

Qt 6.4 > 使用“聚合”进行构建 > [构建可重用的 QML 模块](#)

构建可重用的 QML 模块

下面的示例演示如何创建一个向 QML 公开 C++ 库。该示例的目录结构如下所示：

```
├── CMakeLists.txt
├── example
│   └── mylib
│       ├── CMakeLists.txt
│       ├── mytype.cpp
│       └── mytype.h
```

toplevel 文件执行一些基本设置，然后用于将该文件包含在 mylib 中。子目录结构对应于 QML 模块的 URI，但点被斜杠替换。这与引擎在[导入路径](#)中搜索模块时使用的逻辑相同。声明一个类，并使用声明性注册宏将其公开给引擎。CMakeLists.txt 添加子目录 mytype.h

在子目录中，我们再次调用。但是，调用略有不同：CMakeLists.txt 调用 qt6_add_qml_module

```
qt6_add_qml_module(mylib
    URI example/mylib
    VERSION 1.0
    SOURCES
        mytype.h mytype.cpp
)
```

若要添加 C++ 类型，需要指定 SOURCES 参数。未创建 mylib 的目标。因此，如果传递给的目标不存在，则会自动创建一个库目标，在这种情况下需要该目标。qt6_add_qml_module

构建项目时，除了库之外，还会构建一个 QML 插件。插件的自动生成的类从 `QmlEngineExtensionPlugin` 扩展自 `QmlEngineExtensionPlugin`。mylib 库本身已包含用于向引擎注册类型的代码。但是，这仅在我们可以链接到库的情况下才有用。要使模块在由加载的 QML 文件中可用，[QML 运行时工具](#)需要一个可以加载的插件。然后，该插件负责实际链接到库，并确保类型被注册。

请注意，只有当模块除了注册类型之外不执行任何操作时，才可以自动生成插件。如果它需要执行更高级的功能，例如在 `QmlEngineExtensionPlugin` 中注册图像提供程序，您仍然需要手动编写插件。[qt6_add_qml_module](#) 对此有支持。`initializeEngineNO_GENERATE_PLUGIN_SOURCE`

此外，遵循目录布局约定有助于工具。该布局将镜像到生成目录中。这意味着您可以将构建目录的路径传递给 QML 工具（通过 `-I` 标志），它将找到插件。

在构建之前，将 QML 文件添加到模块中。在 lib 子文件夹中，添加一个文件 `MyType.qml`



```
import example.mylib

MyType{
    answer: 43
}
```

并调整呼叫: `qt6_add_qml_module`

```
qt6_add_qml_module(mylib
    URI example.mylib
    VERSION 1.0
    SOURCES
        mytype.h mytype.cpp
    QML_FILES
        Mistake.qml
)
```

如前所述, 我们犯了一个错误, 因为实际上是一个只读属性。这说明了集成: CMake 创建了一个目标, 一旦我们运行它, 就会发出有关该问题的警告: `answerqmlintqmlintqmlint`

```
$> cmake --build . --target mylib_qmlint
...
Warning: Mistake.qml:4:13: Cannot assign to read-only property answer
    answer: 43
           ^^
```

[◀ 构建 QML 应用程序](#)

[在命令行上生成项目 ▶](#)

©2022 Qt Ltd. 此处包含的文档贡献是其各自所有者的版权。此处提供的文档是根据自由软件基金会发布的 [GNU 自由文档许可证 1.3 版](#) 的条款进行许可的。Qt 及其相应的徽标是 Qt 有限公司在芬兰和/或全球其他国家的 [商标](#)。所有其他商标均为其各自所有者的财产。



Contact Us

Company

About Us

Licensing

Terms & Conditions



Careers
Office Locations

Support

Support Services
Professional Services
Partners
Training

For Customers

Support Center
Downloads
Qt Login
Contact Us
Customer Success

Community

Contribute to Qt
Forum
Wiki
Downloads
Marketplace

© 2022 The Qt Company

[Feedback](#) [Sign In](#)