

🔍 搜索

Qt创作者手册 > [创建文件](#)

# 创建文件

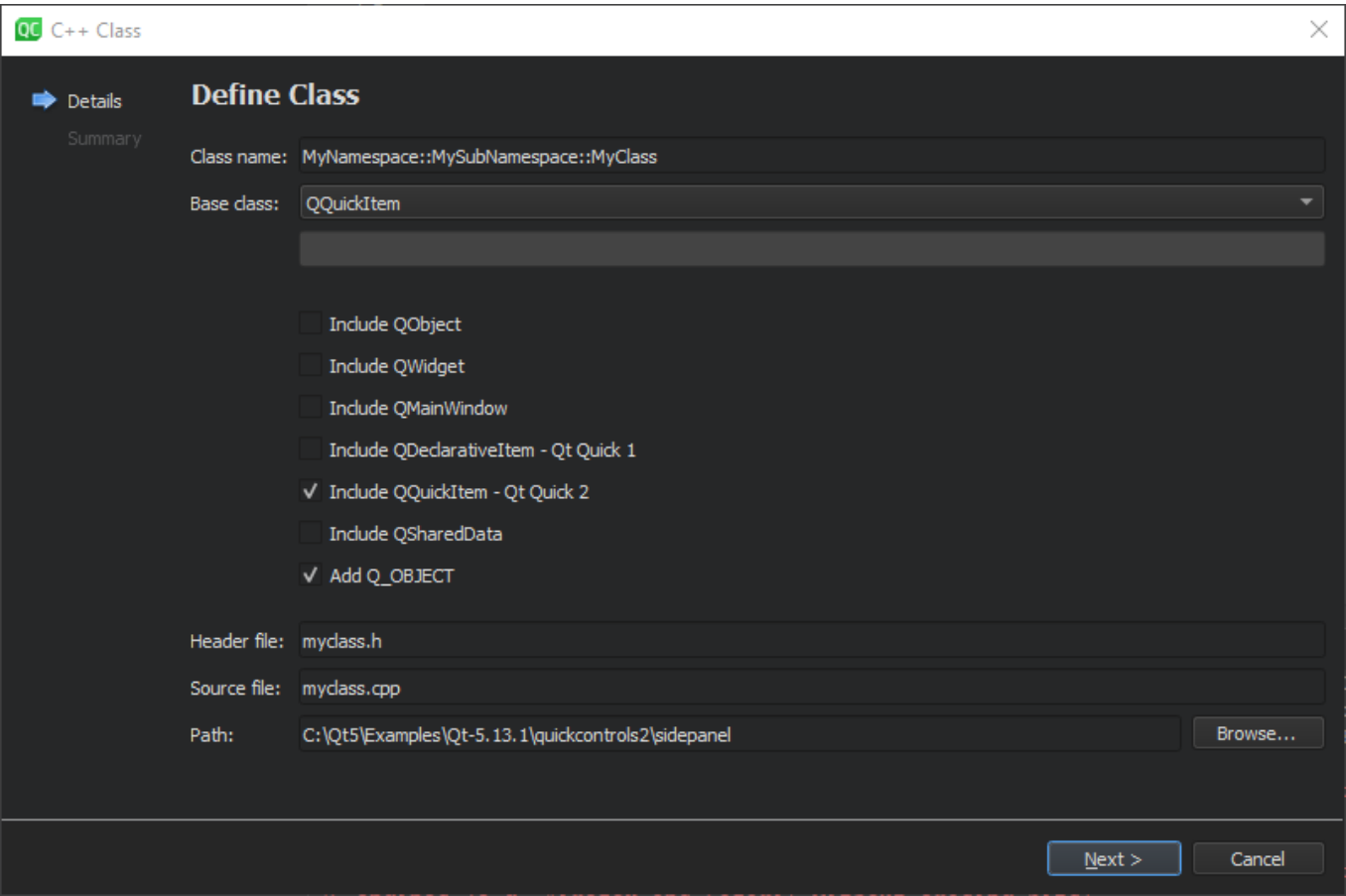
可以使用向导模板将单个文件添加到**项目**中。下表列出了用于创建文件的向导模板。

类别	向导模板	目的
C/C++	C++类	C++可以添加到C++项目中的新类的头文件和源文件。
	C/C++源文件	C++可以添加到C++项目的源文件。
	C/C++ 头文件	C++可以添加到C++项目的头文件。
建模	状态图	状态图 XML（SCXML）文件，其中包含状态机的样板代码。您可以使用Qt SCXML模块中的类嵌入从Qt应用程序中的文件创建的状态机。
	型	具有结构化图的通用建模语言（UML）样式模型。但是，模型编辑器使用UML的变体，并且仅提供用于指定模型元素外观的属性子集。有关详细信息，请参阅 <a href="#">建模</a> 。
	划痕模型	使用临时文件临时模型。
Qt	Qt项目模型	可用于创建派生自QAbstractItemModel、QAbstractTableModel或QAbstractListModel的类的源文件和头文件。
	Qt设计器表单类	Qt Designer表单和用于实现基于Qt小部件的UI的匹配类。
	Qt设计师表格	Qt设计器表单，用于基于Qt小部件的项目。如果已有UI逻辑的现有类，这将非常有用。
	Qt 资源文件	用于在应用程序可执行文件中存储二进制文件的资源文件。
	QML 文件（Qt Quick 2）	导入Qt Quick 2.0以用于Qt Quick Project的QML文件。
	Qt快速UI文件	UI文件（.ui.qml）和相应的实现文件（.qml），用于Qt Quick项目。
	JS文件	可用于在Qt Quick项目中编写应用程序逻辑的JavaScript文件。
GLSL	片段着色器（OpenGL/ES 2.0）	片段着色器，用于生成使用OpenGL渲染的三角形、点和线的最终像素颜色。您可以在Qt Quick项目和基于Qt小部件的项目中使用它。
	顶点着色器（OpenGL/ES 2.0）	顶点着色器，用于转换使用OpenGL渲染的三角形、点和线的位置、法线和纹理坐标。您可以在Qt Quick项目和基于Qt小部件的项目中使用它。

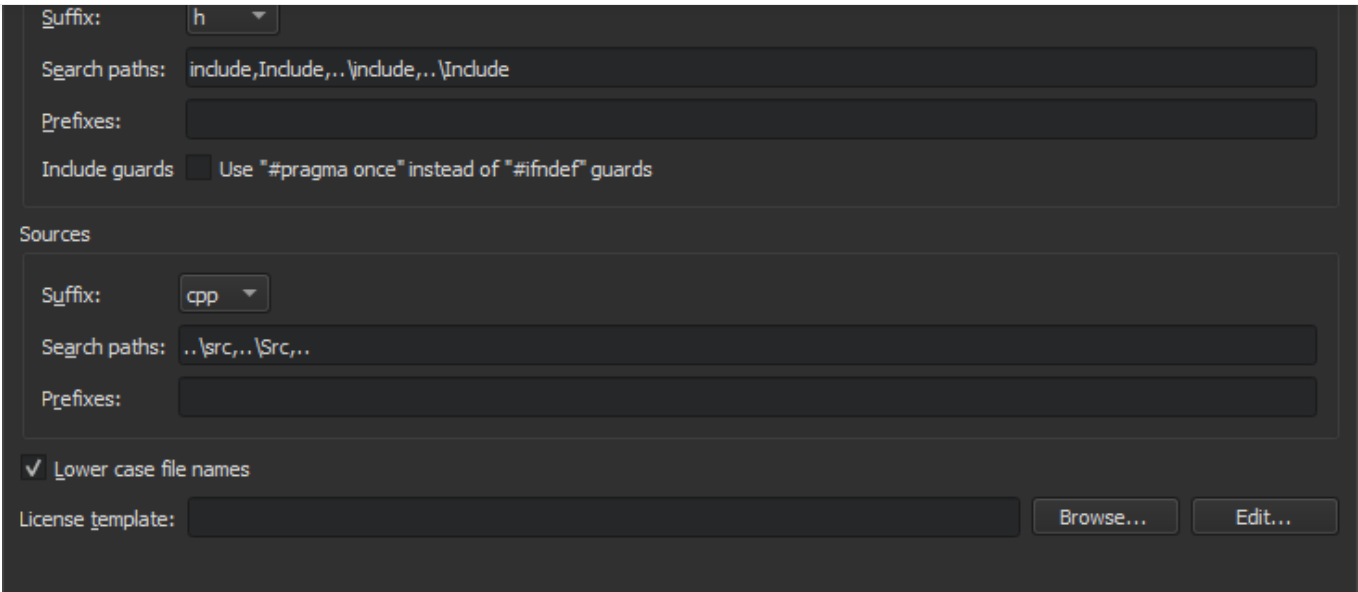
	顶点着色器 (桌面 OpenGL)	顶点着色器，用于Qt Quick项目和基于Qt小部件的项目。
常规	空文件	可以使用任何文件名扩展保存的空文件。
	划痕缓冲区	使用临时文件的暂存缓冲区。您可以创建此类文件来临时存储您不打算保存的信息
爪哇岛	爪哇文件	可用于创建 Java 类的 Java 类文件。
蟒	蟒蛇类	Python 类文件。
	蟒蛇文件	使用 UTF-8 编码的 Python 脚本文件。
尼姆 (实验性)	尼姆脚本文件	使用 UTF-8 编码的空 Nim 脚本文件。
	尼姆文件	使用 UTF-8 编码的空 Nim 源文件。

**C++类向导**允许您为可添加到C++项目中的新类创建C++头文件和源文件。指定类的类名、基类以及头文件和源文件。

该向导支持命名空间。要使用命名空间，请在“类名”字段中输入限定的**类名**。例如：`MyNamespace::MySubNamespace::MyClass`



头文件和源文件的名称基于类名。要更改文件的默认后缀，请选择“**编辑>首选项>C++>文件命名**”。

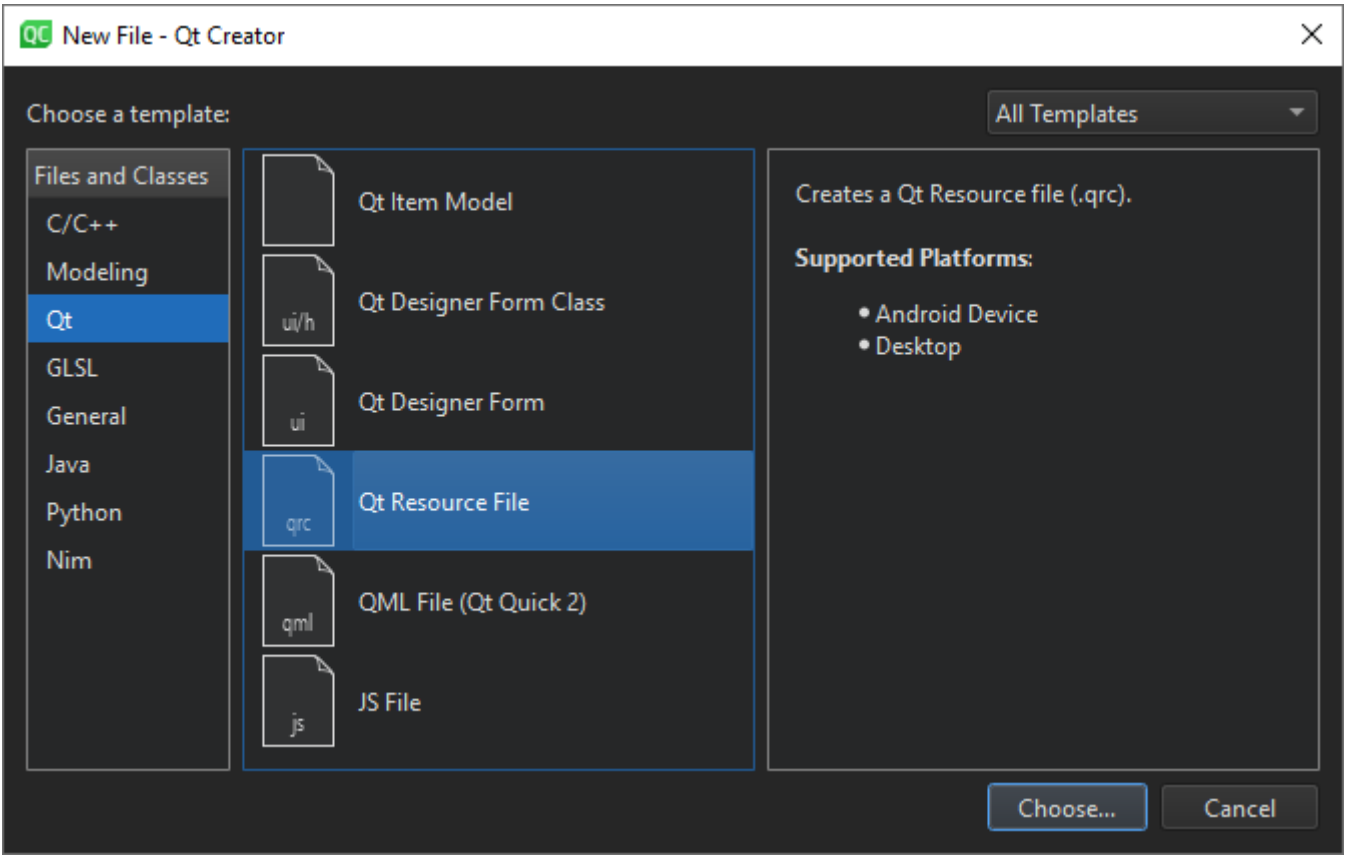


在许可证模板字段中，您可以使用[预定义的向导变量](#)指定要在源文件和头文件中使用的许可证的路径和文件名。

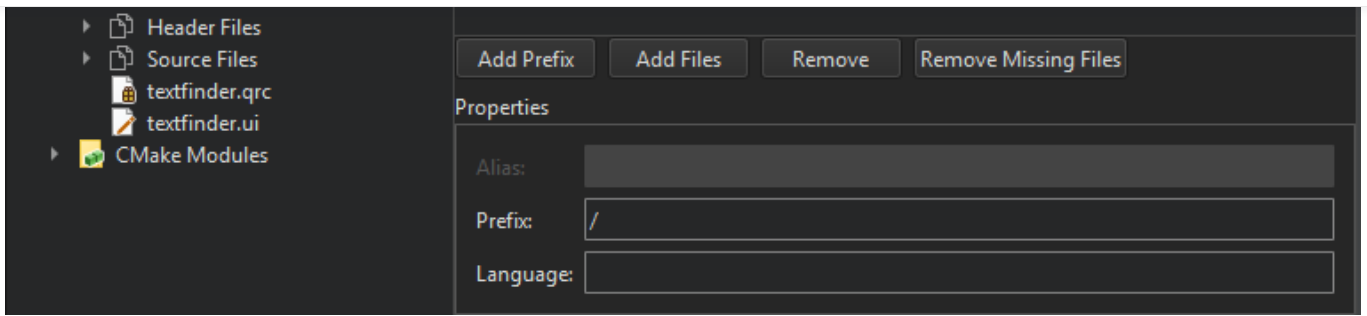
您可以创建自己的项目和类向导。有关详细信息，请参阅[添加新的自定义向导](#)。

## 创建资源文件

Qt Creator支持[Qt资源系统](#)，这是一种独立于平台的机制，用于在应用程序的可执行文件中存储文件。



该向导将创建一个可在资源编辑器中管理的资源集合文件（.qrc）。



选择“添加文件”以查找并添加单个文件。

若要在源树中按字母升序列出文件夹和文件，请在上下文菜单中选择“按字母顺序排序”。

默认情况下，资源在应用程序中的访问文件名与源树中的文件名相同，带有前缀，或者通过带有方案的 URL 访问。若要为文件中的所有文件指定路径前缀，请选择“添加前缀”，然后在“前缀”字段中输入前缀。:/qrc.qrc

某些资源需要根据用户的区域设置进行更改，例如翻译文件或图标。您可以在“语言”字段中指定区域设置。

选择“删除”以从资源集中删除所选文件。在“删除文件”对话框中，选中“永久删除文件”复选框以从文件系统中删除该文件。要删除在文件系统中找不到的文件，请选择“删除丢失的文件”。

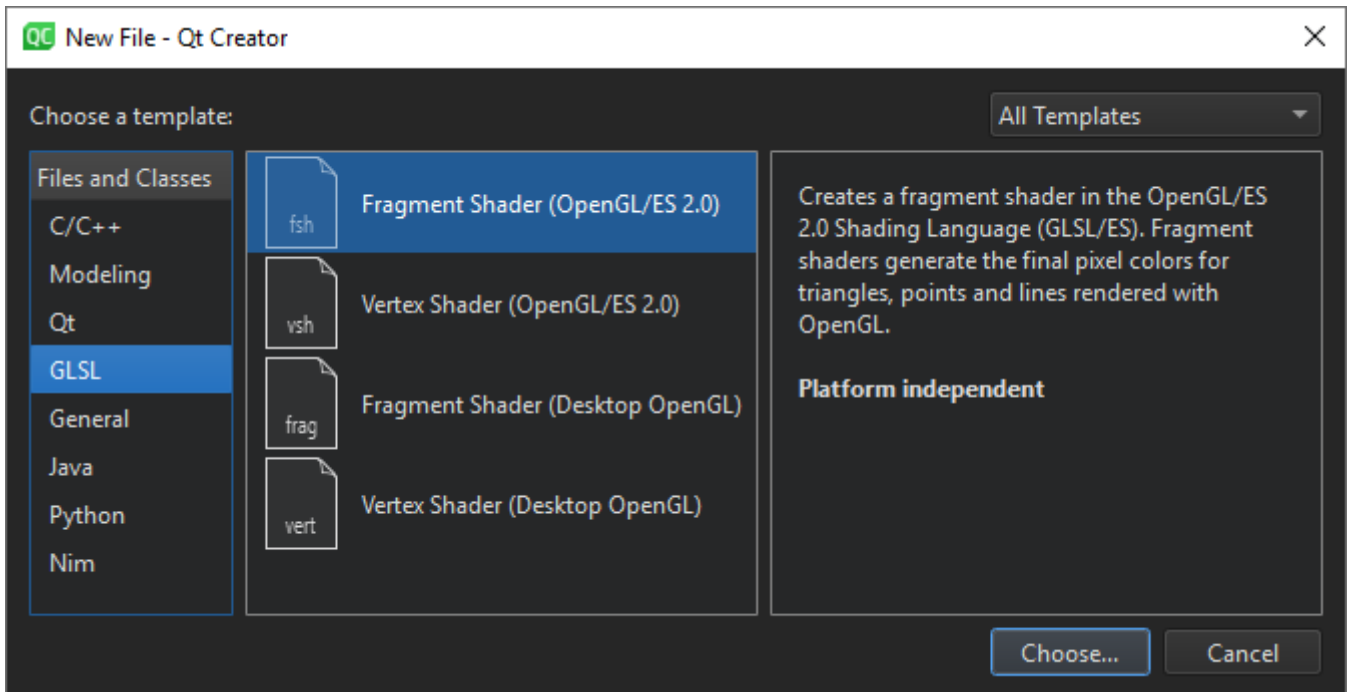
上述函数在“项目”视图的上下文菜单中也可用。

## 创建 OpenGL 片段和顶点着色器

Qt支持与所有平台上的OpenGL实现集成，这允许您在更传统的用户界面旁边显示硬件加速的3D图形。有关更多信息，请参阅Qt GUI。

您可以使用 QOpenGLShader 类编译用 OpenGL 着色语言（GLSL）和 OpenGL/ES 着色语言（GLSL/ES）编写的 OpenGL 着色器。QOpenGLShader 和 QOpenGLShaderProgram 可保护您免受编译和链接顶点和片段着色器的详细信息。

您可以使用Qt Creator代码编辑器在GLSL或GLSL/ES中编写片段和顶点着色器。代码编辑器为文件提供语法突出显示和代码完成。



©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。



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