

[Qt设计工作室手册](#) > [将 QML 模块与插件结合使用](#)

将 QML 模块与插件结合使用

QML 模块可以使用C++**插件**将C++中定义的组件公开给 QML 应用程序。

要创建 QML 模块并使其显示在“**组件**”视图中，请执行以下操作：

1. 创建自定义组件并将所有文件放在专用于模块的目录中。例如：`.qmlimports\asset_imports`
2. 对于Qt Quick UI项目（.qmlproject），请在应用程序的qmlproject文件中指定包含模块的目录的路径，其中要将模块用作变量的值。例如。`importPathsimportPaths: ["imports", "asset_imports"]`
3. 为模块创建一个文件并将其放在模块目录中。有关详细信息，请参阅**模块定义 qmldir 文件**。`qmldir`
4. 按照**生成类型描述文件**中的说明创建文件。`qmltypes`

Qt设计工作室手册3.8.0
Topics >

件为例。`.metainfodesignerqtquickcontrols2.metainfo`

7. 使用与Qt Design Studio相同的Qt版本和编译器构建模块。有关更多信息，请参阅**在设计模式下运行 QML 模块**。

您的模块现在应显示在“**组件**”中。如果存在有效文件，则组件应显示在**组件**的子部分中。`.metainfo`

生成类型描述文件

注册 QML 类型时，请确保 QML 模块具有文件。理想情况下，它应该与文件位于同一目录中。该文件包含由模块插件导出的组件的描述，并在导入模块时由Qt Design Studio加载。`plugins.qmltypesqmldirqmltypes`

有关详细信息，请参阅**类型说明文件**。

自动转储插件

如果带有插件的模块缺少文件，Qt Design Studio会尝试通过在后台运行程序来生成临时文件。但是，这种自动转储是一种具有许多故障点的回退机制，您不能依赖它。`qmltypesqmlldump`

导入 QML 模块

默认情况下，Qt Design Studio 将查找 Qt for QML 模块的 QML 导入路径。

如果您使用 qmake 并且您的应用程序添加了 Qt Design Studio 应该使用的其他导入路径，请在应用程序的文件中指定它们：`QML_IMPORT_PATH.proQML_IMPORT_PATH += path/to/module`

如果使用 CMake，请将以下命令添加到 CMakeLists.txt 文件中以设置 QML 导入路径：

社区

- 为Qt做贡献
- 论坛
- 维基
- 下载
- 市场

© 2022 Qt公司

[反馈](#) [登录](#)