搜索

创建项目文件

项目文件包含 qmake 构建应用程序、库或插件所需的所有信息。通常，使用一系列声明来指定项目中的资源，但对简单编程构造的支持使您能够描述不同平台和环境的不同生成过程。

项目文件元素

qmake 使用的项目文件格式可用于支持简单和相当复杂的构建系统。简单的项目文件使用简单的声明性样式，定义标准变量以指示项目中使用的源文件和头文件。复杂项目可以使用控制流结构来微调生成过程。

以下各节介绍项目文件中使用的不同类型的元素。

变量

在项目文件中，变量用于保存字符串列表。在最简单的项目中，这些变量通知 qmake 要使用的配置选项，或提供要在构建过程中使用的文件名和路径。

qmake 在每个项目文件中查找某些变量，并使用这些变量的内容来确定它应该写入 Makefile 的内容。例如，HEADERS和SOURCES变量中的值列表用于告诉 qmake 与项目文件位于同一目录中的头文件和源文件。

变量也可以在内部用于存储临时值列表，并且可以用新值覆盖或扩展现有值列表。

以下代码片段说明了如何将值列表分配给变量：

```
HEADERS = mainwindow.h paintwidget.h
```

变量中的值列表按以下方式扩展：

```
SOURCES = main.cpp mainwindow.cpp \
          paintwidget.cpp
CONFIG += console
```

注意：第一个赋值仅包括与变量在同一行上指定的值。第二个赋值使用反斜杠 (\) 跨行拆分变量中的值。HEADERSSOURCES

段中，被添加到包含的现有值列表中。consoleCONFIG

下表列出了一些常用变量并描述了它们的内容。有关变量及其说明的完整列表，请参阅变量。

变量	内容
配置	常规项目配置选项。
德斯特迪尔	将放置可执行文件或二进制文件的目录。
形式	要由用户界面编译器 （uic） 处理的 UI 文件的列表。
头	生成项目时使用的头 （.h） 文件的文件名列表。
.QT	项目中使用的Qt模块列表。
资源	要包含在最终项目中的资源 （.qrc） 文件的列表。有关这些文件的更多信息，请参阅Qt 资源系统。
来源	生成项目时要使用的源代码文件的列表。
模板	要用于项目的模板。这决定了构建过程的输出是应用程序、库还是插件。

可以通过在变量名称前面加上前缀来读取变量的内容。这可用于将一个变量的内容分配给另一个变量：$$

```
TEMP_SOURCES = $$SOURCES
```

运算符广泛用于对字符串和值列表进行操作的内置函数。有关详细信息，请参阅qmake 语言。$$

空白

通常，空格分隔变量赋值中的值。若要指定包含空格的值，必须将值括在双引号中：

```
DEST = "Program Files"
```

引用的文本被视为变量保存的值列表中的单个项目。类似的方法用于处理包含空格的路径，特别是在为 Windows 平台定义INCLUDEPATH和LIBS变量时：

```
win32:INCLUDEPATH += "C:/mylibs/extra headers"
unix:INCLUDEPATH += "/home/user/extra headers"
```

评论

您可以向项目文件添加注释。注释以字符开头，并继续到同一行的末尾。例如：#

```
# Comments usually start at the beginning of a line, but they
# can also follow other content on the same line
```

要在变量赋值中包含字符，必须使用内置LITERAL_HASH变量的内容。#

内置函数和控制流

QMAKE提供了许多内置函数来处理变量的内容。简单项目文件中最常用的函数是include（） 函数，它将文件名作为参数。给定文件的内容包含在项目文件中使用函数的位置。该函数最常用于包含其他项目文件：includeinclude

```
include(other.pro)
```

对条件结构的支持是通过在编程语言中表现类似语句的作用域提供的：if

```
win32 {
    SOURCES += paintwidget_win.cpp
}
```

仅当条件为真时，才会进行大括号内的赋值。在这种情况下，必须设置CONFIG选项。这在Windows上自动发生。开口支架必须与条件站在同一条线上。win32

对通常需要循环的变量进行更复杂的操作由内置函数（如find（），unique（）和count（））提供。提供这些函数以及许多其他函数是为了操作字符串和路径、支持用户输入和调用外部工具。有关使用这些函数的详细信息，请参阅qmake 语言。有关所有函数及其说明的列表，请参阅替换函数和测试函数。

项目模板

变量用于定义将生成的项目类型。如果未在项目文件中声明，qmake 将假定应构建应用程序，并将为此目的生成适当的 Makefile（或等效文件）。

下表总结了可用的项目类型，并描述了 qmake 将为每个项目生成的文件：

模板	qmake 输出
应用（默认）	生成文件以生成应用程序。
自由	生成文件以构建库。
辅助	生成文件以不构建任何内容。如果不需要调用编译器来创建目标，例如，因为项目是用解释型语言编写的，请使用此选项。 **注意：**此模板类型仅适用于基于生成文件的生成器。特别是，它不适用于vcxproj和Xcode生成器。
子目录	包含使用SUBDIRS变量指定的子目录的规则的生成文件。每个子目录必须包含自己的项目文件。
VC普	用于生成应用程序的 Visual Studio Project 文件。

当使用模板时，qmake 会生成一个 Makefile 来检查每个指定的子目录，处理它在那里找到的任何项目文件，并在新创建的 Makefile 上运行平台的凳子。变量用于包含要处理的所有子目录的列表。subdirsmakeSUBDIRS

常规配置

CONFIG变量指定应配置项目的选项和功能。

项目可以在*发布*模式和/或*调试*模式下生成。如果同时指定了调试和发布，则最后一个生效。如果指定用于生成项目的调试版本和发布版本的选项，则 qmake 生成的生成文件将包含构建两个版本的规则。可以通过以下方式调用它：debug_and_release

```
make all
```

将选项添加到变量会使此规则成为生成项目时的默认规则。build_allCONFIG

> **注意：**变量中指定的每个选项也可以用作范围条件。您可以使用内置的CONFIG () 函数测试是否存在某些配置选项。例如，以下行将函数显示为范围中的条件，以测试是否仅使用该选项：CONFIGopengl

```
CONFIG(opengl) {
    message(Building with OpenGL support.)
} else {
    message(OpenGL support is not available.)
}
```

这样就可以为构建定义不同的配置。有关详细信息，请参阅使用作用域。releasedebug

以下选项定义要生成的项目类型。

> **注意：**其中一些选项仅在相关平台上使用时生效。

选择	描述
.qt	该项目是一个Qt应用程序，应该链接到Qt库。您可以使用变量来控制应用程序所需的任何其他Qt模块。默认情况下会添加此值，但您可以将其删除以将 qmake 用于非 Qt 项目。QT
x11	该项目是一个 X11 应用程序或库。如果目标使用 Qt，则不需要此值。

应用程序和库项目模板为您提供了更专业的配置选项，以微调生成过程。生成常见项目类型中详细介绍了这些选项。

例如，如果您的应用程序使用 Qt 库，并且您希望在模式下构建它，则您的项目文件将包含以下行：debug

```
CONFIG += qt debug
```

注意：您必须使用"+="，而不是"="，否则qmake将无法使用Qt的配置来确定项目所需的设置。

声明 Qt 库

如果CONFIG变量包含该值，则启用 qmake 对 Qt 应用程序的支持。这样就可以微调应用程序使用的Qt模块。这是通过QT变量实现的，该变量可用于声明所需的扩展模块。例如，我们可以通过以下方式启用 XML 和网络模块：qt

```
QT += network xml
```

注意：默认包含 and模块，因此上述声明将网络和 XML 模块*添加到*此默认列表中。以下赋值*省略了*默认模块，并且在编译应用程序的源代码时会导致错误：QTcoregui

```
QT = network xml # This will omit the core and gui modules.
```

如果要构建*一个没有模块的项目*，则需要使用"-="运算符将其排除。默认情况下，包含 bothand，因此以下行将导致构建一个最小的 Qt 项目：guiQTcoregui

```
QT -= gui # Only the core module is used.
```

有关可以添加到变量的 Qt 模块列表，请参阅QT。QT

配置功能

可以使用功能（.prf）文件中指定的额外配置功能来设置QMAKE。这些额外功能通常为生成过程中使用的自定义工具提供支持。若要将功能添加到生成过程，请将功能名称（功能文件名的词干）追加到变量中。CONFIG

例如，qmake 可以通过以下行配置构建过程以利用pkg 配置支持的外部库，例如 D-Bus 和 ogg 库：

```
CONFIG += link_pkgconfig
PKGCONFIG += ogg dbus-1
```

有关添加功能的详细信息，请参阅添加新配置功能。

如果您在项目中使用Qt提供的库之外的其他库，则需要在项目文件中指定它们。

qmake 搜索库的路径和要链接的特定库可以添加到LIBS变量中的值列表中。您可以指定库的路径，也可以使用 Unix 样式的表示法来指定库和路径。

例如，以下行显示如何指定库：

```
LIBS += -L/usr/local/lib -lmath
```

也可以使用INCLUDEPATH变量以类似的方式指定包含头文件的路径。

例如，要添加多个要搜索头文件的路径：

```
INCLUDEPATH = c:/msdev/include d:/stl/include
```

‹ 开始使用 qmake

构建通用项目类型 ›

社区

为Qt做贡献
论坛
维基
下载
市场

反馈 登录