

Qt 6.4 > Build with CMake > [Building projects on the command line](#)

Building projects on the command line

This page explains how to configure and build existing projects. If you want to know how to create a Qt-based CMake project, see the documentation on [how to get started with CMake](#).

To build a Qt project, CMake needs to know where the Qt installation is located. Usually this is done by setting the CMake variable `CMAKE_PREFIX_PATH` to Qt's installation prefix. If you are cross-compiling, see [Cross-compiling](#) for details on additional variables you will need to set.

If Qt is installed using the online installer, choose a Qt version within the top-level installation directory. For example, the following command shows how this is done on Windows:

```
cmake -DCMAKE_PREFIX_PATH=C:\Qt\6.4.0\msvc2019_64 -S <source-dir> -B <build-dir>
```

The `<source-dir>` and `<build-dir>` placeholders represent the source and build directories of your project.

发电机

CMake 会生成必要的构建系统文件，使 GNU 生成或忍者等构建工具能够构建您的项目。

Topics >

若要在所有平台上获得一致的开发人员体验，请使用 `或` 生成器。NinjaNinja Multi-Config

您可以通过设置环境变量或使用参数来选择 CMake 生成器：CMAKE_GENERATOR-G

```
cmake -G Ninja ...
```

qt-cmake

该脚本是配置项目的便捷替代方法。这样就无需指定 `.您可以在Qt安装前缀的目录中找到它。该脚本将所有参数传递给 CMake，因此您可以像使用一样使用它：qt-cmakeCMAKE_PREFIX_PATHbincmake\`

```
C:\Qt\6.4.0\msvc2019_64\bin\qt-cmake -G Ninja -S <source-dir> -B <build-dir>
```

生成生成生成系统文件后，即可生成项目：

```
cd <build-dir>
ninja
```

```
cmake --build <build-dir>
```

交叉编译

为不同于开发计算机的平台生成项目称为交叉编译。例如，在 Windows 计算机（主机平台）上为安卓（目标平台）进行构建。

对于大多数平台，使用 CMake 进行交叉编译需要一个[工具链文件](#)。除了目标平台的 Qt 版本外，它还需要开发主机的 Qt 版本。例如，您需要安装适用于 Windows 的 Qt 和适用于安卓的 Qt 才能在视窗上为安卓进行交叉编译。

从目标平台的 Qt 安装中使用，为该平台交叉编译项目：qt-cmake

```
<target-qt>/bin/qt-cmake -S <source-dir> -B <build-dir>
```

这将为目标平台配置项目。工具链文件会自动传递，并且可能还会设置其他特定于平台的变量。

指定自定义工具链文件

该脚本将 Qt 内部[工具链文件](#)传递给 CMake。此工具链文件设置了几个特定于Qt目标平台的变量。qt-cmake 如果您使用的是尚未在计算机上构建的 Qt 安装，则需要知道目标平台的 CMake 工具链文件的位置。qt-cmake 在这种情况下，您可以指示通过设置变量来链接加载自定义工具链文件：qt-cmakeQT_CHAINLOAD_TOOLCHAIN_FILE

```
~/Qt/6.4.0/android_armv7/bin/qt-cmake -DQT_CHAINLOAD_TOOLCHAIN_FILE=<file-path> -S <source-dir> -B <build-dir>
```

这指示Qt的内部工具链文件也加载您的自定义工具链文件。

< 开始使用

导入的目标 >

©2022 Qt Ltd. 此处包含的文档贡献是其各自所有者的版权。此处提供的文档是根据自由软件基金会发布的 [GNU 自由文档许可证 1.3](#) 版的条款进行许可的。Qt及其相应的徽标是Qt有限公司在芬兰和/或全球其他国家的[商标](#)。所有其他商标均为其各自所有者的财产。



联系我们

公司

关于我们
投资者
编辑部
职业
办公地点

发牌

条款及细则
开源
常见问题

专业服务
合作 伙伴
训练

下载
秦特登录
联系我们
客户成功案例

社区

为Qt做贡献
论坛
维基
下载
市场

© 2022 Qt公司

[反馈](#) [登录](#)