

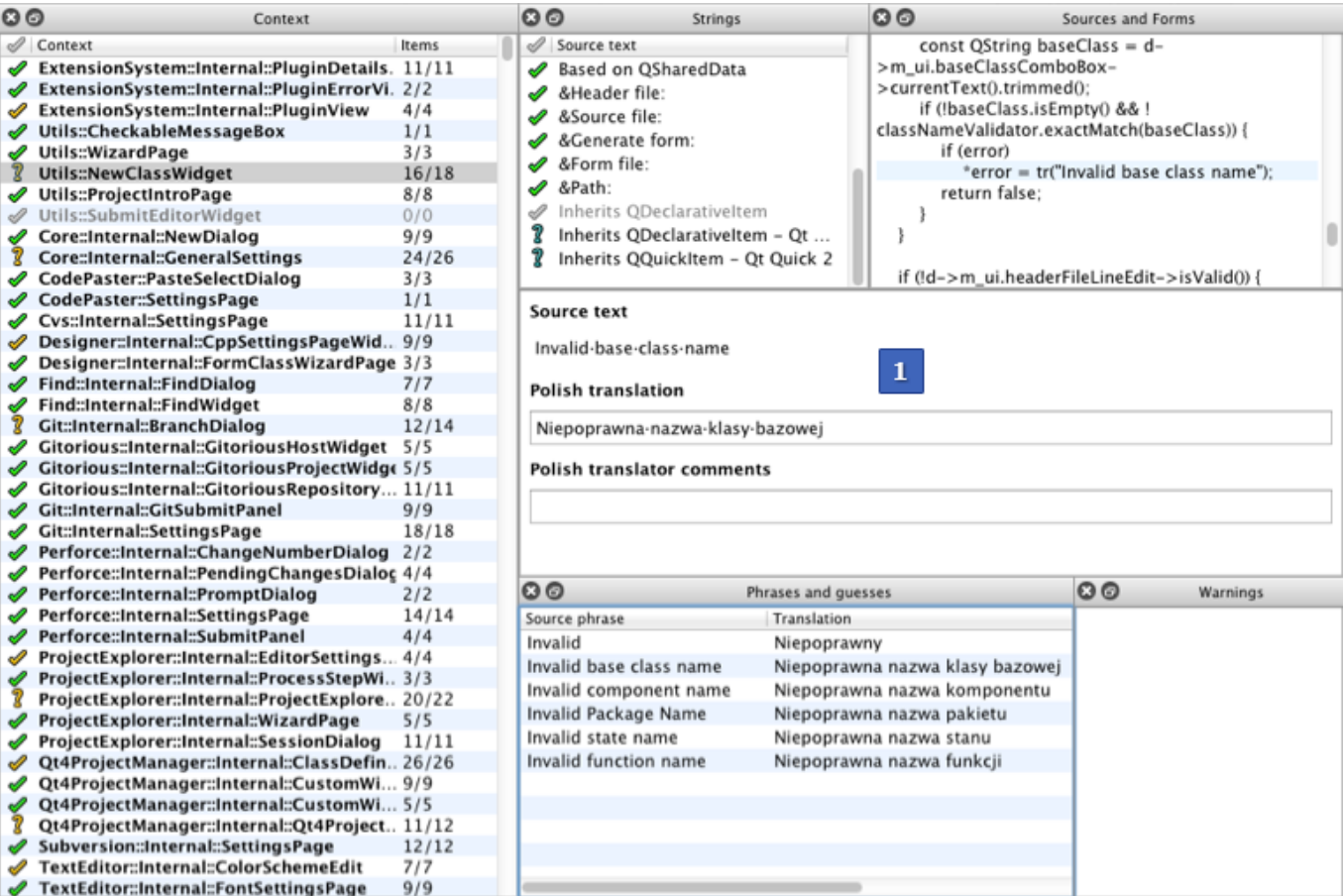
Qt 6.4 > Qt Linguist Manual > [Qt Linguist Manual: Translators](#)

Qt Linguist Manual: Translators

Qt Linguist is a tool for adding translations to Qt applications. Once you have installed Qt, you can start *Qt Linguist* in the same way as any other application on the development host.

The *Qt Linguist* main window contains a menu bar and the following views:

- **Context (F6)** for selecting from a list of contexts in which strings to be translated appear.
- **Strings (F7)** for viewing translatable strings found in a context.
- **Sources and Forms (F9)** for viewing the context where the current string is used if the source code for the context is accessible.
- Translation area for translating strings.
- **Phrases and Guesses (F10)** for viewing possible translations for the current string.
- **Warnings (F8)** for viewing translated strings that fail validation tests.



Translating Strings

You open translation source (TS) files in *Qt Linguist* for translation. TS files are human-readable XML files containing source phrases and their translations. TS files are usually created and updated by *lupdate*. If you do not have a TS file, see [Release Manager](#) to learn how to generate one.

You can use *Qt Linguist* also to translate files in the international XML Localization Interchange File Format (XLIFF) that are generated by other programs. However, for standard Qt projects, only the TS file format is used. The minimum supported version for XLIFF format files is 1.1.


Qt Linguist displays the target language in the translation area, and adapts the number of input fields for plural forms accordingly. When you open several TS files to translate simultaneously, the **Translator** and **Translator comment** fields are displayed for each language. For more information about setting the location information, see [Changing the Target Locale](#).



If the developer provides a [disambiguating comment](#), it is displayed in the **Developer comments** field.

To translate strings:

1. Select **File > Open** to load a TS file.
2. Select a context in the **Context** view to load the translatable strings found in that context into the **Strings** view.
3. Select a string to copy it as the **Source text** in the translation area. The whitespace within the source text is visualized so that you can see the spacing required within the text.
4. Enter the translation of the current string in the **Translation** field.

You can also select an existing translation from the **Phrases and Guesses** view by double-clicking it. The phrases are read from phrase books and the guesses are existing translations of similar phrases in the TS file.

5. Optionally, enter a comment to be read by other translators in the **Translator comment** field.
6. To accept the translation, press **Ctrl+Enter**, select , or click the icon to the left of the selected source string in the string list.
7. Select **File > Save** to save your work.

Repeat this process until all strings in the string list are marked with  (**Accepted/Correct**) or  (**Accepted/Warnings**). Then select the next context and continue.

To view the number of words and characters in the source text and in the translated text, select **View > Statistics**.

Select **File > Release** to create a Qt message QM file with the same base name as the current translation source file. The release manager's command line tool *lrelease* performs the same function on *all* of an application's translation source files.

To print the translation source and the translations, select **File > Print**.

To quit *Qt Linguist*, select **File > Exit**.

Leaving a Translation for Later

If you wish to leave a translation, press **Ctrl+L** (Next Unfinished) to move to the next unfinished translation. To move to the next translation (whether finished or unfinished) press **Shift+Ctrl+L**. You can also navigate using the **Translation** menu. If you want to go to a different context entirely, click the context you want to work on in the **Context** list, then click the source text in the **Strings** view.

The same phrase may occur in two or more contexts without conflict. Once a phrase has been translated in one context, *Qt Linguist* notes that the translation has been made and when the translator reaches a later occurrence of the same phrase, *Qt Linguist* provides the previous translation as a possible translation candidate in the **Phrases and Guesses** view.

If a phrase occurs more than once in a particular context it is shown only once in the **Context** view and the translation is applied to every occurrence within the context. If the same phrase needs to be translated differently within the same context, the developer must provide a distinguishing comment for each occurrence of the phrase. If such comments are used, the duplicate phrases appear in the **Context** view. The developer's comments appear in the translation area on a light blue background.

Changing Keyboard Accelerators

A keyboard accelerator is a key combination that, when pressed, causes an application to perform an action. There are two kinds of keyboard accelerators: Alt key and Ctrl key accelerators.

Alt Key Accelerators

Alt key accelerators are used in menu selection and on buttons. The underlined character in a menu item or button label signifies that pressing the Alt key with the underlined character will perform the same action as clicking the menu item or pressing the button. For example, most applications have a *File* menu with the "F" in the word "File" underlined. In these applications the *File* menu can be invoked either by clicking the word "File" on the menu bar or by pressing *Alt+F*. To identify an accelerator key in the translation text ("File") precede it with an ampersand: *&File*. If a string to be translated has an ampersand in it, then the translation for that string should also have an ampersand in it, preferably in front of the same character.

The meaning of an Alt key accelerator can be determined from the phrase in which the ampersand is embedded. The translator can change the character part of the Alt key accelerator, if the translated phrase does not contain the same character or if that character has already been used in the translation of some other Alt key accelerator. Conflicts with other Alt key accelerators must be avoided within a context. Note that some Alt key accelerators, usually those on the menu bar, may apply in other contexts.

Ctrl Key Accelerators

Ctrl key accelerators can exist independently of any visual control. They are often used to invoke actions in menus that would otherwise require multiple keystrokes or mouse clicks. They may also be used to perform actions that do not appear in any menu or on any button. For example, most applications that have a *File* menu have a *New* submenu item in the *File* menu. The *New* item might appear as "New Ctrl+N" in the *File* menu, meaning the *New* menu can be invoked by simply pressing **Ctrl+N**, instead of either clicking *File* with the mouse and then clicking *New* with the mouse, or by entering *Alt+F* and *N*.

Each Ctrl key accelerator is shown in the **Strings** view as a separate string, for example **Ctrl+Enter**. Since the string does not have a context to give it meaning, such as the context of the phrase in which an Alt key accelerator appears, the translator must rely on the UI developer to include a **disambiguation comment** to explain the action the Ctrl key accelerator is meant to perform. This disambiguating comment (if provided by the developer) will appear under **Developer comments** in the translation area below the **Source text** field.

Ideally, translations for Ctrl key accelerators are simply copied by selecting **Translation > Copy from source text**. However, in some cases the character will not make sense in the target language, and it must be changed. Whichever character (alpha or digit) is chosen, the translation must be in the form "Ctrl+" followed by the upper case character. Qt will automatically display the correct name at run-time. As with Alt key accelerators, if the translator changes the character, the new character must not conflict with any other Ctrl key accelerator.

Warning: Do not translate the "Alt", "Ctrl" or "Shift" parts of the accelerators. Qt relies on these strings being there. For supported languages, Qt automatically translates these strings.

Handling Numbered Arguments and Plurals

Some phrases contain numbered arguments. A numbered argument is a placeholder that will be replaced with text at run-time. A numbered argument appears in a source string as a percent sign followed by a digit. Consider an example: `After processing file %1, file %2 is next in line.` In this string to be translated, `%1` and `%2` are numbered arguments. At run-time, `%1` and `%2` will be replaced with the first and second file names respectively. The same numbered arguments must appear in the translation, but not necessarily in the same order. A German translation of the string might reverse the phrases, for example `Datei %2 wird bearbeitet, wenn Datei %1 fertig ist.` Both numbered arguments appear in the translation, but in the reverse order. `%i` will always be replaced by the same text in the translation strings, regardless of where argument *i* appears in the argument sequence in the source string.

The use of numbered arguments is often accompanied by the use of plurals in the source text. In many languages, the form of the text will depend on the value shown, and more than one translation is required. If the developers have marked up the source text in correct way, fields for each of the possible plural forms will be available in the translation area. For more information, see [Writing Source Code for Translation](#).

Changing the Target Locale

You can set the locale information explicitly in **Edit > Translation File Settings**. If the target language and country are not explicitly set when you open a translation source file, *Qt Linguist* attempts to deduct them from the translation source file name. This requires that the translation files adhere to the following file name convention: `appname_language[_country].ts`, where:

- › `language` is an ISO 639 language code in lowercase.
- › `country` is an ISO 3166 two-letter country code in uppercase.

If this attempt to resolve the target language and country fails, the **Translation File Settings** window opens.

For example, `app_de.ts` sets the target language to German, and `app_de_ch.ts` sets the target language to German and the target country to Switzerland. This also helps loading translations for the current locale automatically. For more information, see [Developers](#).



Selecting Context to Translate

The **Context** view lists the contexts in which strings to be translated appear. The column labeled **Context** lists the context names in alphabetical order. Each context is the name of a subclass of `QObject`. There can also be a context for `QObject` itself, which contains strings passed to the static function `QObject::tr()`. There can also be an `<unnamed context>`, which contains strings that are not in a subclass of `QObject`.

The following icons indicate the current translation state for each context:

		validation tests.
Accepted/Warnings	✓	All strings in the context have been translated or marked as translated, but at least one translation failed the validation tests. In the Strings view, you can see which string failed the test.
Not Accepted	?	At least one string in the context has not been translated or is not marked as translated.
Obsolete	✓	None of the translated strings appears in the context any more. This usually means the context itself no longer exists in the application.

The **Items** column displays the total number of translatable strings in the context and the number of translated strings, separated by a slash (/). If the numbers are equal, all the translatable strings in the context have translations.

Selecting String to Translate

The **Strings** view lists all the translatable strings found in the current context and their translation acceptance state. Selecting a string makes that string the current string in the translation area.

Click the icon in front of a string to change its translation acceptance state. A tick mark, green or yellow, means the string has been translated and the translation has been accepted. A question mark means either that the translation has not been accepted or that the string does not have a translation.

The following icons indicate the current translation state for each string:

State	Icon	Description
Accepted/Correct	✓	The source string has a translation (possibly empty). The user has accepted the translation, and the translation passes all the validation tests . If the translation is empty, the user has chosen to leave it empty. Click the icon to revoke acceptance of the translation and decrement the number of accepted translations in the Items column of the gui Context view by 1. The state is reset to Not Accepted if the string has a translation, or to No Translation if the string's translation is empty. If Update changes the contents of a string, its acceptance state is automatically reset to Not Accepted .
Accepted/Warnings	✓	The user has accepted the translation, but the translation does not pass all the validation tests. The validation test failures are shown in the Warnings view. Click the icon to revoke acceptance of the translation. The state is reset to Validation Failures , and the number of accepted translations in the Items column of the Context view is decremented by 1.
Not Accepted	?	The string has a translation that passes all the validation tests, but the user has not yet accepted the translation. Click the icon or press Ctrl+Enter to accept the translation. The state is reset to Accepted/Correct , and the number of accepted translations in the Items column of the Context view is incremented by 1.
No Translation	?	The string does not have a translation. Click the icon to accept the empty translation anyway. The state is reset to Accepted/Correct , and the number of accepted translations in the Items column of the Context view is incremented by 1.
Validation Failures	!	The string has a translation, but the translation does not pass all the validation tests. Validation test failures are shown in the Warnings view. Click on the icon or press Ctrl+Return to accept the translation even with validation failures. The state is reset to Accepted/Warnings . We recommend editing the translation to fix the causes of
State	Icon	Description

Obsolete



The string is obsolete. It is no longer used in the context. See the [Release Manager](#) for instructions on how to remove obsolete messages from the file.

Viewing Strings in Context

If the source files containing the translatable strings are available to *Qt Linguist*, the **Sources and Forms** view shows the source context of the current string in the **Strings** view. The source code line containing the current string should be shown and highlighted. If the file containing the source string is not found, the expected absolute file path is shown.

If the source context shows the wrong source line, it probably means the translation file is out of sync with the source files. For more information about how to re-sync the translation file with the source files, see [Using lupdate](#).

Forms created by *Qt Designer* are stored in special UI files. *Qt Linguist* can make use of these UI files to show the translations done so far on the form itself if the UI files are available to it during the translation process.

Reusing Translations

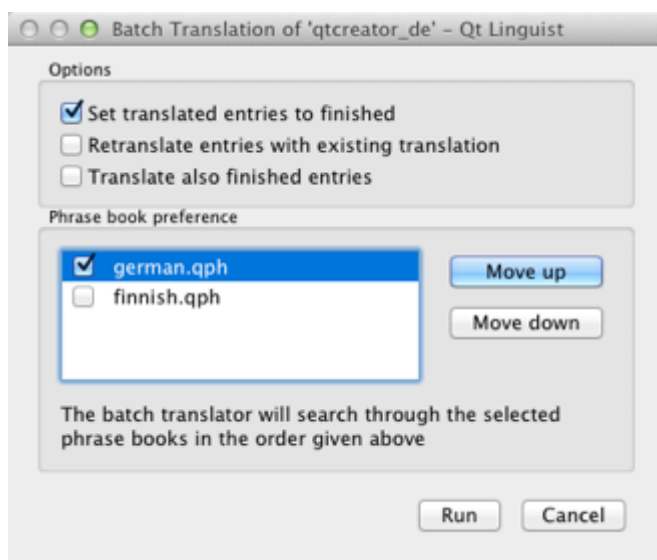
If the translated text is similar to the source text, select **Translation > Copy from source text** (or press **Ctrl+B**) to copy the source text into the translation area.

In *Qt Linguist*, *phrase books* are used to provide a common set of translations to help ensure consistency. A phrase book is a set of source phrases, target (translated) phrases, and optional definitions. Typically, one phrase book is created per language and family of applications. Phrase books can also be used to avoid duplication of effort since the translations for a family of applications can be produced once in the phrase book.

If the current string in the **Strings** view appears in one or more of the phrase books that have been loaded, the current string and its phrase book translations are listed in the **Phrases and Guesses** view. If the current string is the same as, or similar to, another string that has already been translated, that other string and its translation are also listed in this view.

To copy a translation from the **Phrases and Guesses** view to the translation area, double-click it or select it and press **Enter**.

Batch Translation

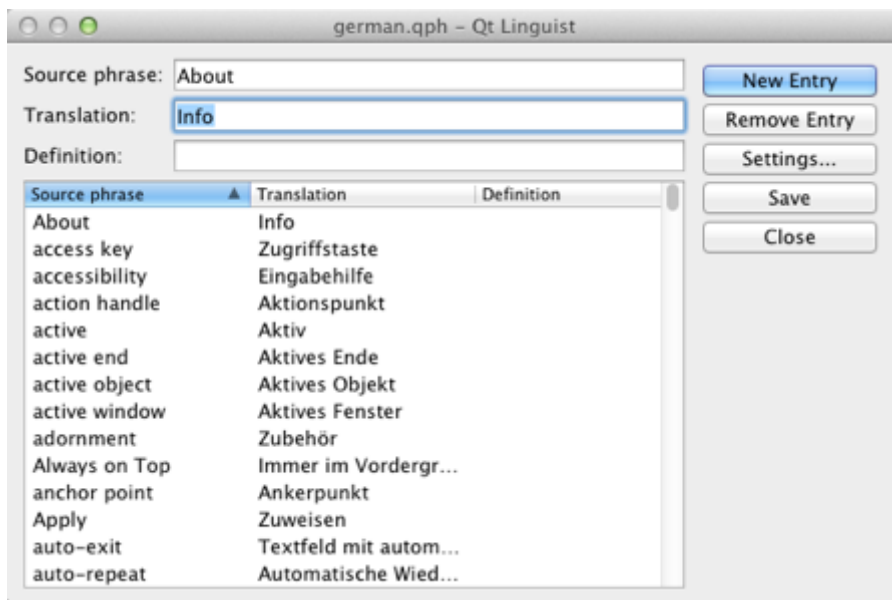


whether batch translated entries should be marked as **Accepted**.

Creating and Editing Phrase Books

Phrase book files are human-readable XML files containing standard phrases and their translations. These files are created and updated by *Qt Linguist* and may be used by any number of projects and applications.

To create a new phrase book, select **Phrases > New Phrase Book**.



To open a phrase book, select **Phrases > Open Phrase Book**, and then select the Qt phrase book file (.qph) to open.

To view and change open phrase books, select **Phrases > Edit Phrase Book**.

To add a new phrase, select **New Entry** (or press **Alt+N**) and type in a new source phrase, the translation, and an optional definition. This is useful to distinguish different translations of the same source phrase.

To add the translation you are working on to the current phrase book, select **Phrases > Add to Phrase Book** or press **Ctrl+T**. If multiple phrase books are loaded, you have to select one.

If you detect an error in a phrase book entry that is shown in the **Phrases and Guesses** view, you can also edit it in place by right clicking on the entry, and selecting **Edit**. After fixing the error press **Return** to leave the editing mode.

To delete a phrase, select it in the **Source phrase** list, and then select **Remove Entry**.

To print an open phrase book, select **Phrases > Print Phrase Book**.

Validating Translations

Qt Linguist provides the following validation tests for translations:

- *Accelerator validation* detects translated phrases that do not have an ampersand when the source phrase does and vice versa.
- *Punctuation validation* detects differences in the terminating punctuation between source and translated phrases when this may be significant. For example, warns if the source phrase ends with an ellipsis, exclamation mark or question mark, and the translated phrase does not, and vice versa.
- *Phrases validation* detects source phrases that are also in the phrase book but whose translation differs from that given in the phrase book.

To switch validation tests on or off, select **Validation** or use the toolbar buttons.

Not accepted strings that fail validation tests are marked with the **Validation Failures** icon in the **Strings** view. Accepted strings are marked with **Accepted/Warnings**.

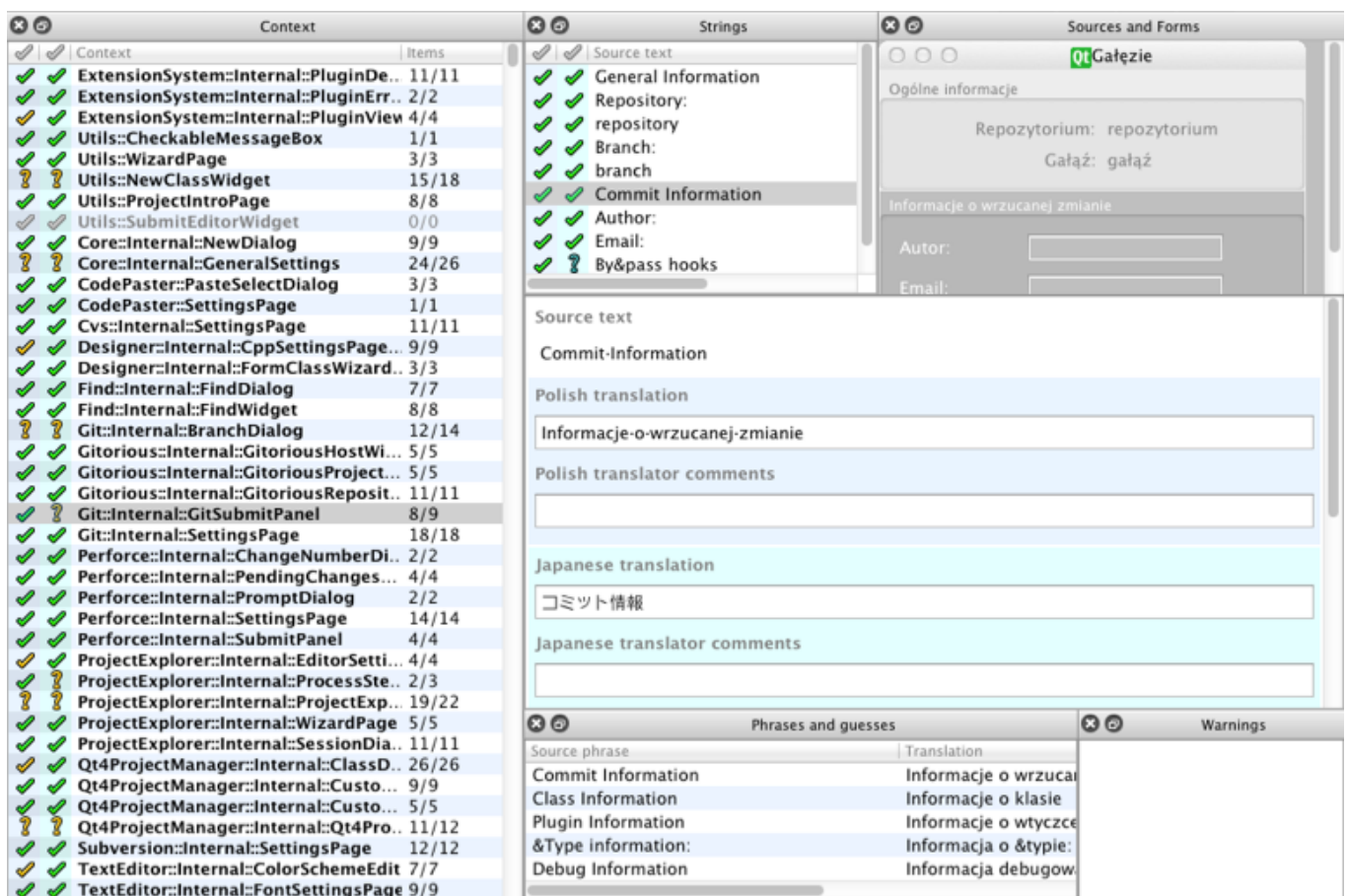
If you switch validation off and then switch it on later, *Qt Linguist* rechecks all phrases and marks any that fail validation.

If the translation you enter for the current string fails any of the active validation tests, the failures are listed in the **Warnings** view. The first of these failure messages is also shown in the status bar at the bottom of the main window.

Note: Only results of *active* validation tests are reported.

Translating Multiple Languages Simultaneously

You can load and edit multiple translation files simultaneously. The following screen shot displays *Polish* and *Japanese* translation files loaded.



The translation area has text editing areas for both Polish and Japanese, and these are color-coded for easier separation. The **Context** view and the **Strings** view both have two status columns instead of one, color-coded with the same colors. The left-most column is related to the top-most language area (Polish above) in the translation area, and the right-most column is related to the bottom language area.

The **Items** column in the **Context** view combines the values for both languages. If the number of translatable strings does not match the number of accepted strings, either or both languages have strings that need to be translated or accented. The **Strings** view shows the translation acceptance state of each string for each language.



© 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](#)

Community

[Contribute to Qt](#)
[Forum](#)
[Wiki](#)
[Downloads](#)
[Marketplace](#)

Licensing

[Terms & Conditions](#)
[Open Source](#)
[FAQ](#)

For Customers

[Support Center](#)
[Downloads](#)
[Qt Login](#)
[Contact Us](#)
[Customer Success](#)

