

🔍 搜索

Qt 6.4 > 使用 CMake 构建 > [CMake 命令参考](#)

# CMake 命令参考

## Qt6: : 核心

|   |                                |
|---|--------------------------------|
| qt_add_big_resources                    | 将大型二进制资源编译为目标代码                |
| qt_add_binary_resources                 | 从 Qt 资源文件列表创建 RCC 文件           |
| qt_add_executable                       | 创建并最终确定特定于平台类型的应用程序目标          |
| qt_add_library                          | 创建并完成库                         |
| qt_add_plugin                           | 创建 Qt 插件目标                     |
| Topics >                                |                                |
| qt_android_add_apk_target               | 定义运行安卓部署qt以生成APK的构建目标          |
| qt_android_apply_arch_suffix            | 将目标二进制文件的名称配置为包含特定于体系结构的后缀     |
| qt_android_generate_deployment_settings | 生成 androiddeployqt 所需的部署设置文件   |
| qt_deploy_qt_conf                       | 在部署时编写 qt.conf 文件              |
| qt_deploy_runtime_dependencies          | 部署可执行文件所需的Qt插件， Qt和非Qt库        |
| qt_disable_unicode_defines              | 防止在目标上自动设置某些与 Unicode 相关的编译器定义 |
| qt_extract_metatypes                    | 从 Qt 目标中提取元类型并生成关联的元类型.json 文件 |
| qt_finalize_project                     | 处理与Qt项目相关的各种常见平台特定任务           |
| qt_finalize_target                      | 处理与Qt目标相关的各种常见平台特定任务           |
| qt_generate_deploy_app_script           | 为应用程序生成部署脚本                    |
| qt_generate_moc                         | 对输入文件调用 moc                    |
| qt_import_plugins                       | 指定要为静态Qt构建导入的自定义插件集            |
| qt_set_finalizer_mode                   | 自定义目标最终确定的各个方面                 |
| qt_standard_project_setup               | 将项目范围的默认设置设置为标准排列              |
| qt_wrap_cpp                             | 从源创建 .moc 文件                   |

|                            |                          |
|----------------------------|--------------------------|
| qt_add_dbus_adaptor        | 为 D-Bus 接口生成适配器类         |
| qt_add_dbus_interface      | 生成实现 D-Bus 接口描述文件接口的C++源 |
| qt_add_dbus_interfaces     | 生成实现 D-Bus 接口描述文件接口的C++源 |
| qt_generate_dbus_interface | 从头文件生成 D-Bus 接口          |

Qt6：： 接口框架

|                            |                          |
|----------------------------|--------------------------|
| qt_add_dbus_adaptor        | 为 D-Bus 接口生成适配器类         |
| qt_add_dbus_interface      | 生成实现 D-Bus 接口描述文件接口的C++源 |
| qt_add_dbus_interfaces     | 生成实现 D-Bus 接口描述文件接口的C++源 |
| qt_generate_dbus_interface | 从头文件生成 D-Bus 接口          |

Qt6：： 语言学家工具

|                       |                                     |
|-----------------------|-------------------------------------|
| qt_add_lrelease       | 添加目标以将Qt语言学家.ts文件转换为.qm文件           |
| qt_add_lupdate        | 添加目标以生成或更新Qt语言学家.ts文件               |
| qt_add_translation    | 将Qt Linguist .ts文件编译为.qm文件          |
| qt_add_translations   | 添加目标以更新Qt Linguist .ts文件并将其转换为.qm文件 |
| qt_create_translation | 设置Qt语言学家翻译工具链                       |

Qt6：： Qml

|                                   |                           |
|-----------------------------------|---------------------------|
| qt_add_qml_module                 | 定义 QML 模块                 |
| qt_add_qml_plugin                 | 定义与 QML 模块关联的插件           |
| qt_deploy_qml_imports             | 部署可执行文件所需的 QML 模块的运行时组件   |
| qt_generate_deploy_qml_app_script | 为 QML 应用程序生成部署脚本          |
| qt_generate_foreign_qml_types     | 在 QML 模块中注册来自一个目标的类型      |
| qt_import_qml_plugins             | 确保为静态构建导入目标所需的 QML 插件     |
| qt_query_qml_module               | 检索有关 QML 模块的信息            |
| qt_target_compile_qml_to_cpp      |                           |
| qt_target_qml_sources             | 将 qml 文件和资源添加到现有 QML 模块目标 |

Qt6：： 远程对象

|                      |                                  |
|----------------------|----------------------------------|
| qt_add_repc_merged   | 从 Qt 远程对象 .rep 文件为源和副本类型创建C++头文件 |
| qt_add_repc_replicas | 从 Qt 远程对象 .rep 文件为副本类型创建C++头文件   |
| qt_add_repc_sources  | 从 Qt 远程对象 .rep 文件为源类型创建C++头文件    |

Qt6: : Scxml

|                    |  |
|--------------------|--|
| qt_add_statecharts |  |
|--------------------|--|

Qt6: : 小部件

|            |             |
|------------|-------------|
| qt_wrap_ui | 为 .ui 文件创建源 |
|------------|-------------|

Qt6: : WebEngineCore

|                             |                                |
|-----------------------------|--------------------------------|
| qt_add_webengine_dictionary | 将 hunspell 字典格式转换为 bdict 二进制格式 |
|-----------------------------|--------------------------------|

< Qt 5 和 Qt 6 兼容性
 CMake 变量引用 >

©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

发牌

- 条款和条件
- 开源
- 常见问题

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

为Qt做贡献  
论坛  
维基  
下载  
市场

© 2022 Qt公司

[反馈](#) [登录](#)