

🔍 搜索

Qt创作者手册 > [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，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`
5. 在模块目录中创建一个名为的目录。`designer`
6. 为您的模块创建一个文件并将其放在目录中。使用与Qt一起提供的元信息文件为例。`.metainfodesignerqtquickcontrols2.metainfo`
7. 按照**导入 QML** 模块中的说明将模块导入到项目中。

## 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 Creator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 Creator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。`qmltypesqmlDump`

## 导入 QML 模块

默认情况下，Qt Creator 将在 Qt 的 QML 导入路径中查找 QML 模块。

如果您使用 qmake 并且您的应用程序添加了 Qt Creator 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：

```
set(QML_IMPORT_PATH ${CMAKE_SOURCE_DIR}/qml ${CMAKE_BINARY_DIR}/imports CACHE STRING "" FORCE)
```

导入路径会影响 CMake 项目生成的所有目标。

[◀ 用户界面文件](#)

[开发基于小部件的应用程序 ▶](#)

©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/或全球其他国家的商标。所有其他商标均为财产 其各自所有者。



The Qt  
Company



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

社区

为Qt做贡献

论坛

维基

下载

市场

© 2022 Qt公司

反馈 登录

https://doc.qt.io/qtcreator/creator-qml-modules-with-plugins.html

3/3