

Qt Linguist Manual: Release Manager

Two tools are provided for the release manager: `lupdate` and `lrelease`. These tools can process `qmake` project files, or operate directly on the file system.

Creating Translation Files

Translation files consist of all the user-visible text and Ctrl key accelerators in an application and translations of that text.

To create translation files:

1. Run `lupdate` to generate the first set of translation source (TS) files with all the user-visible text but no translations.
2. Give the TS files to the translator who adds translations using *Qt Linguist*. *Qt Linguist* takes care of any changed or deleted source text.
3. Run `lupdate` to incorporate any new text added to the application. `lupdate` synchronizes the user-visible text from the application with the translations. It does not destroy any data.
4. To release the application, run `lrelease` to read the TS files and produce the QM files used by the application at runtime.

For `lupdate` to work successfully, it must know which translation files to produce. The files are listed in the application's `.pro` Qt project file.

Using `lupdate`

Usage: `lupdate myproject.pro`

The `lupdate` command line tool finds the translatable strings in the specified source, header and *Qt Designer* interface files, and produces or updates `.ts` translation files. The files to process and the files to update can be set at the command line, or provided in a `.pro` file specified as a command line argument. The developer creates the `.pro` file, as described in [Qt Linguist Manual: Developers](#).

You can generate the translation file for a single QML file in the following way:

```
lupdate main.qml -ts main_en.ts
```

To make a translation file for another language, for example French, you copy `main_en.ts` to `main_fr.ts`, and

```
RESOURCES += qml.qrc
```

To have all QML files processed by `lupdate`:

```
lupdate application.qrc -ts myapp_en.ts
```

You can also process all QML files without using a `.qrc` file:

```
lupdate -extensions qml -ts myapp_en.ts
```

If you are mostly processing QML files, but also have C++ code that contains strings to be translated, you can add it in the same command:

```
lupdate qml.qrc filevalidator.cpp -ts myapp_en.ts
```

The translation files can be mentioned in the `.pro` file, but it is also possible to do without it, and just specify the translation file on the command line.

For example, to generate `.ts` files that will be used for English and French:

```
lupdate qml.qrc filevalidator.cpp -ts myapp_en.ts myapp_fr.ts
```

The produced translation files are given to the translator who uses *Qt Linguist* to read the files and insert the translations.

Companies that have their own translators in-house may find it useful to run `lupdate` regularly, perhaps monthly, as the application develops. This will lead to a fairly low volume of translation work spread evenly over the life of the project and will allow the translators to support a number of projects simultaneously.

Companies that hire in translators as required may prefer to run `lupdate` only a few times during the application life cycle. The first time might be just before the first test phase. This will provide the translator with a substantial single block of work and any bugs that the translator detects may easily be included with those found during the initial test phase. The second and any subsequent `lupdate` runs would probably take place during the final beta phase.

The TS file format is a simple human-readable XML format that can be used with version control systems if required. `lupdate` can also process Localization Interchange File Format (XLIFF) format files; files in this format typically have file names that end with the `.xlf` suffix.

Note: The minimum supported version for XLIFF format files is 1.1. XLIFF 1.0 version files are not supported.

Qt Linguist is also able to import and export XLIFF files. See the [Translators](#) section for more information.

Using lrelease

Usage: `lrelease myproject.pro`

The `lrelease` command line tool produces QM files out of TS files. The QM file format is a compact binary format that is used by the localized application. It provides extremely fast lookups for translations. The TS files `lrelease` processes can be specified at the command line, or given indirectly by a Qt `.pro` project file.

This tool is run whenever a release of the application is to be made, from initial test version through to final release version. If the QM files are not created, for example because an alpha release is required before any translation has been undertaken, the application will run perfectly well using the text the developers placed in the source files. Once the QM files are available the application will detect them and use them automatically.

`lrelease` can be also be run without specifying a `.pro` file:

```
lrelease.exe main_en.ts languages\main_fr.ts
```

Note: The `lrelease` tool only incorporates translations that are marked as "finished". Otherwise the original text is used instead.

Pass the `-help` option to `lrelease` to view supported options.

Missing Translations

Both `lupdate` and `lrelease` may be used with TS translation source files which are incomplete. Missing translations will be replaced with the native language phrases at runtime.

[< Overview of the Translation Process](#)

[Qt Linguist Manual: Translators >](#)

© 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



Investors
Newsroom
Careers
Office Locations

Open Source
FAQ

Support

Support Services
Professional Services
Partners
Training

For Customers

Support Center
Downloads
Qt Login
Contact Us
Customer Success

Community

Contribute to Qt
Forum
Wiki
Downloads
Marketplace