

[Qt 6.4](#) > [Qmake手册](#) > [运行 qmake](#)

运行 qmake

qmake 在运行时可以通过在命令行上指定各种选项来自定义它的行为。这些允许对生成过程进行微调，提供有用的诊断信息，并可用于指定项目的目标平台。

命令语法

用于运行 qmake 的语法采用以下简单形式：

```
qmake [-makefile] [-project] [-file ...]
```

[Topics >](#)

注意：如果你通过包管理器安装了Qt，二进制文件可能是。qmake6

操作模式

Qmake支持两种不同的操作模式。在默认模式下，qmake 使用项目文件中的信息来生成 Makefile，但也可以使用 qmake 生成项目文件。如果要显式设置模式，则必须在所有其他选项之前指定它。可以是以下两个值之一：
mode

- › -makefile
qmake 输出将是一个生成文件。
- › -project
qmake 输出将是一个项目文件。

注意：可能需要编辑创建的文件。例如，添加变量以适应项目所需的模块。QT

您可以使用 来指定常规设置和特定于模式的设置。仅适用于生成文件模式的选项在“生成文件模式选项”部分中介绍，而影响项目文件创建的选项在“项目模式选项”部分中介绍。options

文件

常规选项

可以在命令行上为 qmake 指定各种选项，以便自定义构建过程，并覆盖平台的默认设置。以下基本选项提供有关使用 qmake 的帮助，指定 qmake 写入输出文件的位置，以及控制将写入控制台的调试信息的级别：

- › **-help**
Qmake将介绍这些功能并提供一些有用的帮助。
- › **-o file**
qmake 输出将被定向到。如果未指定此选项，qmake 将尝试为其输出使用合适的文件名，具体取决于其运行的模式。如果指定了“-”，则输出将定向到 stdout。
file
- › **-d**
qmake 将输出调试信息。添加不止一次会增加详细程度。-d

用于项目的模板通常由项目文件中的**TEMPLATE**变量指定。您可以使用以下选项覆盖或修改此设置：

- › **-t tmpl**
qmake将覆盖任何设置变量，但仅在处理完.pro文件之后。TEMPLATEtmpl
- › **-tp prefix**
qmake将添加到变量中。prefixTEMPLATE

可以微调警告信息的级别，以帮助您查找项目文件中的问题：

- › **-Wall**
QMAKE将报告所有已知的警告。
- › **-Wnone**
qmake不会生成任何警告信息。
- › **-Wparser**
qmake只会生成解析器警告。这将提醒您分析项目文件时的常见陷阱和潜在问题。
- › **-Wlogic**
Qmake将警告您的项目文件中的常见陷阱和潜在问题。例如，qmake 将报告列表中多次出现的文件和丢失的文件。

生成文件模式选项

```
qmake -makefile [options] files
```

在生成文件模式下，qmake 将生成一个用于构建项目的生成文件。此外，在此模式下可以使用以下选项来影响项目文件的生成方式：

- › **-after**
qmake 将在指定文件之后处理在命令行上给出的分配。
- › **-nocache**
qmake将忽略该文件。.qmake.cache
- › **-nodepend**
qmake 不会生成任何依赖信息。
- › **-cache file**

qmake将用作平台和编译器信息的路径，并忽略QMAKE_SPEC的值。spec

您也可以在命令行上传递 qmake 赋值。它们在指定的所有文件之前进行处理。例如，以下命令从 test.pro 生成生成文件：

```
qmake -makefile -o Makefile "CONFIG+=test" test.pro
```

但是，可以省略某些指定的选项，因为它们是默认值：

```
qmake "CONFIG+=test" test.pro
```

如果您确定要在指定的文件之后处理变量，则可以传递该选项。指定此选项后，该选项之后的命令行上的所有分配都将推迟到解析指定的文件之后。-after-after

项目模式选项

```
qmake -project [options] files
```

在项目模式下，qmake 会生成一个项目文件。此外，您可以在此模式下提供以下选项：

- -r
qmake将以递归方式查看提供的目录。
- -nopwd
qmake不会在您当前的工作目录中查找源代码。它只会使用指定的。files

在此模式下，参数可以是文件或目录的列表。如果指定了一个目录，它将包含在DEPENDPATH变量中，并且该目录中的相关代码将包含在生成的项目文件中。如果给定了一个文件，它将附加到正确的变量中，具体取决于其扩展名。例如，UI 文件添加到窗体，C++文件添加到源。files

您也可以在此模式下在命令行上传递分配。执行此操作时，这些分配将放在生成的项目文件中的最后。

◀ 构建通用项目类型

平台说明 ▶

©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/或其他国家/地区的商标 全球。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

支持

支持服务
专业服务
合作 伙伴
训练

社区

为Qt做贡献
论坛
维基
下载
市场

发牌

条款和条件
开源
常见问题

对于客户

支持中心
下载
Qt登录
联系我们
客户成功案例