Qt 5 and Qt 6 compatibility

The semantics of the CMake API in Qt 5 and Qt 6 are largely compatible. However, up to Qt 5.14, all imported Qt library targets and commands contained the version number as part of the name. This makes writing CMake code that should work with both Qt 5 and Qt 6 somewhat cumbersome. Qt 5.15 therefore introduced *versionless* targets and commands to enable writing CMake code that is largely agnostic to the different Qt versions.

Versionless targets

One can both reference `Qt6::Core`, or `Qt::Core`:

```
find_package(Qt6 COMPONENTS Core)
if (NOT Qt6_FOUND)
    find_package(Qt5 5.15 REQUIRED COMPONENTS Core)
endif()

add_executable(helloworld
    ...
)

target_link_libraries(helloworld PRIVATE Qt::Core)
```

Above snippet first tries to find a Qt 6 installation. If that fails, it tries to find a Qt 5.15 package. Independent of whether Qt 6 or Qt 5 is used, we can use the imported `Qt::Core` target.

The versionless targets are defined by default. Set QT_NO_CREATE_VERSIONLESS_TARGETS before the first `find_package()` call to disable them.

> **Note:** The imported Qt::Core target will not feature the target properties that are available in the Qt6::Core target.

Versionless commands

Since Qt 5.15, the Qt modules also provide versionless variants of their commands. You can for instance now use qt_add_translation to compile translation files, independent of whether you use Qt 5 or Qt 6.

Mixing Qt 5 and Qt 6

There might be projects that need to load both Qt 5 and Qt 6 in one CMake context (though mixing Qt versions in one library or executable is not supported, so be careful there).

In such a setup the versionless targets and commands will be implicitly referring to the first Qt version that was found via `find_package`. Set the QT_DEFAULT_MAJOR_VERSION CMake variable before the first `find_package` call to make the version explicit.

Supporting older Qt 5 versions

If you need to support also Qt 5 versions older than Qt 5.15, you can do so by storing the current version in an CMake variable:

```
find_package(QT NAMES Qt6 Qt5 REQUIRED COMPONENTS Core)
find_package(Qt${QT_VERSION_MAJOR} REQUIRED COMPONENTS Core)

add_executable(helloworld
    ...
)

target_link_libraries(helloworld PRIVATE Qt${QT_VERSION_MAJOR}::Core)
```

Here we let `find_package(<PackageName>...)` try to find first Qt 6, and if that fails Qt 5, under the name QT. If either of them is found, `find_package` will succeed, and the CMake variable QT_VERSION_MAJOR will be defined to either 5 or 6.

We then do load the package for the determined Qt version again by creating the name `Qt${QT_VERSION_MAJOR}` on the fly. This is needed because CMAKE_AUTOMOC expects the package name to be either Qt5 or Qt6, and will print an error otherwise.

We can use the same pattern to also specify the name of the imported library. Before calling `target_link_libraries`, CMake will resolve `Qt${QT_VERSION_MAJOR}::Widgets` to either `Qt5::Widgets` or `Qt6::Widgets`.

Recommended practices

Use the versionless variants of the CMake commands where possible.

Versionless imported targets are mostly useful for projects that need to compile with both Qt 5 and Qt 6. Because of the missing target properties, we do not recommend using them by default.

Use the versioned versions of the CMake commands and targets if you need to support Qt 5 versions older than Qt 5.15, or if you cannot control whether your CMake code is loaded in a context where QT_NO_CREATE_VERSIONLESS_FUNCTIONS or QT_NO_CREATE_VERSIONLESS_TARGETS might be defined. In this case you can still simplify your code by determining the actual command or target name through a variable.

Unicode support in Windows

In Qt 6, the UNICODE and _UNICODE compiler definitions are set by default for targets that link against Qt modules. This is in line with the qmake behavior, but it is a change compared to the CMake API behavior in Qt 5.

```
find_package(Qt6 COMPONENTS Core)

add_executable(helloworld
    ...
)

qt_disable_unicode_defines(helloworld)
```

‹ Imported targets

CMake Command Reference ›

Marketplace

Feedback Sign In