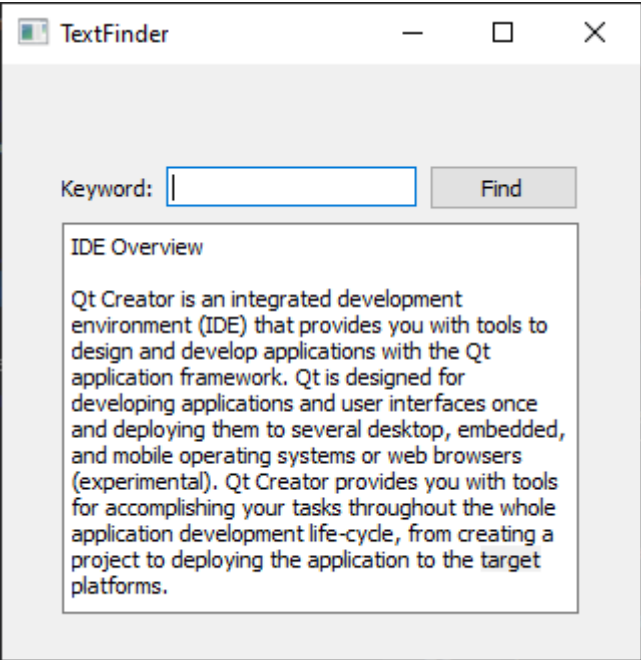


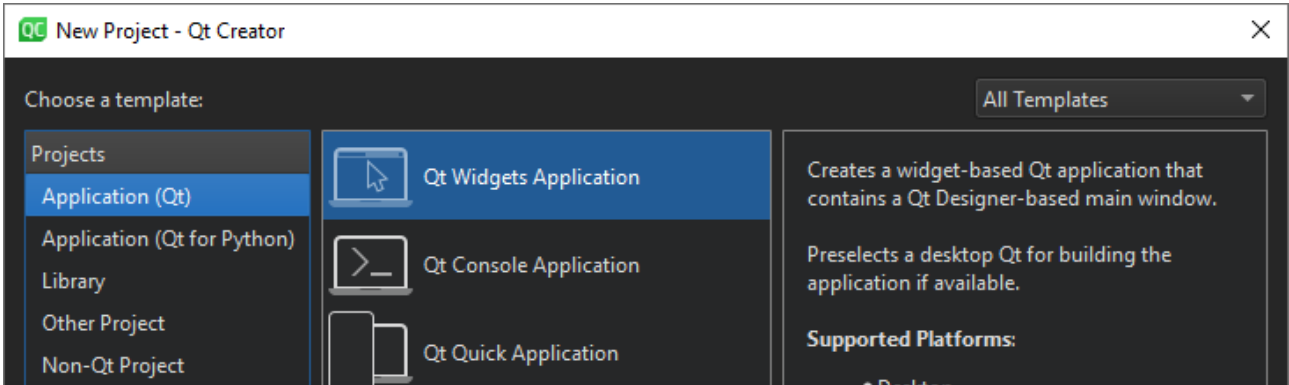
创建基于 Qt Widget 的应用程序

本教程介绍如何使用Qt Creator创建一个小型Qt应用程序Text Finder。它是Qt UI工具文本查找器示例的简化版本。应用程序用户界面是使用Qt Designer从Qt小部件构造的。应用程序逻辑是使用代码编辑器以C++编写的。



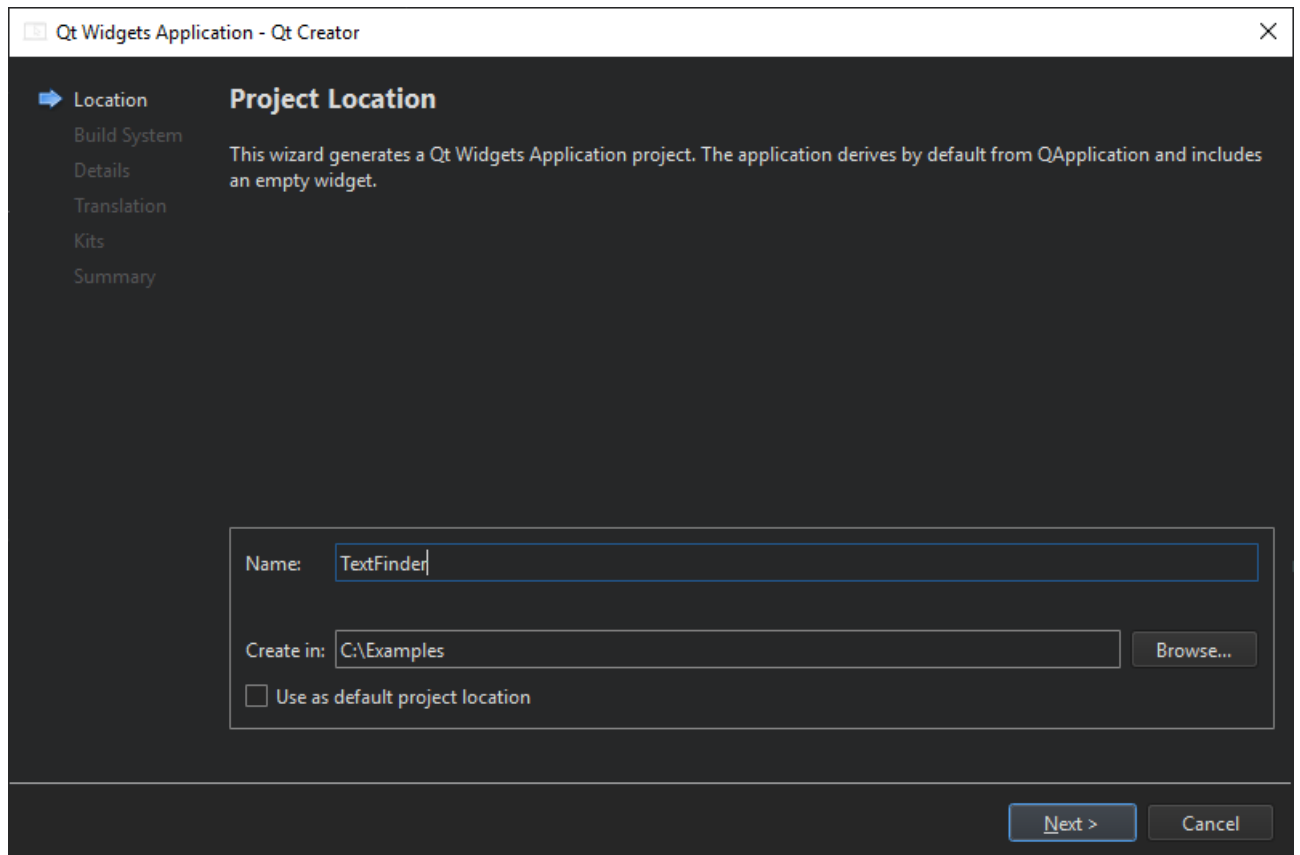
创建文本查找器项目

1. 选择文件>新建项目>应用程序（Qt）>Qt 小部件应用程序>选择。

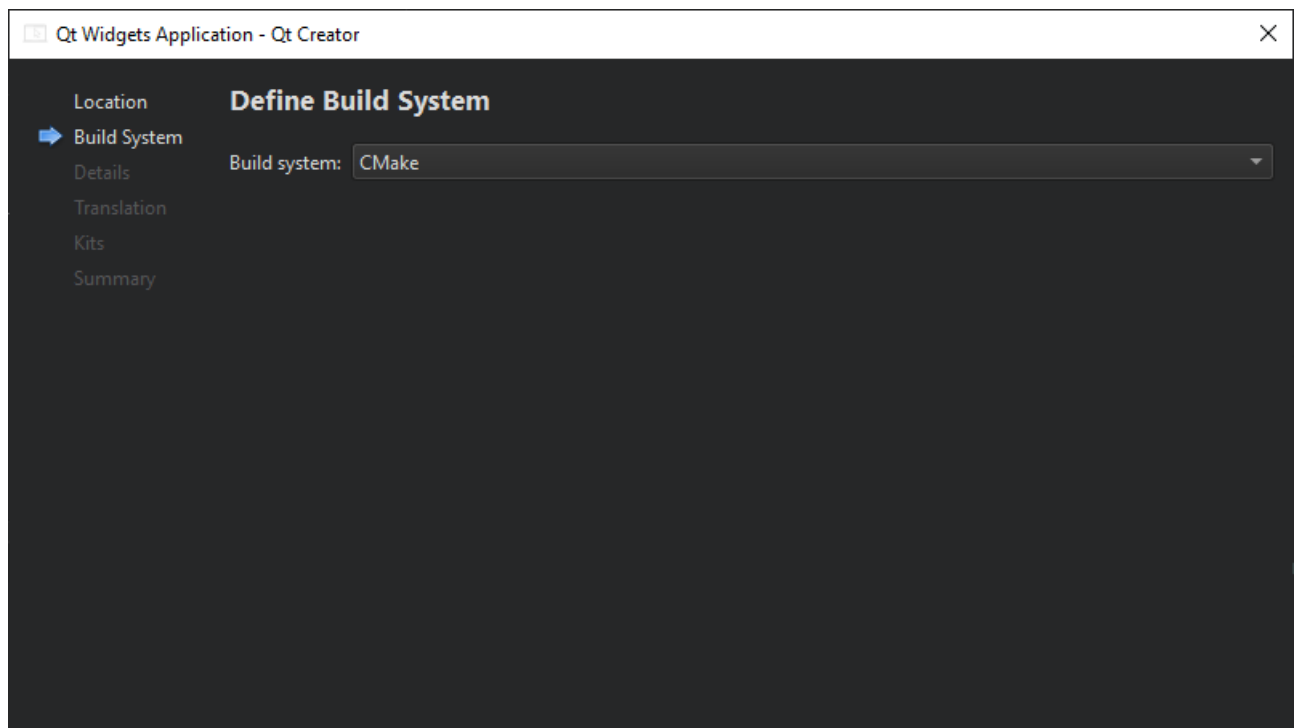




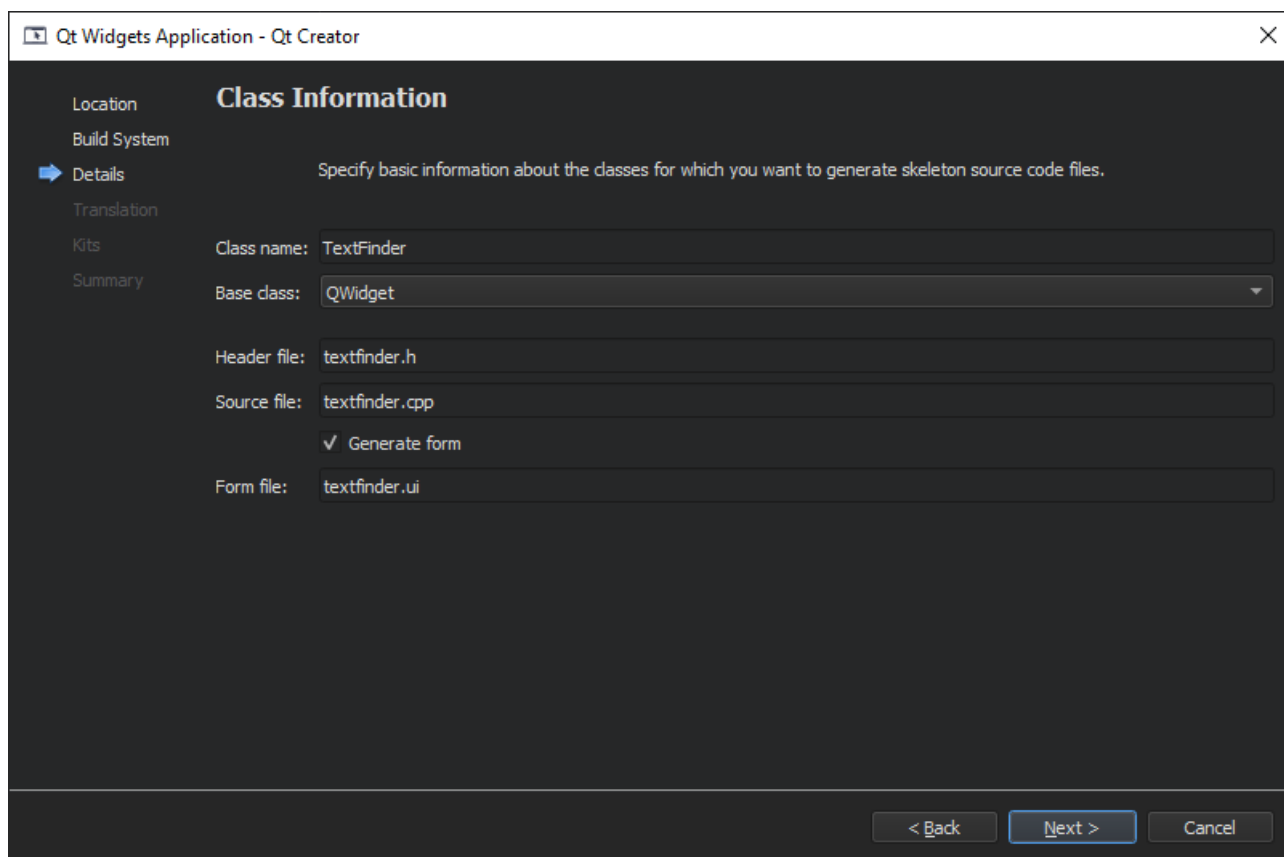
将打开“简介和项目位置”对话框。



2. 在“名称”字段中，键入“文本查找器”。
3. 在“创建位置”字段中，输入项目文件的路径。例如。C:\Qt\examples
4. 选择“下一步”（在 Windows 和 Linux 上）或“继续”（在 macOS 上）以打开“定义生成系统”对话框。



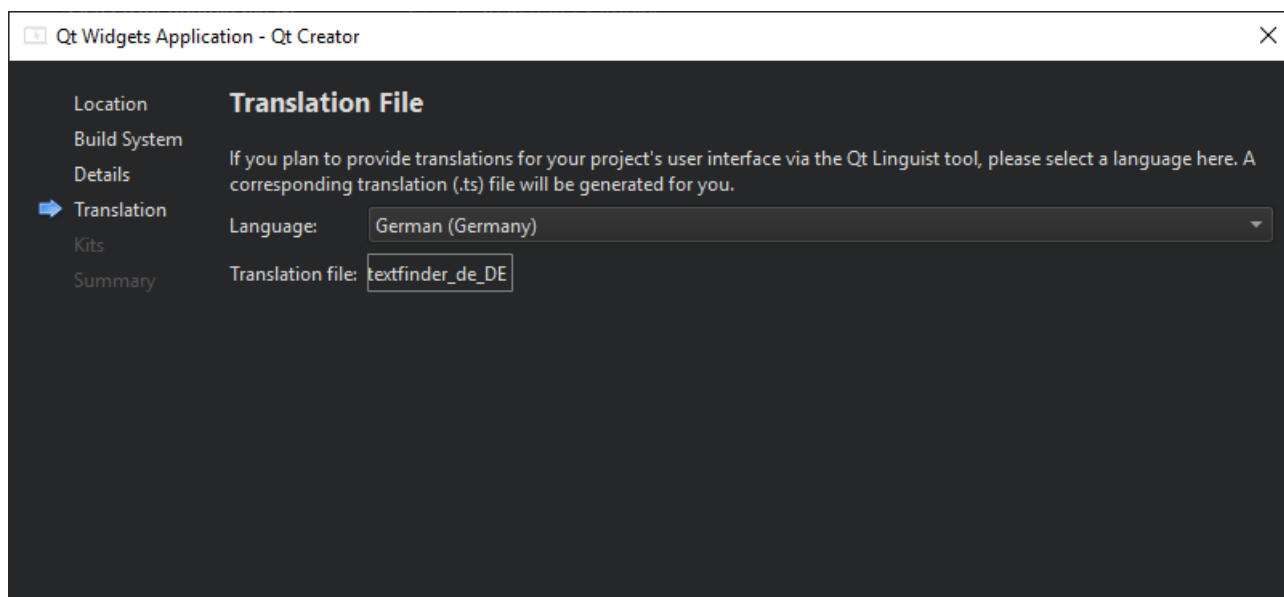
5. 在“生成系统”字段中，选择“CMake”作为用于生成项目的**生成系统**。
6. 选择“**下一步**”或“继续”以打开“**类信息**”对话框。

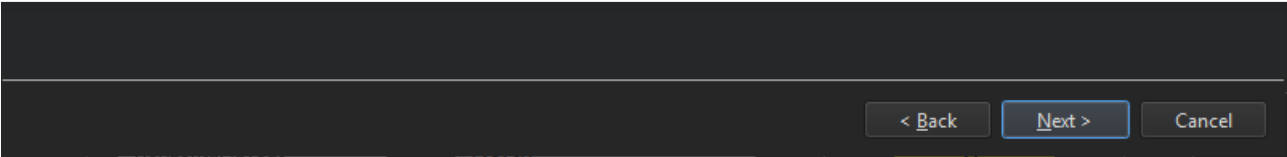


7. 在“类名”字段中，键入“**文本查找器**”作为类名。
8. 在“基类”列表中，选择“**QWidget**”作为基类类型。

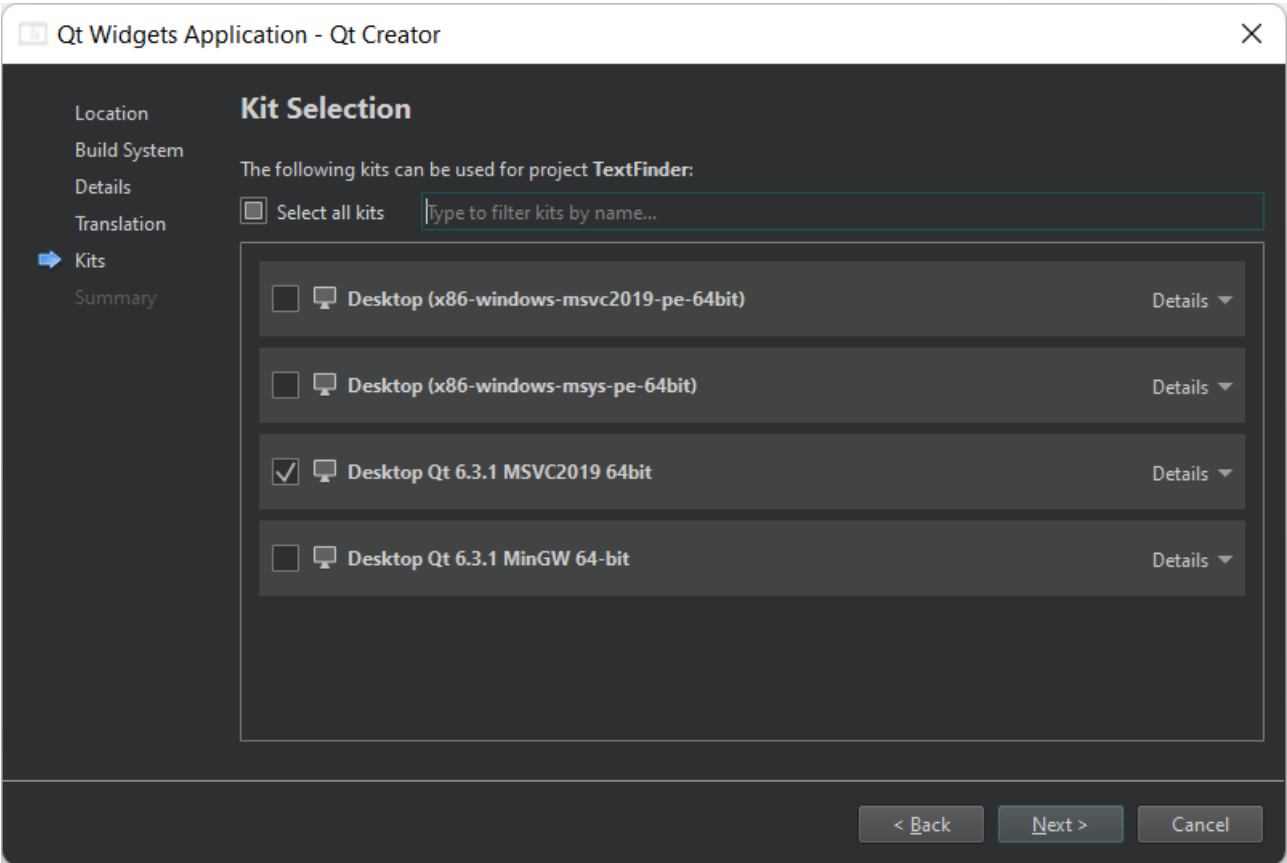
注意：“头文件”、“源文件”和“表单文件”字段将自动更新以匹配类的名称。

9. 选择“**下一步**”或“继续”以打开“**翻译文件**”对话框。

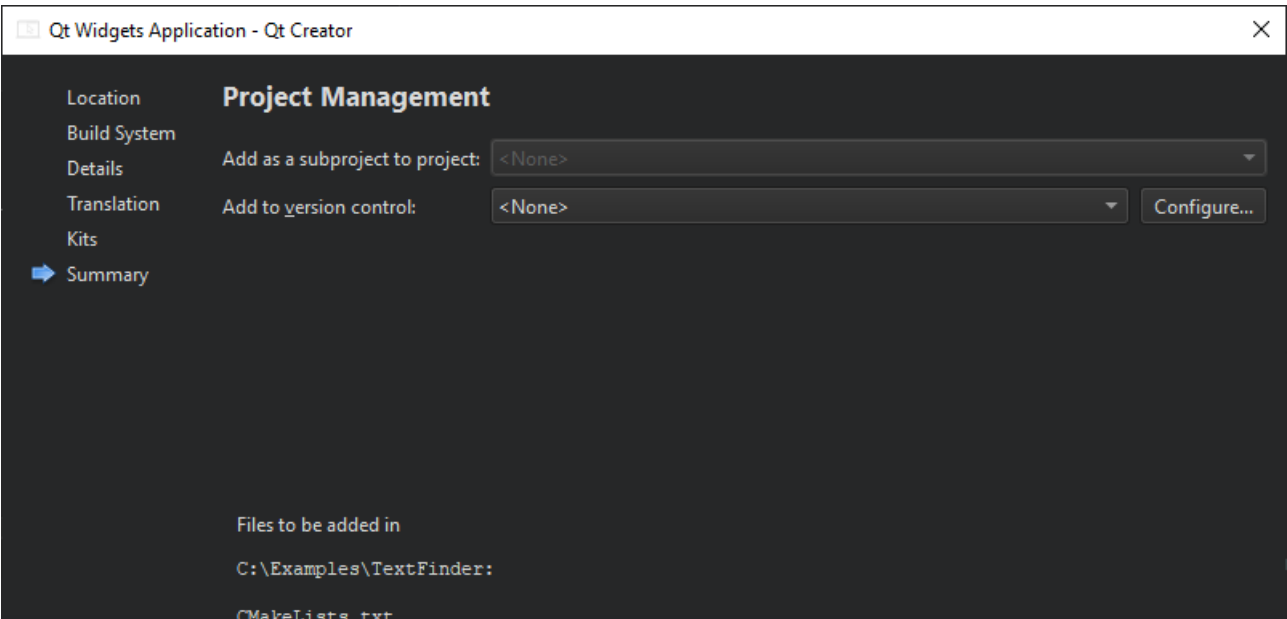


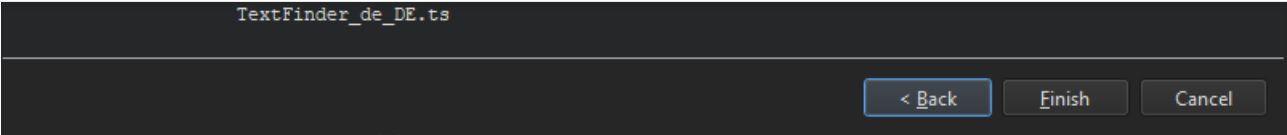


10. 在“语言”字段中，您可以选择计划将应用程序翻译成语言。这将为应用程序设置本地化支持。您可以稍后通过编辑项目文件来添加其他语言。
11. 选择“下一步”或“继续”以打开“套件选择”对话框。



12. 为项目选择生成并运行工具包。
13. 选择“下一步”或“继续”以打开“项目管理”对话框。



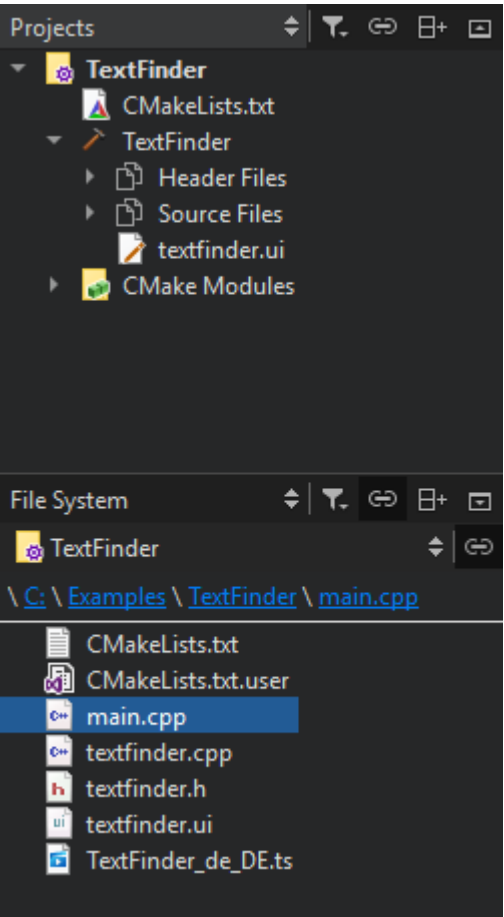


14. 查看项目设置，然后选择“**完成**”（在 Windows 和 Linux 上）或“**完成**”（在 macOS 上）以创建项目。

注意：项目将在“**编辑**”模式下打开，并且这些说明处于隐藏状态。若要返回到这些说明，请打开“**帮助**”模式。

TextFinder 项目现在包含以下文件：

- 主.cpp
- 文本查找器.h
- 文本查找器.cpp
- textfinder.ui
- CMakeList.txt

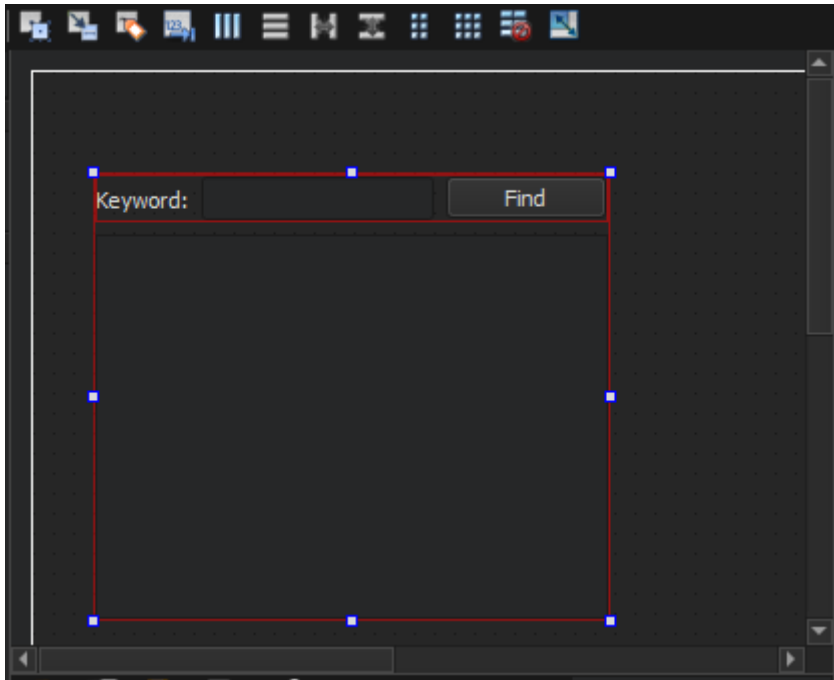


.h 和 .cpp 文件附带必要的样板代码。

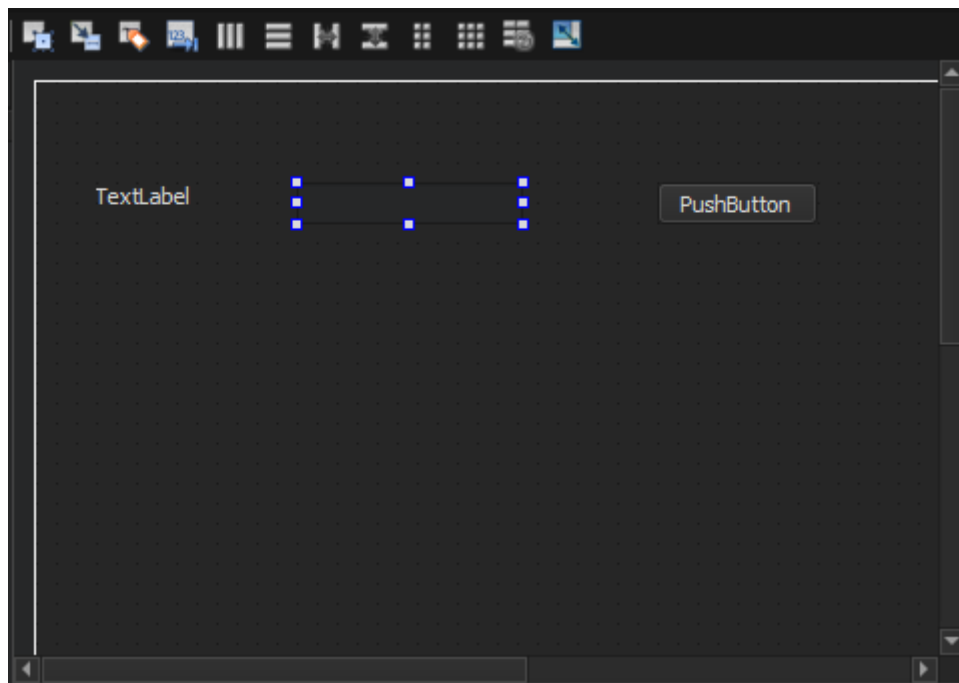
如果您选择CMake作为构建系统，Qt Creator会为您创建一个CMakeLists.txt项目文件。

填补缺失的部分

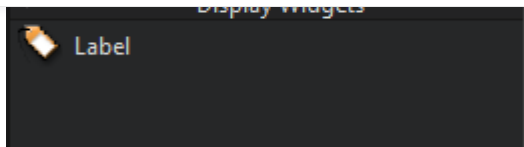
首先设计用户界面，然后继续编写缺失的代码，最后，添加查找功能。



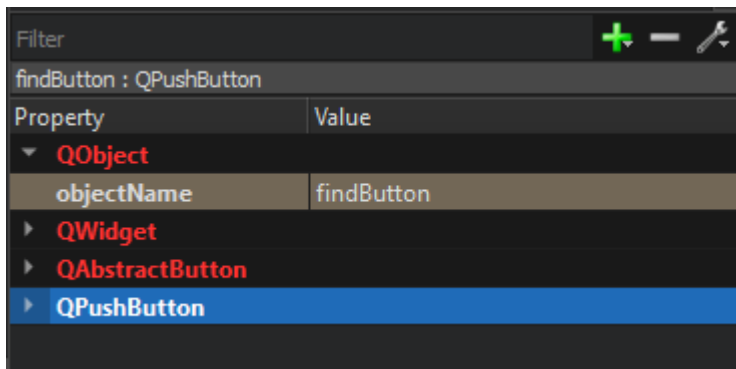
1. 在编辑器模式下，双击“项目”视图中的 textfinder.ui 文件以启动集成的 Qt 设计器。
2. 将以下小部件拖放到表单中：
 - 标签 (QLabel)
 - 行编辑 (QLineEdit)
 - 按钮 (QPushButton)



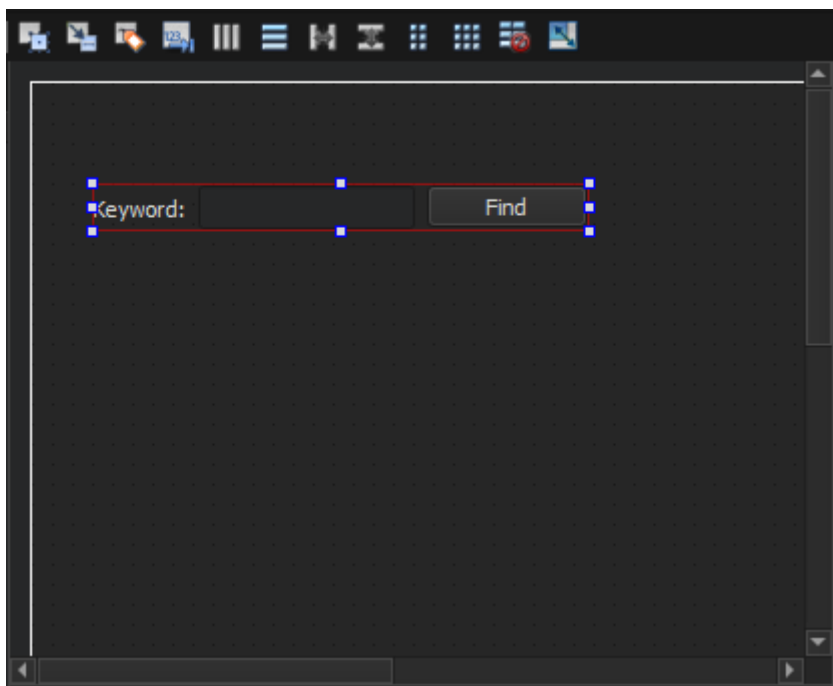
注意：要轻松找到小部件，请使用**边栏**顶部的搜索框。例如，要查找“标签”微件，请开始键入单词**标签**。



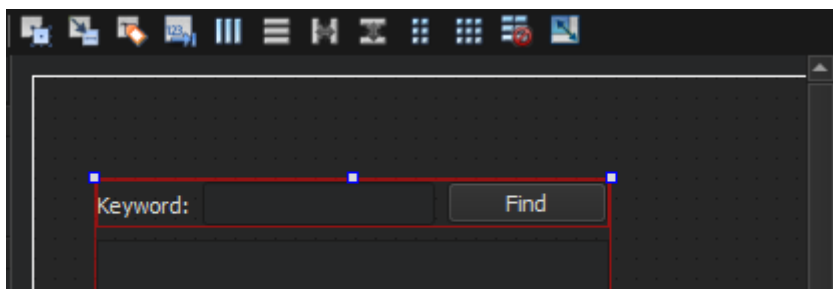
3. 双击**标注**微件并输入文本**关键字**。
4. 双击“**按钮**”微件，然后输入文本**查找**。
5. 在“**属性**”视图中，将**对象名称**更改为findButton。

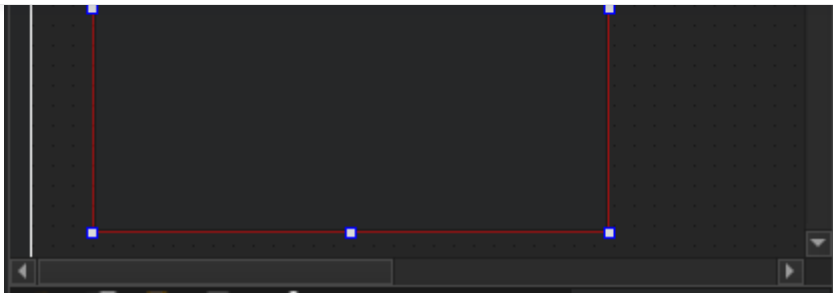


6. 按 Ctrl+A (或Cmd+A) 选择小部件，然后选择“水平布局”（或在 Linux 或 Windows 上按 Ctrl+H，或在 macOS 上按Ctrl+Shift+H）以应用**水平布局**（**QHBoxLayout**）。



7. 将“**文本编辑**”小部件（**QTextEdit**）拖放到表单中。
8. 选择屏幕区域，然后选择“**垂直布局**”（或按Ctrl+L）以应用垂直布局（**QVBoxLayout**）。





应用水平和垂直布局可确保应用程序 UI 缩放到不同的屏幕大小。

9. 要在用户选择“查找”按钮时调用**查找**函数，请使用 Qt 信号和插槽机制。当发生特定事件时发出信号，间隙是为响应特定信号而调用的函数。Qt 小部件具有预定义的信号和插槽，您可以直接从 Qt Designer 使用。要为 find 函数添加插槽：

- › 右键单击“**查找**”按钮以打开上下文菜单。
- › 选择“**转到插槽>单击 ()**”，然后选择“**确定**”。

将专用插槽添加到头文件 textfinder.h 中，将专用函数添加到源文件 textfinder.cpp。
on_findButton_clicked()TextFinder::on_findButton_clicked()

10. 按Ctrl+S（或Cmd+S）保存更改。

有关使用 Qt Designer 设计表单的更多信息，请参阅[Qt Designer 手册](#)。

完成头文件

textfinder.h 文件已经具有必要的#include、构造函数、析构函数和对象。您需要添加一个私有函数，以在 `QTextEdit` 中读取和显示输入文本文件的内容。UiloadTextFile()

1. 在“编辑”视图的“**项目**”视图中，双击该文件以将其打开以进行编辑。textfinder.h
2. 将私有函数添加到该节的指针之后，如以下代码片段所示：privateUi::TextFinder

```
private slots:
    void on_findButton_clicked();

private:
    Ui::TextFinder *ui;
    void loadTextFile();
```

完成源文件

现在头文件已完成，请转到源文件textfinder.cpp。

1. 在“编辑”视图的“**项目**”视图中，双击文本查找器.cpp文件以将其打开以进行编辑。
2. 添加代码以使用 `QFile` 加载文本文件，使用 `QTextStream` 读取它，然后使用 `QTextEdit::setPlainText()` 显示它。以下代码片段对此进行了说明：textEdit

```
void TextFinder::loadTextFile()
{
    QFile inputFile(":/input.txt");
    inputFile.open(QIODevice::ReadOnly);
```



```

        QTextCursor cursor = ui->textEdit->textCursor();
        cursor.movePosition(QTextCursor::Start, QTextCursor::MoveAnchor, 1);
    }
}

```

3. 要使用 `QFile` 和 `QTextStream`，请将以下 `#includes` 添加到 `textfinder.cpp`：

```

#include "../ui_textfinder.h"
#include <QFile>
#include <QTextStream>

```

4. 对于插槽，添加代码以提取搜索字符串，并使用 `QTextEdit::find()` 函数在文本文件中查找搜索字符串。以下代码片段对此进行了说明：`on_findButton_clicked()`

```

void TextFinder::on_findButton_clicked()
{
    QString searchString = ui->lineEdit->text();
    ui->textEdit->find(searchString, QTextDocument::FindWholeWords);
}

```

5. 完成这两个函数后，添加一行来调用构造函数，如以下代码片段所示：`loadTextFile()`

```

TextFinder::TextFinder(QWidget *parent)
    : QWidget(parent)
    , ui(new Ui::TextFinder)
{
    ui->setupUi(this);
    loadTextFile();
}

```

Theslot 在 `uic` 生成的 `ui_textfinder.h` 文件中通过以下代码行自动调用：`on_findButton_clicked()`

```

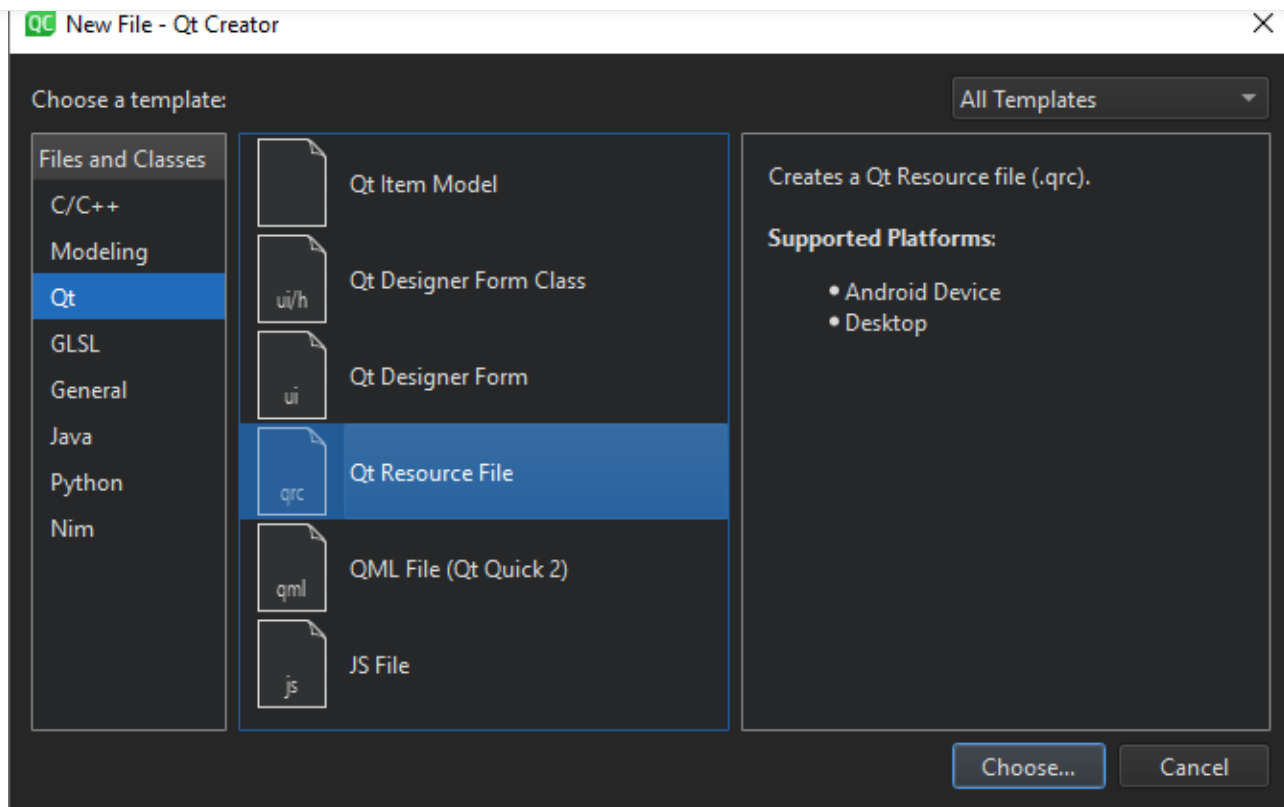
QMetaObject::connectSlotsByName(TextFinder);

```

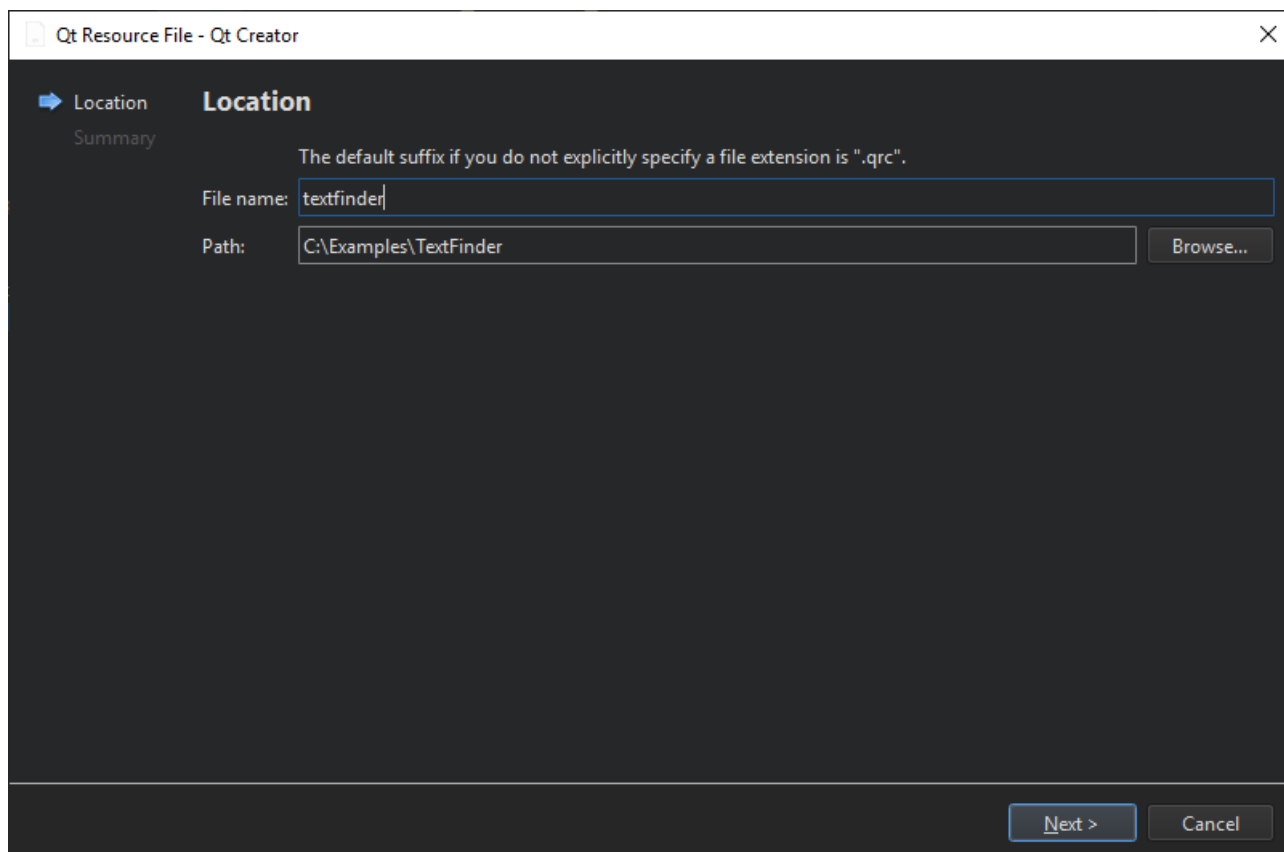
创建资源文件

您需要一个资源文件（.qrc），在其中嵌入输入文本文件。输入文件可以是任何带有一段文本的.txt文件。创建一个名为 `input.txt` 的文本文件，并将其存储在 `textfinder` 文件夹中。

添加资源文件：

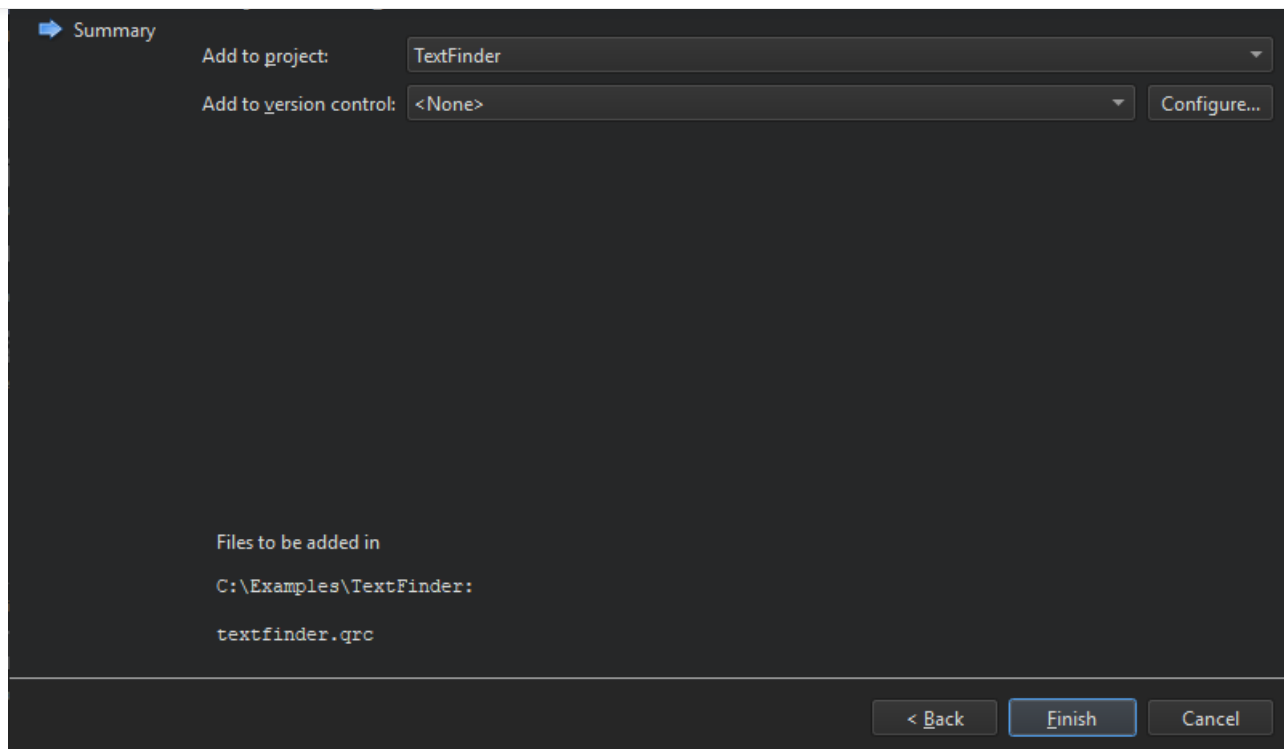


此时将打开“**选择位置**”对话框。

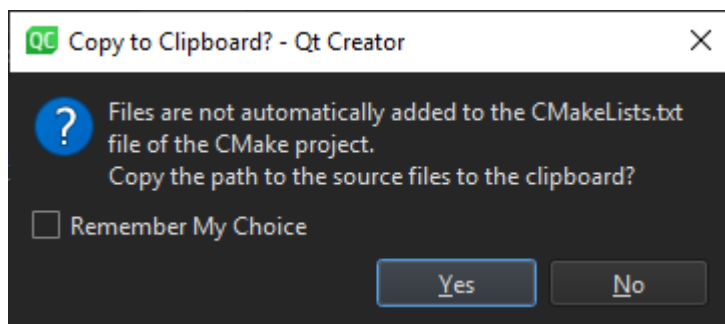


2. 在“名称”字段中，输入**文本查找器**。
3. 在“路径”字段中，输入项目的**路径**，然后选择“**下一步**”或“**继续**”。

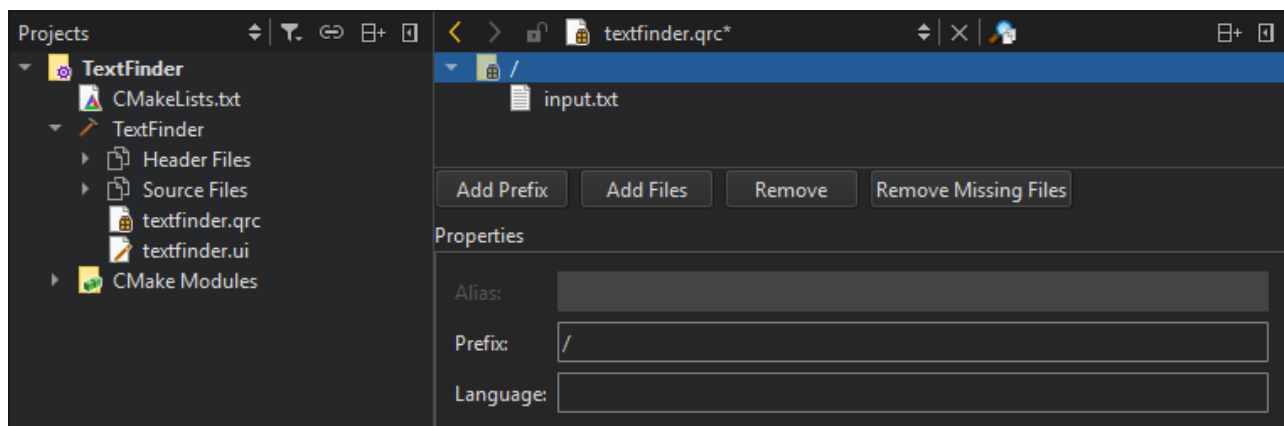
将打开“**项目管理**”对话框。



4. 在“添加到项目”字段中，选择“文本查找器”，然后选择“完成”或“完成”以在代码编辑器中打开文件。
5. 在“复制到剪贴板”对话框中，选择“是”将资源文件的路径复制到剪贴板，以将其添加到 CMakeLists.txt 文件。



6. 选择添加>添加前缀。
7. 在前缀字段中，将默认前缀替换为斜杠 (/)。
8. 选择“添加>添加文件”以查找并添加输入.txt。



```
set(PROJECT_SOURCES
    main.cpp
    textfinder.cpp
    textfinder.h
    textfinder.ui
    ${TS_FILES}
    textfinder.qrc
)
```

编译和运行应用程序

现在，你已拥有所有必需的文件，请选择用于▶编译和运行应用程序的按钮。

< 创建Qt快速应用程序

创建移动应用程序 >

©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。



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



社区

为Qt做贡献

论坛

维基

下载

市场

© 2022 Qt公司

[反馈](#) [登录](#)