Qt DOCUMENTATION

≡

Qt Creator Manual 8.0.2

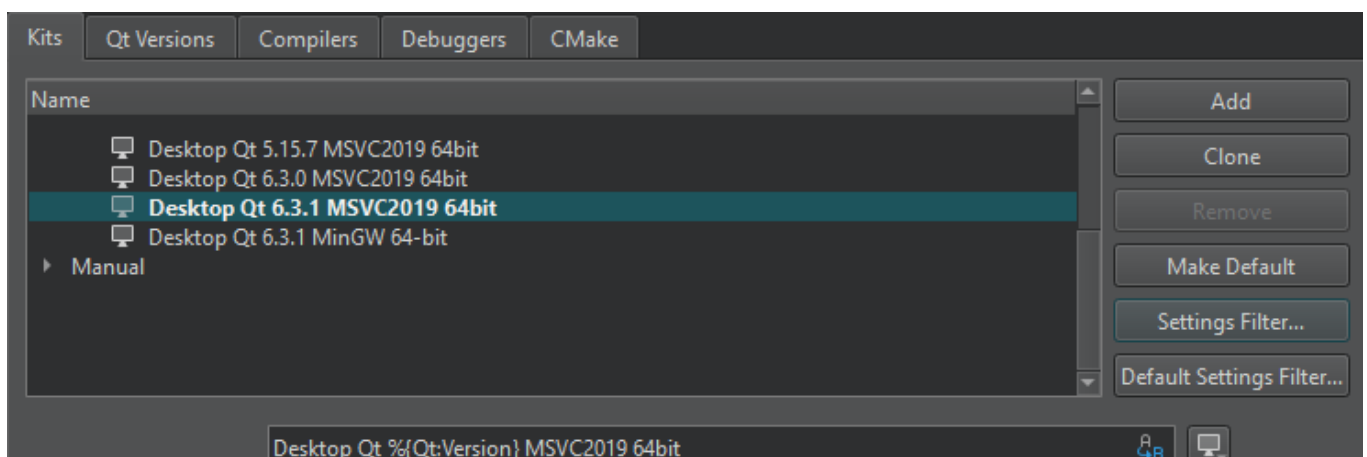[Search]   Topics ›

Adding Kits

Qt Creator groups settings used for building and running projects as kits to make cross-platform and cross-configuration development easier. Each kit consists of a set of values that define one environment, such as a device, compiler, Qt version, and debugger command to use, and some metadata, such as an icon and a name for the kit. Once you have defined kits, you can select them to build and run projects.

Qt Creator supports development for the desktop and for the following types of devices:

> Android Device

> Bare Metal Device

> Boot2Qt Device (commercial only)

> Boot2Qt Emulator Device (commercial only)

> Generic Remote Linux Device

> iOS Device

> iOS Simulator

> MCU Device (commercial only)

> QNX Device

> WebAssembly Runtime

Filtering Kit Settings

Typically, only a subset of the kit settings is relevant for a particular setup. Therefore, Qt Creator plugins register sets of relevant settings that you can view and modify in **Edit** > **Preferences** > **Kits**. For example, if you use CMake to build all your projects, you can hide Qbs and qmake settings by default.

To hide and show settings in the **Kits** tab for the current kit, select **Settings Filter**. To view and modify the settings displayed when you add a new kit, select **Default Settings Filter**.

Specifying Kit Settings

To add kits:

1. Select **Edit** > **Preferences** > **Kits** > **Add**.

   To clone the selected kit, select **Clone**.

2. Specify kit settings. The settings to specify depend on the build system and device type.

3. Select **OK** to create the kit.

Qt Creator uses the *default kit* if it does not have enough information to choose the kit to use. To set the selected kit as the default kit, select **Make Default**.

Kit Settings

The following table summarizes the available kit settings.

Setting	Value
Name	Name of the kit. You can use variables to generate the kit name based on the values you set in the other fields.
🖥	Image to use as an icon for the kit.
File system name	Name for the kit to use as a part of directory names. This value is used for the `CurrentKit:FileSystemName` variable, which determines the name of the shadow build directory, for example.
Device type	Type of the device. Double-click the icon next to the field to select the image that is displayed in the kit selector for this kit. You can use any image in a supported file format (for example, PNG). The image is scaled to the size 64x64 pixels. For example, using the compiler logo as an icon allows you to easily see, which compiler is used to build the project for the selected kit.
Setting	**Value**

Qt DOCUMENTATION

Sysroot	Directory where the device image is located. If you are not cross-compiling, leave this field empty.
Emulator skin	Skin to use for the Boot2Qt Emulator Device.
Compiler	C or C++ compiler that you use to build the project. You can add compilers to the list if they are installed on the development PC, but were not detected automatically. For more information, see Adding Compilers. This setting is used to tell the code model which compiler is used. If your project type and build tool support it, Qt Creator also tells the build tool to use this compiler for building the project.
Environment	Select **Change** to modify environment variable values for build environments in the **Edit Environment Changes** dialog. For more information about how to add and remove variable values, see Batch Editing.
Force UTF-8 MSVC compiler output	Either switches the language of MSVC to English or keeps the language setting and just forces UTF-8 output, depending on the MSVC compiler used.
Debugger	Debugger to debug the project on the target platform. Qt Creator automatically detects available debuggers and displays a suitable debugger in the field. You can add debuggers to the list. For more information, see Adding Debuggers. For Android kits, the **Android GDB server** field will display the path to GDB server executable.
Qt version	Qt version to use for building the project. You can add Qt versions to the list if they are installed on the development PC, but were not detected automatically. For more information, see Adding Qt Versions. Qt Creator checks the directories listed in the PATH environment variable for the qmake executable. If a qmake executable is found, it is referred to as **Qt in PATH** and selected as the Qt version to use for the **Desktop** kit that is created by default.
Qt mkspec	Name of the mkspec configuration that should be used by qmake. If you leave this field empty, the default mkspec of the selected Qt version is used.
Additional Qbs profile settings	Select **Change** to add settings to Qbs build profiles. For more information, see Editing Qbs Profiles.
CMake Tool	CMake executable to use for building the project. Select **Manage** to add installed CMake executables to the list. For more information, see Adding CMake Tools.
CMake generator	Select **Change** to edit the CMake Generator to use for producing project files. Only the generators with names beginning with the string **CodeBlocks** produce all the necessary data for the Qt Creator code model. Qt Creator displays a warning if you select a generator that is not supported. For more information, see Using Ninja as a CMake Generator.
CMake configuration	Select **Change** to edit the parameters of the CMake configuration for the kit.
Meson tool	Meson tool to use for building the project. Select **Manage** to add installed Meson tools to the list. For more information, see Adding Meson Tools.
Ninja tool	Ninja tool to use for building the project with Meson. Select **Manage** to add installed Ninja tools to the list.

Editing Qbs Profiles

Qt DOCUMENTATION
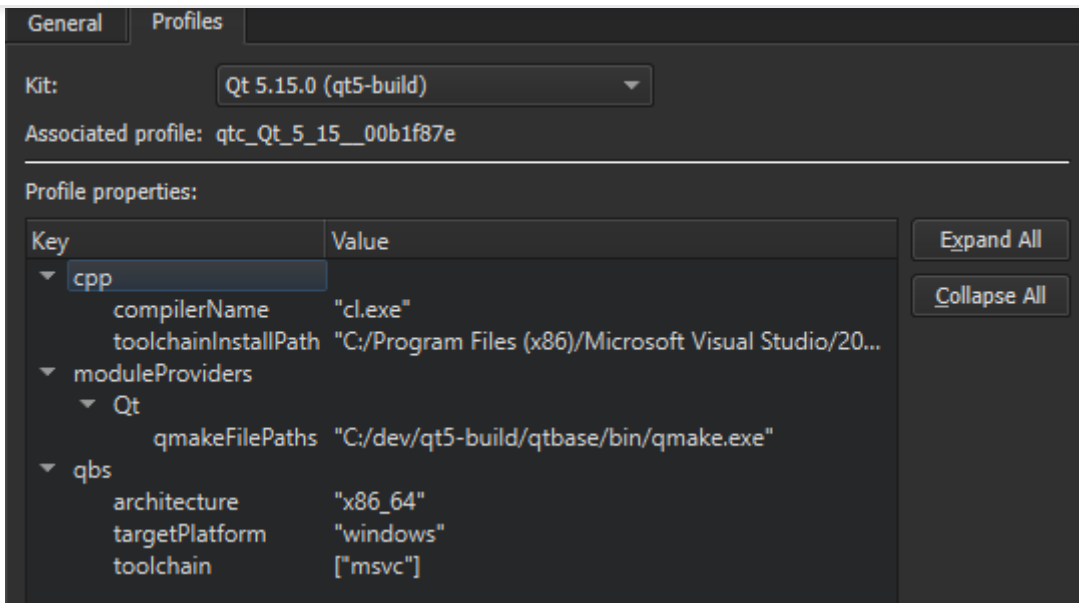

You can add keys and values to the profile or remove them from it, as well as modify existing values. For a list of available keys and values, see List of Modules in the Qbs Manual.

To edit the Qbs profile associated with the kit:

1. In **Edit** > **Preferences** > **Kits**, select the kit, and then select **Change** next to the **Additional Qbs Profile Settings** field to open the **Custom Properties** dialog.
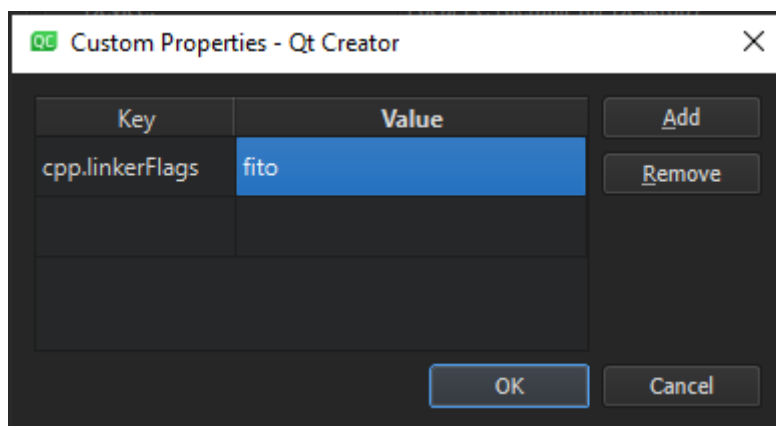
2. Double-click an empty cell in the **Key** column to specify the key to add or modify as: `<module_name>.<property_name>`.

3. Double-click the cell on the same row in the **Value** column to specify a value as a JSON literal.

4. Select **Add** to add the key-value pair.

5. Click **OK**.

To modify an existing value, double-click it in the **Value** field.

To remove the selected property, select **Remove**.

‹ Configuring Projects   Adding Qt Versions ›

DOCUMENTATION

The Qt Company

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

© 2022 The Qt Company

Feedback   Sign In