Search

Qt 6.4 > Build with CMake > CMake Variable Reference

CMake Variable Reference

Module variables

Qt modules loaded with set various variables.`find_package`

> **Note:** You rarely need to access these variables directly. Common tasks like linking against a module should be done through the library targets each module defines.

For example, , when successful, makes the following variables available:`find_package(Qt6 COMPONENTS Widgets)`

Variable	Description
`Qt6Widgets_COMPILE_DEFINITIONS`	A list of compile definitions to use when building against the library.
`Qt6Widgets_DEFINITIONS`	A list of definitions to use when building against the library.
`Qt6Widgets_EXECUTABLE_COMPILE_FLAGS`	A string of flags to use when building executables against the library.
`Qt6Widgets_FOUND`	A boolean that describes whether the module was found successfully.
`Qt6Widgets_INCLUDE_DIRS`	A list of include directories to use when building against the library.
`Qt6Widgets_LIBRARIES`	The name of the imported target for the module: `Qt5::Widgets`
`Qt6Widgets_PRIVATE_INCLUDE_DIRS`	A list of private include directories to use when building against the library and using private Qt API.
`Qt6Widgets_VERSION_STRING`	A string containing the module's version.

For all packages found with , equivalents of these variables are available; they are case-sensitive.`find_package`

Installation variables

Variable	Description
QT_DEFAULT_MAJOR_VERSION	An integer that controls the Qt version that commands forward to in case of mixed Qt 5 and Qt 6 projects. It needs to be set to either or before the respective calls.qt_56find_package() If set to , commands starting with will call their counterpart starting with . If set to , they will call their counterpart starting with .5qt_qt5_6qt6_ If not set, the first call defines the default version.find_package
QT_LIBINFIX	A string that holds the infix used in library names, when Qt is configured with .-libinfix
QT_NO_CREATE_VERSIONLESS_FUNCTIONS	Hides commands that start with , leaving only the versioned ones starting with .qt_qt6_
QT_NO_CREATE_VERSIONLESS_TARGETS	Hides the imported targets starting with . Instead, you need to use the targets starting with .Qt::Qt6::
QT_VISIBILITY_AVAILABLE	On Unix, a boolean that describes whether Qt libraries and plugins were compiled with . This means that only selected symbols are exported.-fvisibility=hidden

Project variables

These variables can influence CMake commands provided by Qt. They may be set by the project, a toolchain file or other third-party packages.

Topics >

ANDROID_NDK_HOST_SYSTEM_NAME	特定于安卓系统的主机系统架构
ANDROID_SDK_ROOT	安卓开发工具包的位置
QT_ANDROID_ABIS	构建项目文件包所依据的 ABI 列表
QT_ANDROID_APPLICATION_ARGUMENTS	要传递给安卓应用程序的参数列表
QT_ANDROID_BUILD_ALL_ABIS	支持使用自动检测的适用于安卓的 Qt SDK 列表构建多 ABI 包
QT_ANDROID_SIGN_AAB	使用指定的密钥库、别名和存储密码对 .aab 包进行签名
QT_ANDROID_SIGN_APK	使用指定的密钥库、别名和存储密码对软件包进行签名
QT_DEPLOY_BIN_DIR	用于在某些目标平台上部署运行时二进制文件的前缀相对子目录
QT_DEPLOY_LIB_DIR	用于在某些目标平台上部署库的前缀相对子目录
QT_DEPLOY_PLUGINS_DIR	前缀相对子目录，用于在某些目标平台上部署 Qt 插件
QT_DEPLOY_PREFIX	部署的基本位置
QT_DEPLOY_QML_DIR	前缀相对子目录，用于在某些目标平台上部署 QML 插件
QT_DEPLOY_SUPPORT	要包含的用于设置部署支持的文件的名称

QT_IOS_LAUNCH_SCREEN	所有目标使用的 iOS 启动屏幕情节提要的路径
QT_NO_COLLECT_BUILD_TREE_APK_DEPS	防止在 Android 部署期间收集项目构建的共享库目标
QT_NO_SET_XCODE_BUNDLE_IDENTIFIER	禁止在 iOS 上完成目标期间提供回退应用捆绑 ID
QT_NO_SET_XCODE_DEVELOPMENT_TEAM_ID	禁止在 iOS 上完成目标期间提供回退团队 ID
QT_NO_STANDARD_PROJECT_SETUP	防止对 qt_standard_project_setup（）的后续调用进行任何更改
QT_PATH_ANDROID_ABI_<阿比>	用于为相应的 ABI 指定安卓版 Qt 路径的变量集

Qt6：

QT_QML_OUTPUT_DIRECTORY	基本输出目录，默认情况下将在其下创建 QML 模块

Qt6：：接口框架

‹ 命令参考

断续器属性参考 ›

客户成功案例

社区

为Qt做贡献

论坛

维基

下载

市场

反馈 **登录**