搜索

将 UI 项目转换为应用程序

Qt快速UI项目对于创建用户界面很有用。要在Qt Creator中使用它们进行应用程序开发，您必须添加：

- 项目配置文件（CMakeLists.txt 或 .pro）
- C++代码 （.cpp）
- 资源文件
- 将应用程序部署到设备所需的代码

有关集成 QML 和C++的更多信息，请参阅概述 - QML 和C++集成。

注意： 从Qt Design Studio 2.3.0开始，Qt Design Studio项目向导模板会生成可以使用CMake构建的项目。您可以在Qt Creator中打开*CMakeLists.txt*项目文件以继续开发项目。

有关使用Qt Design Studio创建项目的更多信息，请参阅Qt Design Studio手册。

如果要使用 qmake 作为构建系统，可以使用 Qt Creator 向导模板创建使用 qmake 构建系统构建的 Qt Quick 应用程序，然后将源文件从 Qt UI 快速项目复制到应用程序项目。

您可以使用项目配置文件中的选项自动将所有 QML 文件和相关资产添加到Qt 资源集合文件 （.qrc） 中。但是，大文件应作为外部二进制资源包含在内，而不是将它们编译到二进制文件中。`RESOURCES`

向导会自动将选项添加到项目文件中，以指定所需的QML 导入路径。仅当多个子目录包含 QML 文件时，才需要该路径。`QML_IMPORT_PATH`

然后，您可以使用主C++源文件中的QQuickView类在应用程序启动时显示主 QML 文件。

*Qt Quick Designer Components*模块是在安装 Qt Design Studio 时安装的。如果要在Qt Creator中编辑的项目中使用模块中的Qt Quick Studio组件或效果，则必须构建模块并将其安装到Qt中才能构建项目。有关更多信息，请参阅将Qt快速设计器组件添加到Qt安装。

Qt Quick Timeline模块是在安装Qt Design Studio时安装的。如果您只安装Qt Creator和Qt，请记住还要选择Qt Quick Timeline模块进行安装。如果您的Qt版本低于5.14，则必须构建Qt快速时间轴模块并将其安装到Qt中才能构建项目。有关更多信息，请参阅将Qt快速时间轴模块添加到Qt安装。

Qt创建者手册8.0.2
Topics >

要将具有 .qmlproject 文件的项目转换为具有 .pro 文件的项目，请执行以下操作：

1. 选择**文件**>**新建项目**>**应用程序 （Qt）**>**Qt 快速应用程序**>**选择**。
2. 在“生成系统”字段中，选择“qmake”作为用于生成和运行项目的**生成系统**，然后选择“**下一步**”（或在 macOS 上**选择**“**继续**”）。
3. 按照向导的说明创建项目。
4. 在文件资源管理器中，将源文件从Qt Quick UI项目目录复制到应用程序项目目录中的子目录。出于这些说明的目的，调用该目录。`qml`
5. 打开应用程序项目文件，并编辑选项的值以添加以下行：`RESOURCES`

6. 同时编辑选项的值以指定 QML 导入路径：QML_IMPORT_PATH

```
QML_IMPORT_PATH = qml/imports
```

导入路径在哪里。qml/imports

7. 选择"**生成>运行 qmake**"，将该选项应用于生成配置。RESOURCES
8. 打开*主.cpp*文件，将QQmlApplicationEngine对象替换为QQuickView对象：

```
QQuickView view;
view.engine()->addImportPath("qrc:/qml/imports");
view.setSource(QUrl("qrc:/qml/ProgressBar.ui.qml"));
if (!view.errors().isEmpty())
    return -1;
view.show();
```

导入路径在哪里，是Qt Quick UI项目中主QML文件的路径和名称。qrc:/qml/importsqrc:/qml/ProgressBar.ui.qml

9. 选择"**生成>运行**"以生成并运行项目。

例如，如果您将*进度条*示例的源文件从Qt Design Studio安装（位于目录中）复制到空的Qt Quick应用程序项目并进行必要的更改，则*主.cpp*文件应如下所示：\share\qtcreator\examples\ProgressBar

```
#include <QGuiApplication>
#include <QQmlApplicationEngine>
#include <QQuickView>

int main(int argc, char *argv[])
{
    QCoreApplication::setAttribute(Qt::AA_EnableHighDpiScaling);

    QGuiApplication app(argc, argv);

    QQuickView view;
    view.engine()->addImportPath("qrc:/qml/imports");
    view.setSource(QUrl("qrc:/qml/ProgressBar.ui.qml"));
    if (!view.errors().isEmpty())
        return -1;
    view.show();

    app.exec();
}
```

处理大型数据文件

UI 中使用的图形资产（如图像、效果或 3D 场景）是 UI 中性能问题的典型原因。如果您尝试将大型资产文件（如 100 MB 3D 模型或 64 MB 纹理）包含在文件中以将它们编译为二进制文件，则即使构建应用程序也需要大量内存。.qrc

首先尝试优化资源，如优化设计和创建优化的 3D 场景中所述。

如果你真的需要大型资产，它们应该直接从文件系统加载，或者以动态方式使用Qt资源系统。有关更多信息，请参阅 Qt 文档中的 Qt 资源系统。

要使用Qt Quick UI项目中的自定义字体，请从*main.cpp*文件调用QFontDatabase：：addApplicationFont（）函数。

将Qt快速设计器组件添加到Qt安装中

如果您在项目中使用Qt Quick Studio组件或效果，则必须从QtCode Review中签出并安装*Qt Quick Designer Components*模块。

例如：

```
git clone https://code.qt.io/qt-labs/qtquickdesigner-components.git
```

然后使用Qt安装中的qmake来构建模块并将其添加到Qt中。 切换到包含源代码的目录（通常是qtquickdesigner-components），确保签出qmake分支，然后输入以下命令：

```
<path_to_qmake>\qmake -r
make
make install
```

在 Windows 上，请改用 theand命令。`nmakenmake install`

如果您更喜欢 CMake，并且希望从 QML 编译中受益，那么您可以改为查看开发分支。CMake 仅在 Qt 6.2 起受支持。输入以下命令：

```
mkdir build
cd build
cmake -GNinja -DCMAKE_INSTALL_PREFIX=<path_to_qt_install_directory> <path_to_qtquickdesigner-components>
cmake --build .
cmake --install .
```

将Qt快速时间轴模块添加到Qt安装中

> **注意：**仅当您的Qt版本低于5.14时，您才需要执行此操作。

查看Qt Code Review中的Qt Quick Timeline模块。

例如：

```
git clone "https://codereview.qt-project.org/qt/qtquicktimeline"
```

要使用 qmake，您需要签出包含 qmake 配置文件的分支或标签。

例如：

```
git checkout v5.15.2
```

然后构建模块并将其添加到Qt中，如上一节所述。