

🔍 搜索

Qt创作者手册 > [使用命令行选项](#)

使用命令行选项

您可以启动Qt Creator并从命令行指定一些选项。例如，您可以打开任何行和列的文件。

要指定命令行选项，请在Qt Creator安装或构建目录中输入以下命令：

```
qtcreeator [option] [filename[:line_number[:column_number]]]
```

注意：您可以使用冒号 (:)或加号 (+) 作为文件名和行号以及行号和列号之间的分隔符。您还可以在分隔符和行号之间使用空格。

Qt创建者手册8.0.2
Topics >

- > C:\qtcreeator\bin>qtcreeator C:\TextFinder\textfinder.cpp:100:2
- > C:\qtcreeator\bin>qtcreeator C:\TextFinder\textfinder.cpp +100+2

在 macOS 上：

- > Qt\ Creator.app/Contents/MacOS/Qt\ Creator -help

若要打开位于特定文件夹中的项目，可以将文件夹名称作为命令行参数传递。Qt Creator 查找与文件夹名称匹配的会话并加载它。或者，它会在文件夹中查找项目文件并将其打开。例如：

```
qtcreeator .
```

注意：要在Windows上从命令行运行自建Qt Creator，请确保Qt安装目录包含在PATH环境变量中。您可以在命令行中输入以下命令，将 Qt 添加到路径中：

```
set PATH=<Qt_installation_directory>\mingw\bin;c:<Qt_installation_directory>\bin;%PATH%
```

下表总结了可用选项：

选择	描述
----	----

-客户	尝试连接到已在运行的Qt Creator实例。
-pid	尝试使用指定的进程ID连接到已在运行的Qt Creator实例。
-块	在正在运行的Qt Creator实例中的编辑器中打开文件，并阻止命令行，直到第一个编辑器关闭。
-无崩溃检查	禁用以前崩溃的Qt Creator实例的启动检查。
-加载<插件>	启用指定的插件及其依赖的所有插件。您可以组合和选项并多次指定这两个选项以启用和禁用多个插件。操作按指定顺序执行。-load-noload
-全部加载	启用所有插件。
-noload <插件>	禁用指定的插件和依赖于它的所有插件。
-全部不加载	禁用所有插件。
-轮廓	输出有关插件启动和关闭的分析数据。
-pluginpath <path>	添加Qt Creator查找插件的路径。要指定多个路径，请为每个路径添加选项。-pluginpath
-设置路径<路径>	覆盖存储用户设置的默认路径。
-安装设置路径<路径>	覆盖从中读取与用户无关的设置的默认路径（例如，由安装程序编写）。
-临时清洁设置， -TCS	出于调试或测试原因使用干净的设置。当Qt Creator退出时，这些设置将被删除。
-test <plugin> [, testfunction[: testdata]] ...	对于Qt Creator插件开发人员：默认情况下使用单独的设置路径运行插件的测试。
-全部测试	对于Qt Creator插件开发人员：从所有插件运行测试。
-notest <插件>	对于Qt Creator插件开发人员：从测试运行中排除插件的所有测试。
-方案<方案名称>	对于Qt Creator插件开发人员：运行指定的场景。
-颜色<颜色>	核心插件：覆盖选定的 UI 颜色。
-演示模式	核心插件：按下键盘快捷键时将其显示为弹出窗口。在向其他人展示Qt Creator时最有用。
-主题<默认 黑暗>	核心插件：将深色主题应用于Qt Creator，而无需使用样式表。
-诺图尔	欢迎插件：启动时跳过 UI 教程。
-调试 <pid>	调试器插件：使用给定的进程 ID 附加到进程。
-debug <可执行文件>[, kit=<kit>]	调试器插件：使用名称启动和调试可执行文件。可以通过 ID 或名称指定 A，以指向非默认调试器和系统根。executablekit
-debug [可执行文件,]core=<corefile>[, kit=<kit>]	调试器插件：加载名为的核心文件。该参数指定生成核心文件的可执行文件。如果省略此参数，Qt Creator将尝试从核心文件本身重建它。对于超过 80 个字符的路径，此操作将失败。在这种情况下，参数是强制性的。可以通过 ID 或名称指定 A，以指向非默认调试器和系统根。corefileexecutableexecutablekit

<server: port> [, kit=<kit>]	
-wincrashevent <event-handle: pid>	调试器插件：使用指定的事件句柄和进程 ID 附加到崩溃的进程。
-git-show <git commit hash>	Git 插件：显示指定的提交哈希。
-自定义向导- 详细	项目资源管理器插件：在加载自定义向导时显示其他信息。有关自定义向导的详细信息，请参阅 添加新的自定义向导
-ensure-kit-for-binary <二进制路径>	ProjectExplorer插件：创建一个工具包，其中包含与给定二进制文件的体系结构相对应的工具链。
-上届会议	项目资源管理器插件：在Qt Creator启动时加载最后一个会话。打开上次退出Qt Creator时打开的项目和文件。有关管理会话的详细信息，请参阅 管理会话 。
<会话>	项目资源管理器插件：在Qt Creator启动时加载给定的会话。打开上次退出Qt Creator时打开的项目和文件。有关管理会话的详细信息，请参阅 管理会话 。

使用自定义样式

Qt Creator是一个Qt应用程序，因此，它接受所有Qt应用程序都接受的命令行选项。例如，您可以使用theoptions 来应用自定义样式和[样式表](#)。样式仅在当前会话期间应用。-style-stylesheet

应用样式时要小心，因为覆盖现有样式可能会使某些项目难以看到。此外，设置样式表可能会影响[文本编辑器配色方案](#)和集成Qt Designer的样式。

您还可以切换到深色主题以自定义小部件、颜色和图标的外观，而无需使用样式表。

◀ 设置柯南

键盘快捷键 ▶

©2022 Qt有限公司 此处包含的文档贡献的版权归 他们各自的所有者。此处提供的文档根据自由软件基金会发布的[GNU自由文档许可证版本 1.3](#)的条款进行许可。Qt和相应的徽标是Qt有限公司在芬兰和/或全球其他国家的商标。所有其他商标均为财产 其各自所有者。



联系我们

公司

关于我们

Qt 5.15.2

发牌

条款和条件

Qt 5.15.2



职业
办公地点

支持

支持服务
专业服务
合作 伙伴
训练

对于客户

支持中心
下载
Qt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](#)