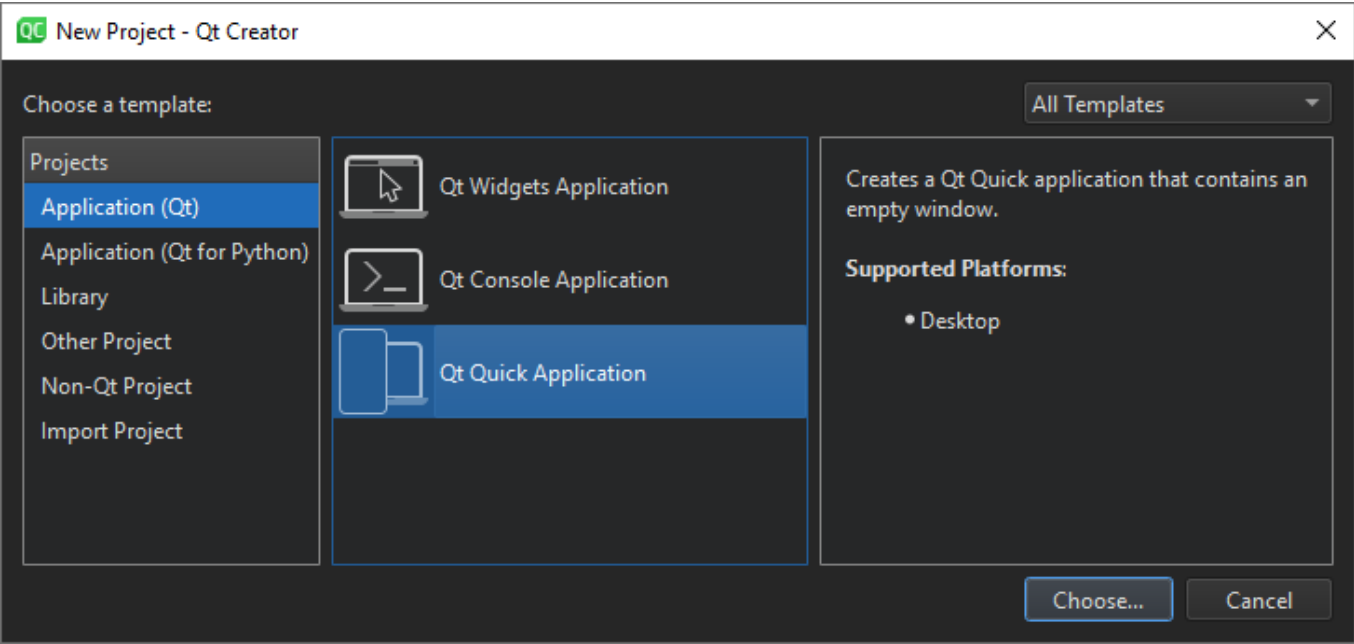


# 创建Qt快速项目



下表列出了用于从头开始创建新的 Qt Quick 项目的向导模板。

类别	向导模板	目的
应用（批量）	Qt快速申请	创建一个 Qt Quick 2 应用程序项目，该项目可以同时包含 QML 和C++代码。您可以构建应用程序并将其部署到桌面、嵌入式和移动目标平台。
应用程序（Qt为蟒蛇）	Qt 为蟒蛇 - Qt 快速应用	创建一个包含空 Qt 快速应用程序的 Python 项目。
其他项目	Qt 快速用户界面原型	创建一个 Qt 快速 UI 项目，其中包含一个包含主视图的 QML 文件。您可以在 QML 场景预览工具中预览 Qt 快速 2 UI 项目。您不需要生成它们，因为它们不包含任何 C++代码。 此项目类型与Qt设计工作室兼容。但是，仅当要进行原型设计时，才使用此模板。不能使用此模板创建完整的应用程序。 Qt快速UI项目无法部署到嵌入式或移动目标平台。对于这些平台，请改为创建 Qt 快速应用程序。
图书馆	Qt 快速 2 扩展插件	创建C++插件，以提供可以动态加载到Qt Quick 2应用程序中的扩展。

**注意：** 特定目标平台的 SDK 可能会为该平台安装其他模板。例如，QNX 模板是作为 QNX 开发工具包的一

Qt 创建器创建必要的样板文件。某些文件特定于特定的目标平台。

## 创建Qt快速应用程序

1. 选择“**文件**>**新项目**>**应用程序 (Qt)** > **Qt 快速应用程序**>**选择**”。
2. 在“**项目位置**”对话框的“**名称**”字段中，输入项目的名称。请记住，以后无法轻松更改项目名称。
3. 在“**创建位置**”字段中，输入项目文件的路径。选中“**用作默认项目位置**”复选框以默认在此文件夹中创建新项目。您可以稍后移动项目文件夹而不会出现问题。
4. 选择“**下一步**”（或在 macOS 上选择“**继续**”）以打开“**定义生成系统**”对话框。
5. 在“**生成系统**”字段中，选择要用于生成和运行项目的生成系统：**qmake**、**CMake** 或 **Qbs**。
6. 选择“**下一步**”以打开“**定义项目详细信息**”对话框。
7. 在“所需的最低 Qt 版本”字段中选择要开发的 **Qt 版本**。Qt 版本确定 QML 文件中使用的 Qt 快速导入。
8. 选中“**使用 Qt 虚拟键盘**”复选框，将对 **Qt 虚拟键盘** 的支持添加到应用程序中。

**注意：**如果在安装 Qt 时尚未安装 Qt 虚拟键盘模块，则当您尝试打开 *main.qml* 进行编辑时，将出现一条错误消息。您可以使用**Qt维护工具**安装Qt虚拟键盘。

9. 选择“**下一步**”以打开“**翻译文件**”对话框。
10. 在“**语言**”字段中，选择计划将应用程序**翻译成**的语言。您可以稍后通过编辑项目文件来添加其他语言。
11. 在“**翻译文件**”字段中，您可以编辑将为所选语言生成的翻译源文件的名称。
12. 选择“**下一步**”以打开“**工具包选择**”对话框。
13. 为要为其构建应用程序的平台选择**工具包**。

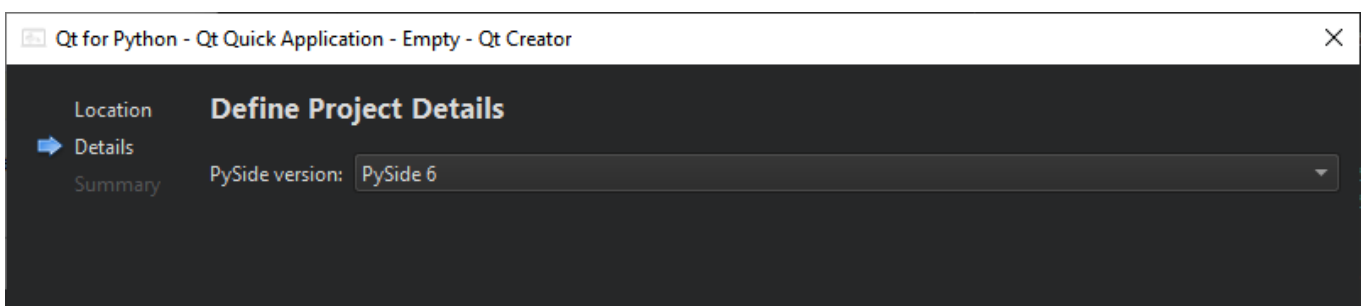
**注意：**如果在“**编辑**>**首选项**>**工具包**”（在 Windows 和 Linux 上）或“**Qt 创建者**>**首选项**>**工具包**”（在 macOS 上）中指定了工具包，则会列出**这些工具包**。有关详细信息，请参阅**添加工具包**。

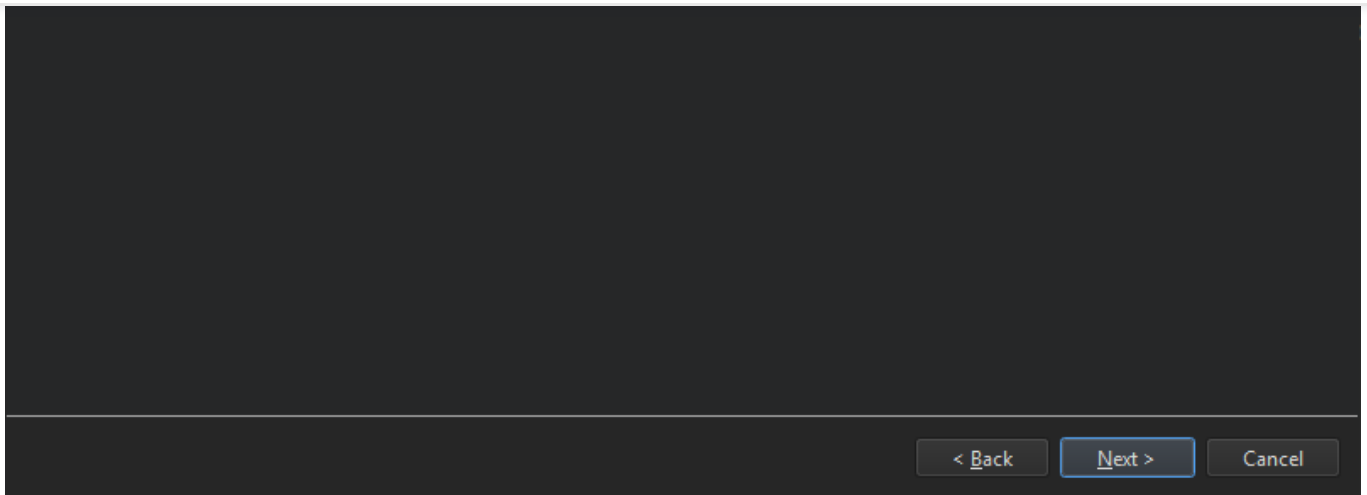
14. 选择“**下一步**”以打开“**项目管理**”对话框。
15. 查看项目设置，然后选择“**完成**”（在 Windows 和 Linux 上）或“**完成**”（在 macOS 上）以创建项目。

Qt 创建器创建一个 QML 文件 *main.qml*，您可以在**编辑**模式下修改该文件。

## Creating Qt Quick Based Python Applications

The **Qt for Python - Qt Quick Application - Empty** wizard enables you to create a Python project that contains a main QML file. Specify the minimum PySide version to run the application.





The wizard adds the following imports to the source file to provide access to `QGuiApplication` and `QQmlApplicationEngine`:

```
import sys
from pathlib import Path

from PySide6.QtGui import QGuiApplication
from PySide6.QtQml import QQmlApplicationEngine
```

The wizard also adds a main function, where it creates a `QGuiApplication` instance and passes system arguments to the `QGuiApplication` object:

```
if __name__ == "__main__":
    app = QGuiApplication(sys.argv)
    ...
```

The following lines in the main class create a `QQmlApplicationEngine` instance and load the generated QML file to the engine object:

```
engine = QQmlApplicationEngine()
qml_file = Path(__file__).resolve().parent / "main.qml"
engine.load(qml_file)
```

Finally, the wizard adds code that checks whether the file was successfully loaded. If loading the file fails, the application exits with an error code. If loading succeeds, the wizard calls the method to enter the Qt main loop and start executing the Qt code: `app.exec()`

```
if not engine.rootObjects():
    sys.exit(-1)
sys.exit(app.exec())
```

## Creating Qt Quick UI Projects

Qt Quick UI Prototype projects are useful for testing or prototyping user interfaces, or for setting up a separate project just for QML editing, for example. You cannot use them for application development because they do not contain:

- › C++ code
- › Resource files (.qrc)
- › Code needed for deploying applications to [devices](#)

For more information about how to turn Qt Quick UI Prototype projects into Qt Quick Application projects, see [Converting UI Projects to Applications](#).

To create a Qt Quick UI Prototype project:

1. Select **File > New Project > Other Project > Qt Quick UI Prototype**.
2. Select **Choose** to open the **Project Location** dialog.
3. In the **Name** field, enter a name for the application.
4. In the **Create in** field, enter the path for the project files. Select the **Use as default project location** check box to create new projects in this folder by default.
5. Select **Next** (or **Continue** on macOS) to open the **Define Project Details** dialog.
6. In the **Minimum required Qt version** field, select the Qt version to develop with. The Qt version determines the Qt Quick imports that are used in the QML files.

You can add imports later to combine Qt Quick basic types with Qt Quick Controls, Qt Quick Dialogs, and Qt Quick Layouts (available since Qt 5.1).

7. Select the **Use Qt Virtual Keyboard** check box to add support for [Qt Virtual Keyboard](#) to the application.

**Note:** If you have not installed the Qt Virtual Keyboard module when you installed Qt, an error message will appear when you try to open *main.qml*.

8. Select **Next** to open the **Kit Selection** dialog.
9. Select [kits](#) for the platforms that you want to build the application for.

**Note:** Kits are listed if they have been specified in **Edit > Preferences > Kits** (on Windows and Linux) or in **Qt Creator > Preferences > Kits** (on macOS). For more information, see [Adding Kits](#).

10. Select **Next** to open the **Project Management** dialog.
11. Review the project settings, and select **Finish** (on Windows and Linux) or **Done** (on macOS) to create the project.

Qt Creator creates the following files:

- › .qmlproject project file defines that all QML, JavaScript, and image files in the project folder belong to the project. Therefore, you do not need to individually list all the files in the project.
- › .qml file defines a UI item, such as a component or the whole application UI.

To use JavaScript and image files in the application, copy them to the project folder.

[< Developing Qt Quick Applications](#)

[Using Qt Quick Designer >](#)

© 2022 The Qt Company Ltd. Documentation contributions included herein are the copyrights of their respective owners. The documentation provided herein is licensed under the terms of the [GNU Free Documentation License version 1.3](#) as published by the Free Software Foundation. Qt and respective logos are trademarks of The Qt Company Ltd in Finland and/or other countries worldwide. All other trademarks are property of their respective owners.



Contact Us

## Company

[About Us](#)  
[Investors](#)  
[Newsroom](#)  
[Careers](#)  
[Office Locations](#)

## Licensing

[Terms & Conditions](#)  
[Open Source](#)  
[FAQ](#)

## 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](#)

