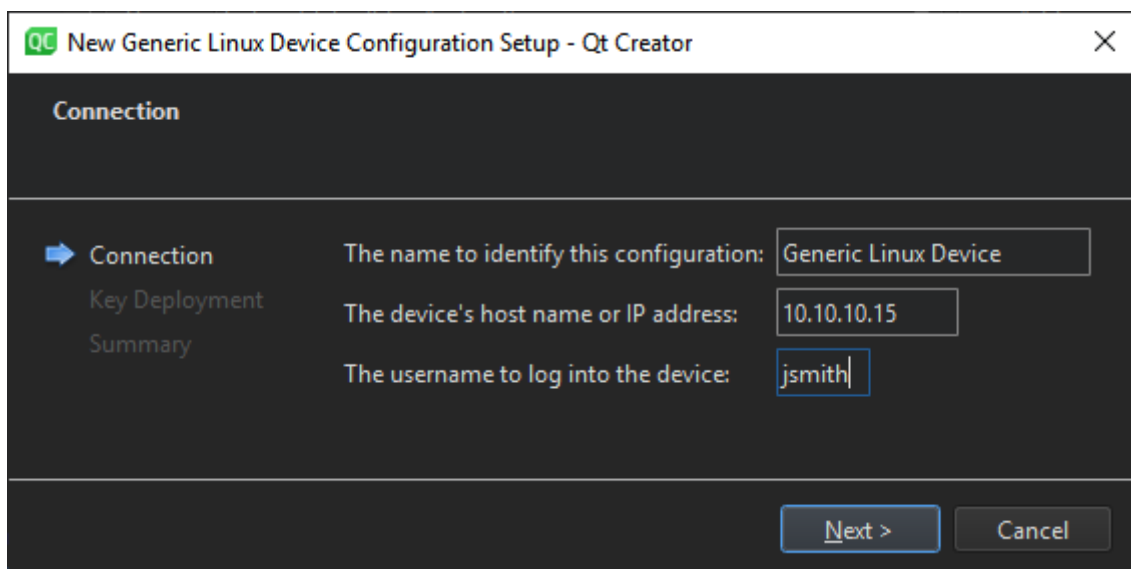
You can protect the connections between Qt Creator and a device by using an [OpenSSH](#) connection. OpenSSH is a connectivity tool for remote login using the SSH protocol. The OpenSSH suite is not delivered with Qt Creator, so you must download it and install it on the development PC. Then, you must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

you can use the `ssh-keygen` tool to create it in Qt Creator. For more information, see [Generating SSH Keys](#).

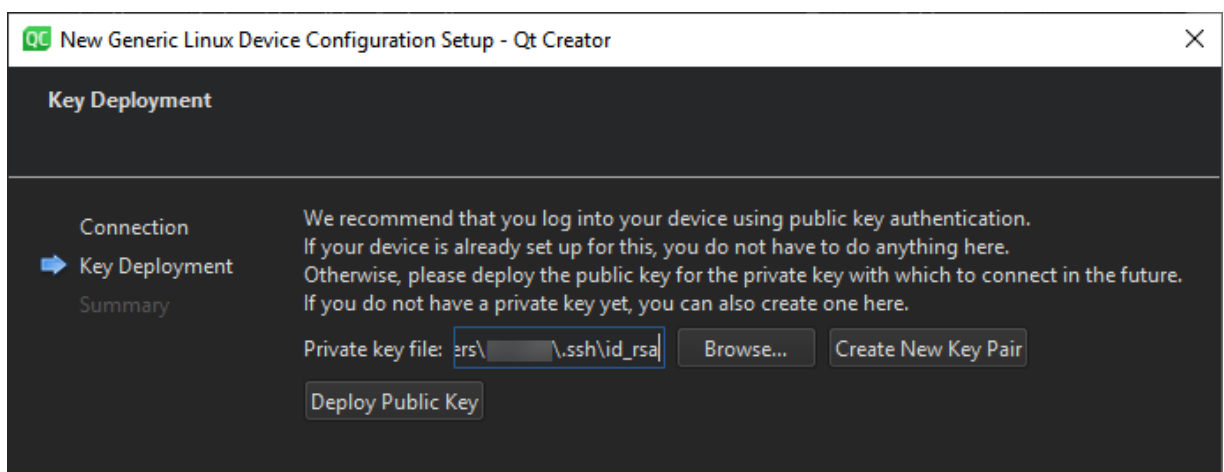
**Note:** Qt Creator does not store passwords, so if you use password authentication, you may need to enter the password on every connection to the device, or, if caching is enabled, at every Qt Creator restart.

To configure connections between Qt Creator and an embedded Linux device and to specify build and run settings for the device:

1. Make sure that your device can be reached via an IP address.
2. Select **Edit > Preferences > Kits > Qt Versions > Add** to add the Qt version for embedded Linux.
3. Select **Edit > Preferences > Kits > Compilers > Add** to add the compiler for building the applications.
4. To deploy applications and run them remotely on devices, specify parameters for accessing the devices:
  - A. Select **Edit > Preferences > Devices > Devices > Add > Generic Linux Device > Start Wizard**.



- B. In the **The name to identify this configuration** field, enter a name for the connection.
- C. In the **The device's host name or IP address** field, enter the host name or IP address of the device. This value will be available in the variable `%{Device:HostAddress}`.
- D. In the **The username to log into the device** field, enter the username to log into the device and run the application as. This value will be available in the variable `%{Device:UserName}`.
- E. Select **Next** to open the **Key Deployment** dialog.



- F. In **Private key file**, select a private key file to use for authentication. This value will be available in the variable `{Device:PrivateKeyFile}`.
- G. If you do not have a public-private key pair, select **Create New Key Pair**. For more information, see [Generating SSH Keys](#).
- H. Select **Deploy Public Key** to copy the public key to the device.
- I. Select **Next** to create the connection.

All of these parameters can be edited later, as well as additional ones that the wizard does not show because there are sensible default values. One of these is the SSH port number, which is available in the variable `{Device:SshPort}`.

5. Select **Edit > Preferences > Kits > Add** to add a kit for building for the device. Select the Qt version, compiler, and device that you added above, and choose **Generic Linux Device** for the device type.
6. 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.
7. Select **Run** to specify run settings. Usually, you can use the default settings.

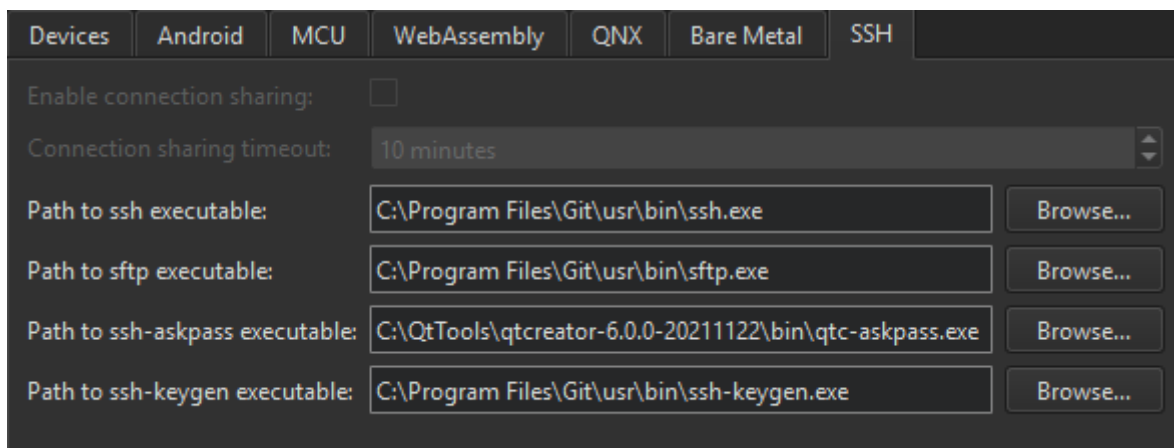
When you run the project, Qt Creator deploys the application as specified by the deploy steps. By default, Qt Creator copies the application files to the device. For more information, see [Deploying Applications to Generic Remote Linux Devices](#).

## Configuring SSH Connections

SSH connections are established via an OpenSSH client running in master mode, if possible. Connection sharing is enabled by default to allow sharing multiple sessions over a single SSH connection. This way, a connection is only established once and then re-used by subsequent run and deploy procedures, saving connection setup overhead particularly with embedded devices. Because connection sharing is not supported on Windows, a new SSH connection is created for each deploy or run procedure.

To create SSH connections, you must install the [OpenSSH](#) suite, which includes the `ssh`, `sftp`, and `ssh-keygen` tools on the development PC.

To tell Qt Creator where it can find the tools, specify the paths to the directories where the tools are installed in **Edit > Preferences > Devices > SSH**:



- Deselect the **Enable connection sharing** check box to create a new SSH connection for each deploy and run procedure. This option is grayed on Windows, where connection sharing is not supported.

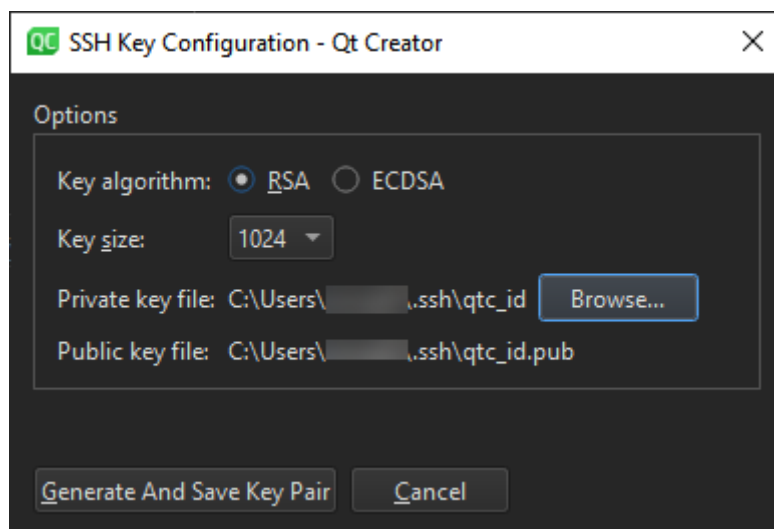
- › In the **Path to sftp executable** field, enter the path to the directory where the SFTP executable is installed.
- › In the **Path to ssh-askpass executable** field, enter the path to the directory where the ssh-askpass executable is installed. Usually, you can use the default path that points to the implementation of the tool delivered with Qt Creator, qtc-askpass.
- › In the **Path to ssh-keygen executable** field, enter the path to the directory where the ssh-keygen executable is installed.

## Generating SSH Keys

If you do not have an SSH public and private key pair, you can generate it in Qt Creator. The connection wizard can create the key pair for you, or you can create it separately.

You can specify key length and the key algorithm, RSA or ECDSA. If you only use the keys to protect connections to the emulator or device, you can use the default values.

1. Select **Edit > Preferences > Devices > Devices > Create New**.



2. In the **Private key file** field, select the location to save the private key.  
The **Public key file** field displays the location to save the corresponding public key.
3. Select **Generate And Save Key Pair** to generate and save the keys at the specified locations.

## Managing Device Processes

You can view processes running on devices and kill them. Select **Edit > Preferences > Devices > Devices > Show Running Processes**.

You can filter the processes by name or ID in the **List of Processes** dialog.

To update the process list, select **Update List**.

To kill a process, select it in the list, and then select **Kill Process**.



Contact Us

Company

- About Us
- Investors
- Newsroom
- Careers
- Office Locations

Support

- Support Services
- Professional Services
- Partners
- Training

Community

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

Licensing

- Terms & Conditions
- Open Source
- FAQ

For Customers

- Support Center
- Downloads
- Qt Login
- Contact Us
- Customer Success