

 搜索Qt 6.4 > Qt语言学家手册 > [Qt语言学家手册：开发人员](#)

Qt语言学家手册：开发人员

在Qt应用程序中，对多种语言的支持非常简单，并且不会给开发人员的工作量增加开销。

Qt通过在创建每个窗口的短语时翻译短语，最大限度地降低了使用翻译的性能成本。在大多数应用程序中，主窗口只创建一次。对话框通常创建一次，然后根据需要显示和隐藏。初始转换发生后，已翻译的窗口不会再产生运行时开销。只有那些创建、销毁和随后创建的窗口才会有翻译性能成本。

使用Qt可以创建可以在运行时切换语言的应用程序，但需要一定量的开发人员干预，当然会产生一些运行时性能成本。

在Qt源文件中指定翻译源

Topics >

地语言的每种语言的条目。典型的条目如下所示：.proTRANSLATIONS

```
TRANSLATIONS = arrowpad_fr.ts \  
              arrowpad_nl.ts
```

在翻译文件名中使用区域设置对于确定在运行时加载哪种语言非常有用。有关更多信息，请参阅[QLocale](#)。

该工具从应用程序中提取用户界面字符串。它读取应用程序.pro 文件以识别哪些源文件包含要翻译的文本。这意味着源文件必须列在.pro 文件的理论条目中，或列在条目中列出的资源文件中。如果未列出您的文件，则找不到其中的文本。lupdateSOURCESHEADERSRESOURCE

包含四个翻译源文件的完整文件示例：.pro

```
HEADERS      = main-dlg.h \  
              options-dlg.h  
SOURCES      = main-dlg.cpp \  
              options-dlg.cpp \  
              main.cpp  
FORMS        = search-dlg.ui  
TRANSLATIONS = superapp_dk.ts \  
              superapp_fi.ts \  
              superapp_no.ts \  
              superapp_se.ts
```

lupdate默认情况下，期望所有源代码都以 UTF-8 编码。具有 BOM（字节顺序标记）的文件也可以用 UTF-16 或 UTF-32 编码。将 qmake 变量设置为 UTF-16 解析没有 BOM 的文件。CODECFORSRCUTF-16

但是，某些编辑器（如 Visual Studio）默认使用不同的编码。避免编码问题的一种方法是将任何源代码限制为 ASCII，并对带有其他字符的可翻译字符串使用转义序列，例如：

```
label->setText(tr("F\374r \310lse"));
```

国际化应用程序

设计应用程序，使其无需工程更改即可适应各种语言和区域。Qt 试图让你的国际化尽可能轻松。Qt 中的所有输入控件和文本绘制方法都为所有支持的语言提供内置支持。但是，在为应用程序编写源代码时，仍需要牢记以下事项：

- › 使应用程序查找并加载相应的翻译文件。
- › 将用户可见的文本和 Ctrl 键盘快捷键标记为翻译目标。
- › 为要翻译的文本提供上下文。
- › 消除相同文本的歧义。
- › 使用编号参数（%n）作为在运行时替换为文本或数字的参数的占位符。
- › 国际化数字、日期、时间和货币。
- › 在函数外部标记可翻译的数据文本字符串。

您可以在同一应用程序中开发同时使用 C++ 和 QML 源的应用程序，甚至可以在两个源中都有用户界面字符串。这些工具创建一个组合翻译文件，字符串可从 C++ 和 QML 访问。

支持 Qt 应用程序国际化的类在 [Qt 的国际化](#) 中进行了描述。使源代码可翻译的过程在 [编写翻译源代码](#) 和使用 [Qt Quick 的国际化和本地化](#) 中进行了描述。

每一段需要翻译的文本都需要上下文来帮助翻译人员识别文本在程序中出现的位置。在多个相同的文本需要不同翻译的情况下，翻译人员还需要一些信息来消除源文本的歧义。标记要翻译的文本将自动使类名用作基本上下文信息。在某些情况下，开发人员可能需要添加其他信息来帮助翻译人员。

部署翻译

应用程序所需的文件应放置在使用 [QTranslator](#) 的加载程序代码可以找到它们的位置。通常，这是通过指定相对于 [QCoreApplication::applicationDirPath\(\)](#) 的路径来完成的。[.qm](#)

通常，应用程序有文件，如果使用的 Qt 版本未安装在系统上，则还需要部署 Qt 的 [.qm](#) 文件。[.qm](#)

在 Qt 4 中，每个语言环境都有一个大的单片文件。例如，该文件包含所有库的德语翻译。[.qmqt_de.qm](#)

在 Qt 5 中，文件被按模块拆分，有一个所谓的元目录文件，其中包括所有模块的文件。元目录文件的名称与 Qt 4 的整体文件的名称相同，因此只要找到所有包含的文件，现有的加载器代码就可以像以前一样工作。[.qm.qm.qm.qm](#)

但是，没有必要总是部署 Qt 5 的所有文件。我们建议使用部署步骤中的工具将所需模块的文件连接到一个与元目录文件名匹配的文件。例如，要使用模块 [Qt Core](#)、Qt GUI 和 [Qt Quick](#) 为应用程序创建德语翻译文件，请运行：[.qm.qmlconvert](#)

教程

以下教程说明了如何准备Qt应用程序进行翻译：

- › **Hello tr ()** 是一个C++应用程序，它演示了QTranslator对象的创建。它还显示了该函数的最简单用法，以标记用户可见的源文本以进行翻译。tr()
- › **箭头板**是一个C++应用程序，它演示如何使用应用程序加载适用于当前区域设置的翻译文件。它还显示了双参数形式的使用，其中为翻译人员提供了额外的信息。tr()
- › **Troll Print**是一个C++应用程序，它演示了如何区分相同的源文本，即使它们出现在相同的上下文中。本教程还讨论了翻译工具如何在升级应用程序时帮助最大程度地减少翻译人员的工作。
- › **国际化**是一个Qt Quick应用程序，它演示了如何国际化Qt快速应用程序。

◀ Qt语言学家手册：译者

Qt语言学家手册：TS文件格式 ▶

©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/或其他国家/地区的商标 全球。所有其他商标均为其各自所有者的财产。



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

为Qt做贡献
论坛
维基
下载
市场

© 2022 Qt公司

[反馈](#) [登录](#)