

Qt 6.4 > 质量手册 > [配置 qmake](#)

配置 qmake

性能

qmake 有一个持久化配置系统，它允许你在 qmake 中设置一个属性一次，每次调用 qmake 时都查询它。您可以在 qmake 中设置属性，如下所示：

```
qmake -set PROPERTY VALUE
```

应将适当的属性和值替换为 和 。PROPERTYVALUE

您可以按如下方式从 qmake 中检索此信息：

```
qmake -query PROPERTY
qmake -query #queries all current PROPERTY/VALUE pairs
```

注：除了使用 设置的属性外，还列出了内置属性。qmake -queryqmake -set PROPERTY VALUE

此信息将保存到 [QSets](#) 对象中（这意味着它将存储在不同平台的不同位置）。

以下列表总结了这些属性：built-in

- › QMAKE_SPEC - 在主机构建期间解析并存储在 [QMAKESPEC](#) 变量中的主机的短名称mkspec
- › QMAKE_VERSION - qmake 的当前版本
- › QMAKE_XSPEC - 在目标生成期间解析并存储在 [QMAKESPEC](#) 变量中的目标的短名称mkspec
- › QT_HOST_BINS - 主机可执行文件的位置
- › QT_HOST_DATA - qmake 使用的主机可执行文件的数据位置
- › QT_HOST_LIBS - 主机库的位置
- › QT_HOST_LIBEXECS - 运行时主机库所需的可执行文件的位置
- › QT_HOST_PREFIX - 所有主机路径的默认前缀
- › QT_INSTALL_ARCHDATA - 依赖于一般架构的 Qt 数据的位置
- › QT_INSTALL_BINS - Qt二进制文件（工具和应用程序）的位置

- › QT_INSTALL_DOCS - 文档的位置
- › QT_INSTALL_EXAMPLES - 示例的位置
- › QT_INSTALL_HEADERS - 所有头文件的位置
- › QT_INSTALL_LIBEXEC - 运行时库所需的可执行文件的位置
- › QT_INSTALL_LIBS - 图书馆的位置
- › QT_INSTALL_PLUGINS - Qt插件的位置
- › QT_INSTALL_PREFIX - 所有路径的默认前缀
- › QT_INSTALL_QML - QML 2.x 扩展的位置
- › QT_INSTALL_TESTS - Qt测试用例的位置
- › QT_INSTALL_TRANSLATIONS - Qt 字符串的翻译信息位置
- › QT_SYSCONFIG - 目标构建环境使用的 sysroot
- › QT_VERSION - Qt版本。我们建议您使用 `$$QT.<模块>.version` 变量来查询 Qt 模块特定的版本号。

例如，您可以使用以下属性查询此版本的 qmake 的 Qt 安装：QT_INSTALL_PREFIX

```
qmake -query "QT_INSTALL_PREFIX"
```

可以按如下方式查询项目文件中的属性值：

```
QMAKE_VERS = $$[QMAKE_VERSION]
```

克马克佩克

qmake 需要一个平台和编译器描述文件，其中包含许多用于生成相应生成文件的默认值。标准Qt发行版附带了许多这些文件，位于Qt安装子目录中。mkspecs

环境变量可以包含以下任何内容：QMAKESPEC

- › A complete path to a directory containing a file. In this case qmake will open the file from within that directory. If the file does not exist, qmake will exit with an error.qmake.confqmake.conf
- › The name of a platform-compiler combination. In this case, qmake will search in the directory specified by the subdirectory of the data path specified when Qt was compiled (see [QLibraryInfo::DataPath](#)).mkspecs

Note: The path will be automatically added to the generated Makefile after the contents of the INCLUDEPATH system variable.QMAKESPEC

Cache File

The cache file is a special file qmake reads to find settings not specified in the file, project files, or at the command line. When qmake is run, it looks for a file called in parent directories of the current directory, unless you specify. If

File Extensions

Under normal circumstances qmake will try to use appropriate file extensions for your platform. However, it is sometimes necessary to override the default choices for each platform and explicitly define file extensions for qmake to use. This is achieved by redefining certain built-in variables. For example, the extension used for `moc` files can be redefined with the following assignment in a project file:

```
QMAKE_EXT_MOC = .mymoc
```

The following variables can be used to redefine common file extensions recognized by qmake:

- › `QMAKE_EXT_MOC` modifies the extension placed on included moc files.
- › `QMAKE_EXT_UI` modifies the extension used for *Qt Designer* UI files (usually in `FORMS`).
- › `QMAKE_EXT_PRL` modifies the extension placed on `library dependency` files.
- › `QMAKE_EXT_LEX` changes the suffix used in Lex files (usually in `LEXSOURCES`).
- › `QMAKE_EXT_YACC` changes the suffix used in Yacc files (usually in `YACCSOURCES`).
- › `QMAKE_EXT_OBJ` changes the suffix used on generated object files.

All of the above accept just the first value, so you must assign to it just one value that will be used throughout your project file. There are two variables that accept a list of values:

- › `QMAKE_EXT_CPP` causes qmake to interpret all files with these suffixes as C++ source files.
- › `QMAKE_EXT_H` causes qmake to interpret all files with these suffixes as C and C++ header files.

[◀ Using Precompiled Headers](#)

[Reference ▶](#)

© 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](#)

Licensing

[Terms & Conditions](#)



Careers

Office Locations

Support

- Support Services
- Professional Services
- Partners
- Training

For Customers

- Support Center
- Downloads
- Qt Login
- Contact Us
- Customer Success

Community

- Contribute to Qt
- Forum
- Wiki
- Downloads
- Marketplace

© 2022 The Qt Company

[Feedback](#) [Sign In](#)