搜索

在 Qt for Python 应用程序中使用设计器 UI 文件

将表单转换为 Python 代码

为了演示，我们使用Qt小部件动画缓动示例。

该应用程序由一个源文件、一个 UI 文件、一个资源文件和项目文件组成，该文件采用 YAML 格式：`easing.pyform.uieasing.qrceasing.pyproject`

```
{
}
```

Topics >

UI 文件将转换为使用用户界面编译器 （uic） 构建表单的 Python 代码：

```
uic -g python form.ui > ui_form.py
```

由于顶级小部件已命名，因此生成了一个名为 Python 类。它提供了一个函数，将小部件作为参数，调用该函数来创建 UI 元素：`FormUi_FormsetupUi()`

```
from ui_form import Ui_Form
...
class Window(QtWidgets.QWidget):
    def __init__(self, parent=None):
        super(Window, self).__init__(parent)

        self.m_ui = Ui_Form()
        self.m_ui.setupUi(self)
```

稍后，可以通过类访问小部件：`Ui_Form`

Qt DOCUMENTATION

此外，还提供了另一种方法，该方法可以调用以响应QEvent类型的QEvent。语言更改，指示应用程序语言的更改。setupUi()Ui_FormretranslateUi()

UiTools 方法

类提供了一个表单加载器对象，用于在运行时构造用户界面。此用户界面可以从任何QIODevice 检索，例如QFile对象。QUiLoader：：load（） 函数使用文件中包含的用户界面描述构造表单小部件。

它由 uiloader 示例演示：

```
from PySide2.QtUiTools import QUiLoader

if __name__ == '__main__':
    # Some code to obtain the form file name, ui_file_name
    app = QApplication(sys.argv)
    ui_file = QFile(ui_file_name)
    if not ui_file.open(QIODevice.ReadOnly):
        print("Cannot open {}: {}".format(ui_file_name, ui_file.errorString()))
        sys.exit(-1)
    loader = QUiLoader()
    widget = loader.load(ui_file, None)
    ui_file.close()
    if not widget:
        print(loader.errorString())
        sys.exit(-1)
    widget.show()
    sys.exit(app.exec_())
```

‹ 在C++应用程序中使用设计器 UI 文件

在Qt Designer中使用自定义小部件 ›

职业
办公地点

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

反馈 登录