

[Qt 创建者手册](#) > [设置](#)

设置

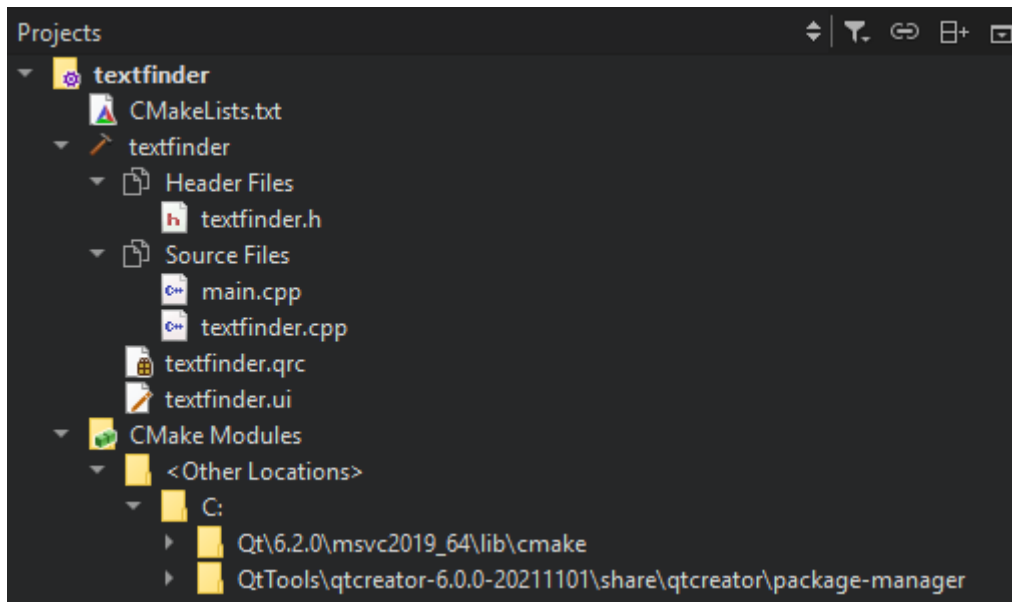
CMake 可自动配置生成系统。它通过使用简单的配置文件（称为文件）来控制软件编译过程。CMake 会生成本机生成配置和工作区，你可以在所选的编译器环境中使用这些配置和工作区。CMakeLists.txt

您可以使用Qt Creator的CMake为桌面以及移动和嵌入式设备构建应用程序。您还可以构建单个文件来测试您的更改。

Qt 创建器会自动检测在中指定的 CMake 可执行文件。可以向其他 CMake 可执行文件添加路径，并在不同的生成和运行工具包中使用它们。PATH

安装 CMake 时，将以 Qt 帮助文件格式（.qch）安装文档。它由Qt Creator自动注册，您可以在帮助模式下查看它。

Qt 创建器会在您编辑项目中的配置文件时自动运行 CMake 以刷新项目信息。生成项目时，项目信息也会自动刷新。CMakeLists.txt



如果 Qt Creator 无法加载 CMake 项目，则“项目”视图将显示<文件系统>项目节点，以避免扫描文件系统并更快地加载项目。该节点显示与“文件系统”视图相同的文件。选择“生成>清除 CMake 配置”，然后选择“生成>运行 CMake”以重新配置项目。

添加制作工具

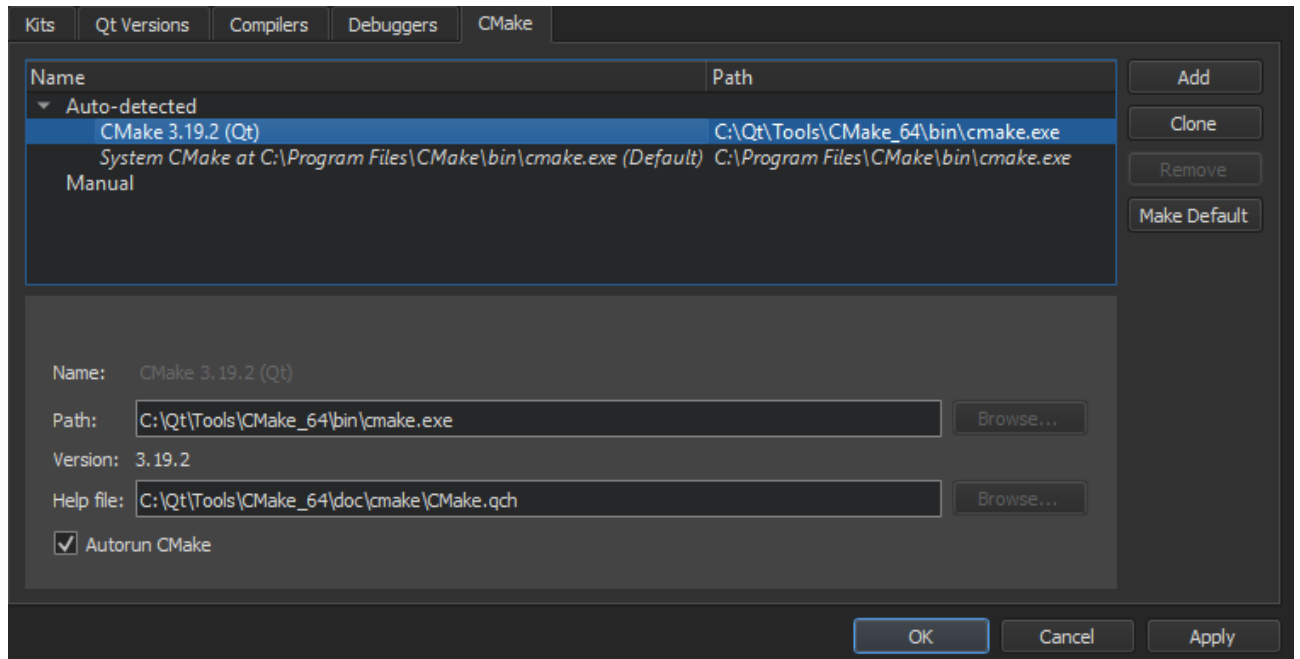
Qt 创建者需要 CMake 基于文件的 API，因此你需要 CMake 版本 3.14 或更高版本。

对于具有较旧版本的 CMake 的系统，只有解决方法可用：

- › 使用 qmake 构建创建一个临时项目文件，并在 Qt 创建器中**打开**该文件。请注意，如果没有进一步的手动更改，这通常不可编译。qmake -project
- › 手动为通用项目创建一个临时**项目**文件，并在 Qt Creator 中打开该文件。请注意，如果没有进一步的手动更改，这通常是不可编译的。

要查看和指定 CMake 的设置，请执行以下操作：

1. 选择“**编辑>首选项>CMake**”工具包”。



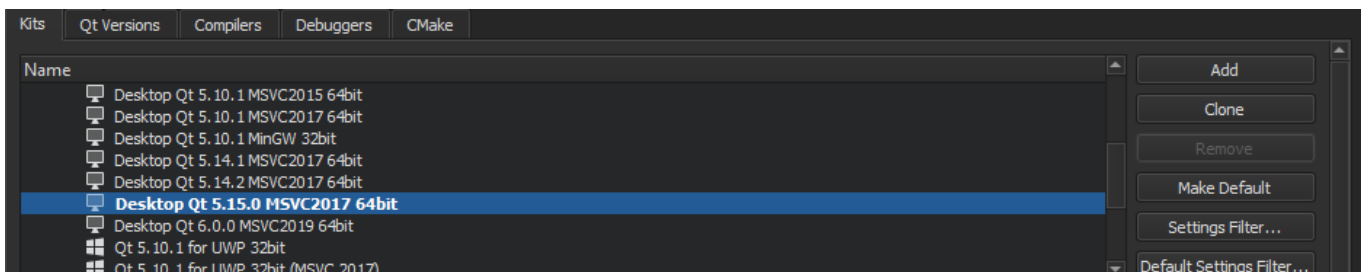
2. “**名称**”字段显示 CMake 安装的名称。
3. “**路径**”字段显示 CMake 可执行文件的路径。
4. “**帮助文件**”字段显示由 CMake 提供并随 CMake 一起安装的 CMake 帮助文件（.qch）的路径。
5. 如果不希望每次保存对文件的更改时都自动运行 CMake，请取消选中“**自动运行 CMake**”复选框。
6. 选择“**应用**”以保存更改。

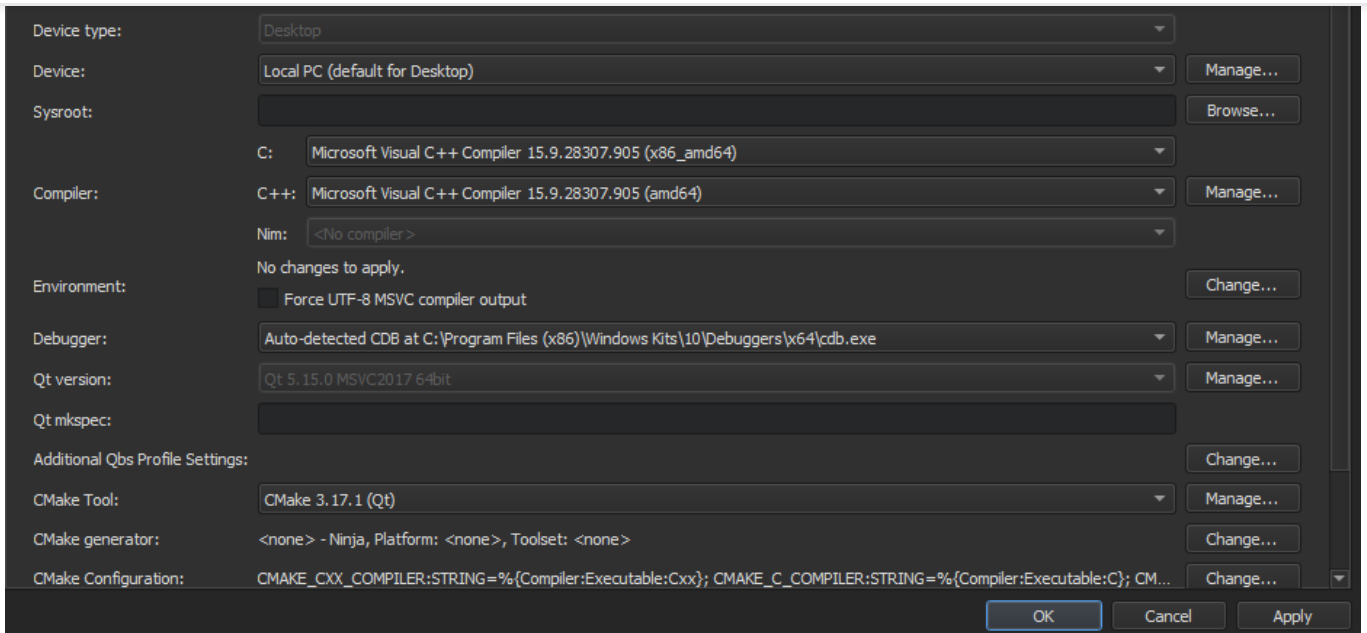
若要向 Qt 创建程序未自动检测到的 CMake 可执行文件添加路径，并为其指定设置，请选择“**添加**”。若要对自动检测到的安装进行更改，请选择“**克隆**”。

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

To remove the selected CMake executable from the list, select **Remove**.

Select the **Kits** tab to add the CMake tool to a build and run kit. The kit also specifies the CMake generator that is used for producing project files for Qt Creator and the initial configuration parameters:





For more information, see [Adding Kits](#).

Editing CMake Configuration Files

To open a CMakeLists.txt file for editing, right-click it in the **Projects** view and select **Open With > CMake Editor**.

You can also use the filter in the [locator](#) to open the CMakeLists.txt file for the current run configuration in the editor. This is the same build target as when you select **Build > Build for Run Configuration.cmo**

The following features are supported:

- › Pressing **F2** when the cursor is on a filename to open the file
- › Keyword completion
- › Code completion
- › Path completion
- › Auto-indentation
- › Matching parentheses and quotes

Warnings and errors are displayed in [Issues](#).

Adding External Libraries to CMake Projects

Through external libraries, Qt Creator can support code completion and syntax highlighting as if they were part of the current project or the Qt library.

Qt Creator detects the external libraries using the macro. Some libraries come with the CMake installation. You can find those in the directory of your CMake installation. For more information, see [cmake-packages\(7\).find_package\(\)](#) Modules

Syntax completion and highlighting work once your project successfully builds and links against the external library.

Related Topics



- › CMake Build Configuration
- › Specifying Run Settings
- › Deploying Applications to Generic Remote Linux Devices

[‹ Build Systems](#)[Setting Up Qbs ›](#)

© 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](#)

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](#)

