搜索

添加Qt设计器插件

您可以使用Qt API创建扩展Qt应用程序的插件。这使您能够将自己的小部件添加到Qt Designer中。将插件包含在应用程序中的最灵活方法是将其编译为单独提供的动态库，并在运行时检测和加载。

应用程序可以检测存储在标准插件子目录中的插件。有关如何创建和定位插件以及更改默认插件路径的更多信息，请参阅如何创建Qt插件。

有关如何为 Qt Designer 创建插件的更多信息，请参阅在 Qt 设计器中使用自定义控件。

定位Qt设计器插件

Qt创建者手册8.0.2
Topics >

集成的Qt Designer从Windows和Linux上的Qt Creator安装目录中获取插件。有关如何在 macOS 上配置插件的信息，请参阅在 macOS 上配置 Qt 设计器插件。`\bin\plugins\designer`

要检查哪些插件成功加载，哪些失败，请选择**工具>表单编辑器>关于Qt Designer插件。**

独立的Qt Designer是用于构建项目的Qt库的一部分，位于Qt安装目录中。它从子目录中获取插件。要检查哪些插件已成功加载，哪些插件失败，请选择**帮助>关于插件。**`<Qt_version>\<compiler>\bin\plugins\designerbin`

在 macOS 上配置 Qt 设计器插件

在 macOS 上，必须从捆绑包构建和运行 GUI 应用程序。捆绑包是一种目录结构，在"访达"中查看时显示为单个实体。应用程序的捆绑包通常包含可执行文件及其所需的所有资源。

Qt Creator使用位于捆绑包中的一组Qt库，因此，您需要配置要与Qt Creator一起使用的Qt Designer插件。有关如何将应用程序部署到 macOS 的更多信息，请参阅Qt for macOS - 部署。

以下示例说明了如何配置Qwt - Qt Widgets for Technical Applications库的5.2.1版本以与Qt Creator一起使用：

1. 要检查 Qwt 库中使用的路径，请输入以下命令：`otool`

```
otool -L /Developer/Applications/Qt/plugins/designer/libqwt_designer_plugin.dylib
```

Qwt 5.2.1 的输出表明该插件使用 Qt 核心库（QtDesigner、QtScript、QtXml、QtGui和 QtCore）和 libqwt.5.dylib：

```
/Developer/Applications/Qt/plugins/designer/libqwt_designer_plugin.dylib:
        libqwt_designer_plugin.dylib (compatibility version 0.0.0, current version 0.0.0)
        libqwt.5.dylib (compatibility version 5.2.0, current version 5.2.1)
        QtDesigner.framework/Versions/4/QtDesigner (compatibility version 4.6.0, current version 4.6.2)
        QtScript.framework/Versions/4/QtScript (compatibility version 4.6.0, current version 4.6.2)
        QtXml.framework/Versions/4/QtXml (compatibility version 4.6.0, current version 4.6.2)
        QtGui.framework/Versions/4/QtGui (compatibility version 4.6.0, current version 4.6.2)
        QtCore.framework/Versions/4/QtCore (compatibility version 4.6.0, current version 4.6.2)
        /usr/lib/libstdc++.6.dylib (compatibility version 7.0.0, current version 7.9.0)
```

2. 您必须将 Qt 设计器插件和 Qwt 库文件复制到以下位置：
   - `libqwt_designer_plugin.dylib`自`Qt Creator.app/Contents/PlugIns/designer`
   - `libqwt.*.dylib`自`Qt Creator.app/Contents/Frameworks`

   输入以下命令：

```
sudo cp /Developer/Applications/Qt/plugins/designer/libqwt_designer_plugin.dylib \
        /Developer/Applications/Qt/Qt\ Creator.app/Contents/MacOS/designer
sudo cp -R /usr/local/qwt-5.2.1/lib/*  \
        /Developer/Applications/Qt/Qt\ Creator.app/Contents/Frameworks/
```

3. 输入以下命令以检查 Qwt 库使用的库：`otool`

```
otool -L /usr/local/qwt-5.2.1/lib/libqwt.5.dylib
```

   该命令返回以下输出：

```
/usr/local/qwt-5.2.1/lib/libqwt.5.dylib:
        libqwt.5.dylib (compatibility version 5.2.0, current version 5.2.1)
        QtGui.framework/Versions/4/QtGui (compatibility version 4.6.0, current version 4.6.2)
        QtCore.framework/Versions/4/QtCore (compatibility version 4.6.0, current version 4.6.2)
        /usr/lib/libstdc++.6.dylib (compatibility version 7.0.0, current version 7.9.0)
        /usr/lib/libgcc_s.1.dylib (compatibility version 1.0.0, current version 438.0.0)
        /usr/lib/libSystem.B.dylib (compatibility version 1.0.0, current version 125.0.1)
```

4. 输入以下命令以修复库的引用：`install_name_tool`

```
cd /Developer/Applications/Qt/Qt\ Creator.app/Contents/MacOS/designer
sudo install_name_tool -change
    QtCore.framework/Versions/4/QtCore \
    @executable_path/../Frameworks/libQtCore.4.dylib \
    libqwt_designer_plugin.dylib
sudo install_name_tool -change QtGui.framework/Versions/4/QtGui \
    @executable_path/../Frameworks/libQtGui.4.dylib \
    libqwt_designer_plugin.dylib
sudo install_name_tool -change QtXml.framework/Versions/4/QtXml \
    @executable_path/../Frameworks/libQtXml.4.dylib \
    libqwt_designer_plugin.dylib
sudo install_name_tool -change QtScript.framework/Versions/4/QtScript \
    @executable_path/../Frameworks/libQtScript.4.dylib \
    libqwt_designer_plugin.dylib
sudo install_name_tool -change QtDesigner.framework/Versions/4/QtDesigner \
    @executable_path/../Frameworks/libQtDesigner.4.dylib \
    libqwt_designer_plugin.dylib
sudo install_name_tool -change libqwt.5.dylib \
    @executable_path/../Frameworks/libqwt.5.dylib \
    libqwt_designer_plugin.dylib

cd /Developer/Applications/Qt/Qt\ Creator.app/Contents/Frameworks
sudo install_name_tool -change \
    QtCore.framework/Versions/4/QtCore \
    @executable_path/../Frameworks/libQtCore.4.dylib \
    libqwt.5.2.1.dylib
sudo install_name_tool -change \
```

```
libqwt.5.2.1.dylib
```

匹配构建密钥

Windows上预构建的Qt包中包含的Qt Creator是使用Microsoft Visual Studio编译器构建的，而用于构建应用程序的Qt版本则配置为使用MinGW / g ++编译器。 Qt Creator无法加载使用此版本的Qt构建的插件，因为构建键不匹配。这些插件只能在独立版本的Qt Designer中使用。选择**帮助**>**关于Qt**Creator以检查Qt Creator的Qt版本。

要使用为随附的Qt版本构建的Qt Designer插件，请确保Qt Creator是使用相同的编译器构建的，方法是使用MinGW重新编译Qt Creator或使用Microsoft Visual Studio重新编译Qt，具体取决于您要用于应用程序的配置。

‹ 开发基于小部件的应用程序 优化移动设备的应用程序 ›