

平台说明

许多跨平台项目可以通过基本的 qmake 配置功能来处理。但是，在某些平台上，利用特定于平台的功能有时很有用，甚至是必要的。Qmake知道其中的许多功能，这些功能

macOS、iOS、tvOS 和 watchOS

特定于这些平台的功能包括支持创建通用二进制文件、框架和捆绑包。

源包和二进制包

源包中提供的 qmake 版本与二进制包中提供的版本略有不同，因为它使用不同的功能规范。源包通常使用规范，二进制包通常配置为使用规范。macx-g++macx-xcode 每个包的用户都可以通过使用选项调用 qmake 来覆盖此配置（有关详细信息，请参阅[运行 qmake](#)）。例如，要使用二进制包中的 qmake 在项目目录中创建生成文件，请调用

```
qmake -spec macx-g++
```

使用框架

qmake能够自动生成构建规则，用于链接到macOS上标准框架目录中的框架，位于。/Library/Frameworks/

```
LIBS += -F/path/to/framework/directory/
```

通过将选项和框架名称附加到LIBS变量来链接框架本身：-framework

```
LIBS += -framework TheFramework
```

创建框架

可以配置任何给定的库项目，以便将生成的库文件放置在框架中，准备进行部署。为此，请将项目设置为使用lib模板，并将选项添加到CONFIG变量：lib_bundle

```
TEMPLATE = lib
CONFIG += lib_bundle
```

与库关联的数据是使用QMAKE_BUNDLE_DATA变量指定的。它包含将与库捆绑包一起安装的项目，通常用于指定头文件的集合，如以下示例所示：

```
FRAMEWORK_HEADERS.version = Versions
FRAMEWORK_HEADERS.files = path/to/header_one.h path/to/header_two.h
FRAMEWORK_HEADERS.path = Headers
QMAKE_BUNDLE_DATA += FRAMEWORK_HEADERS
```

您可以使用变量来指定特定框架所需的标头。将其附加到变量可确保有关这些标头的信息添加到将与库捆绑包一起安装的资源集合中。此外，框架名称和版本由QMAKE_FRAMEWORK_HEADERSQMAKE_BUNDLE_DATA

有关部署应用程序和库的更多信息，请参阅[Qt for macOS - 部署](#)。

创建和移动 Xcode 项目

在 macOS 上，开发人员可以从现有的 qmake 项目文件生成 Xcode 项目文件。例如：

```
qmake -spec macx-xcode project.pro
```

注意：如果稍后在磁盘上移动了项目，则必须再次运行 qmake 来处理项目文件并创建新的 Xcode 项目文件。

同时支持两个构建目标

实现这一点目前是不可行的，因为活动构建配置的 Xcode 概念在概念上与构建目标的 qmake 思想不同。

Xcode 活动构建配置设置用于修改 Xcode 配置、编译器标志和类似的构建选项。与 Visual Studio 不同，Xcode 不允许根据选择调试还是发布构建配置来选择特定的库文件。qmake 目前无法从 qmake 生成的 Xcode 项目文件中在 Xcode 配置设置中设置文件。库在 Xcode 构建系统的 *框架和库* 阶段链接的方式。

此外，选定的 *活动构建配置* 存储在 .pbxuser 文件中，该文件由 Xcode 在首次加载时生成，而不是由 qmake 创建。

窗口

特定于此平台的功能包括支持 Windows 资源文件（提供或自动生成），创建 Visual Studio 项目文件，以及在部署使用 Visual Studio 2005 或更高版本开发的 Qt 应用程序时处理清

添加窗口资源文件

本节介绍如何使用 qmake 处理 Windows 资源文件，使其链接到应用程序可执行文件（EXE）或动态链接库（DLL）。qmake 可以选择自动生成适当填充的 Windows 资源文件。链接的 Windows 资源文件可能包含许多可由其 EXE 或 DLL 访问的元素。但是，Qt 资源系统应该用于以独立于平台的方式访问链接资源。但是链接的 Windows 资源文件的某些对于 Qt 创建的 Windows EXE 或 DLL 来说，同时使用这两种技术是一种很好的做法：通过 Qt 资源系统链接独立于平台的资源，并通过 Windows 资源文件添加 Windows 特定的资源。通常，资源定义脚本（.rc 文件）编译为 Windows 资源文件。在 Microsoft 工具链中，RC 工具生成一个 .res 文件，该文件可以与 Microsoft 链接器一起链接到 EXE 或 DLL。Min qmake 通过设置至少一个系统变量 VERSION 和 RC_ICONS 来触发 qmake 适当填充的 .rc 文件的可选自动生成。生成的 .rc 文件会自动编译和链接。添加到 .rc 文件的元素由系统 QMAKE_TARGET_ORIGINAL_FILENAME, QMAKE_TARGET_INTERNALNAME, QMAKE_TARGET_COMMENTS, QMAKE_TARGET_TRADEMARKS, QMAKE_MANIFEST, RC_CO 如果这些元素还不够，qmake 将两个系统变量 RC_FILE 和 RES_FILE 直接指向外部创建的 .rc 或 .res 文件。通过设置这些变量之一，指定的文件将链接到 EXE 或 DLL。

注意：如果设置了 RC_FILE 或 RES_FILE，则阻止 qmake 生成 .rc 文件。在这种情况下，qmake 不会对给定的 .rc 文件或 .res 或 .o 文件进行进一步的更改，与 .rc 文件生成相关的变

创建 Visual Studio 项目文件

本节介绍如何将现有的 qmake 项目导入 Visual Studio。qmake 能够获取项目文件并创建一个 Visual Studio 项目，其中包含开发环境所需的所有必要信息。这是通过将 qmake 项这也可以使用命令行选项进行设置，例如：

```
qmake -tp vc
```

可以通过键入以下内容递归地生成子目录中的文件和主目录中的文件：.vcproj.sln

```
qmake -tp vc -r
```

每次更新项目文件时，都需要运行 qmake 来生成更新的 Visual Studio 项目。

注意：如果您使用的是 Visual Studio 加载项，请选择 Qt>Import from .pro 文件以导入文件。 .pro

视觉工作室清单文件

部署使用 Visual Studio 2005 或更高版本构建的 Qt 应用程序时，请确保正确处理链接应用程序时创建的清单文件。对于生成 DLL 的项目，这是自动处理的。

删除应用程序可执行文件的清单嵌入可以通过对 CONFIG 变量的以下赋值来完成：

```
CONFIG -= embed_manifest_exe
```

此外，可以通过对变量的以下赋值来删除 DLL 的清单嵌入：CONFIG

```
CONFIG -= embed_manifest_dll
```

Windows 部署指南中对此进行了更详细的讨论。

< 运行 qmake

Qt

The Qt Company

f

in

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

© 2022 Qt公司

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

反馈

登录