

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，并且对开发人员的工作负载几乎没有增加开销。

Qt通过在创建每个窗口时翻译短语，最大限度地降低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。一旦进行了初始转换，已翻译的窗口就不会有进一步的运行时开销。只有那些被创建、销毁和随后创建的窗口才会产生翻译性能成本。

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

在Qt项目文件中指定翻译源

要使发布管理器能够使用 `lupdate` 和 `lrelease`，请指定 Qt 项目文件。对于本地语言之外的每种语言，项目文件的部分中都必须有一个条目。典型的条目如下所示：`.pro`TRANSLATIONS

```
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。有关详细信息，请参阅 [Q 本地](#)。

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，以确定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 `或` 条目中列出的资源文件中。如果未列出您的文件，则将找不到其中的文本。`lupdateSOURCESHEADERSRESOURCE`

包含四个翻译源文件的完整文件的示例：`.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
```

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

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

国际化应用程序

设计应用程序，使其无需更改工程即可适应各种语言和区域。Qt 试图让你的国际化尽可能无痛。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，您仍然需要牢记以下事项：

- › 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）作为在运行时用文本或数字替换的参数的占位符。
- › 国际化数字、日期、时间和货币。
- › 在函数外部标记数据文本字符串可翻译。

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

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

需要翻译的每段文本都需要上下文来帮助翻译人员识别文本在程序中的显示位置。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导致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](#) 的加载器代码可以找到它们的位置。通常，这是通过指定相对于 [QCoreApplication](#) 的路径来完成的：`applicationDirPath()`。 .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

In Qt 4, there is one big, monolithic file per locale. For example, the file contains the German translation of all libraries. `qmqt_de.qm`

In Qt 5, the files were split up by module and there is a so-called meta catalog file which includes the files of all modules. The name of the meta catalog file is identical to the name of Qt 4's monolithic file so that existing loader code works as before provided all included files are found. `.qm.qm.qm.qm`

However, it is not necessary to always deploy all of Qt 5's files. We recommend concatenating the files of the modules required to one file matching the meta catalog file name using the tool in the deploy step. For example, to create a German translation file for an application using the modules [Qt Core](#), [Qt GUI](#), and [Qt Quick](#), run: `.qm.qmlconvert`

Tutorials

The following tutorials illustrate how to prepare Qt applications for translation:

- › **Hello tr()** is a C++ application that demonstrates the creation of a **QTranslator** object. It also shows the simplest use of the function to mark user-visible source text for translation. `tr ()`
- › **Arrow Pad** is a C++ application that demonstrates how to make the application load the translation file applicable to the current locale. It also shows the use of the two-argument form of which provides additional information to the translator. `tr ()`
- › **Troll Print** is a C++ application that demonstrates how identical source texts can be distinguished even when they occur in the same context. This tutorial also discusses how the translation tools help minimize the translator's work when an application is upgraded.
- › **Internationalization** is a Qt Quick application that demonstrates how to internationalize Qt Quick applications.

[< Qt Linguist Manual: Translators](#)

[Qt Linguist Manual: TS File Format >](#)

© 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

Support

Support Services
Professional Services
Partners
Training

Licensing

Terms & Conditions
Open Source
FAQ

For Customers

Support Center
Downloads
Qt Login
Contact Us



Community

- Contribute to Qt
- Forum
- Wiki
- Downloads
- Marketplace

© 2022 The Qt Company

[Feedback](#) [Sign In](#)