

Translations

The Bitcoin-Core project has been designed to support multiple localisations. This makes adding new phrases, and completely new languages easily achievable. For managing all application translations, Bitcoin-Core makes use of the Transifex online translation management tool.

Helping to translate (using Transifex)

Transifex is setup to monitor the GitHub repo for updates, and when code containing new translations is found, Transifex will process any changes. It may take several hours after a pull-request has been merged, to appear in the Transifex web interface.

Multiple language support is critical in assisting Bitcoin's global adoption, and growth. One of Bitcoin's greatest strengths is cross-border money transfers, any help making that easier is greatly appreciated.

See the [Transifex Bitcoin project](#) to assist in translations. You should also join the translation mailing list for announcements - see details below.

Writing code with translations

We use automated scripts to help extract translations in both Qt, and non-Qt source files. It is rarely necessary to manually edit the files in `src/qt/locale/`. The translation source files must adhere to the following format:

```
bitcoin_xx_YY.ts or bitcoin_xx.ts
```

`src/qt/locale/bitcoin_en.ts` is treated in a special way. It is used as the source for all other translations. Whenever a string in the source code is changed, this file must be updated to reflect those changes. A custom script is used to extract strings from the non-Qt parts. This script makes use of `gettext`, so make sure that utility is installed (ie, `apt-get install gettext` on Ubuntu/Debian). Once this has been updated, `lupdate` (included in the Qt SDK) is used to update `bitcoin_en.ts`.

To automatically regenerate the `bitcoin_en.ts` file, run the following commands:

```
cd src/  
make translate
```

Example Qt translation

```
QToolBar *toolbar = addToolBar(tr("Tabs toolbar"));
```

Creating a pull-request

For general PRs, you shouldn't include any updates to the translation source files. They will be updated periodically, primarily around pre-releases, allowing time for any new phrases to be translated before public releases. This is also important in avoiding translation related merge conflicts.

When an updated source file is merged into the GitHub repo, Transifex will automatically detect it (although it can take several hours). Once processed, the new strings will show up as "Remaining" in the Transifex web interface and are ready for translators.

To create the pull-request, use the following commands:

```
git add src/qt/bitcoinstrings.cpp src/qt/locale/bitcoin_en.ts
git commit
```

Creating a Transifex account

Visit the [Transifex Signup](#) page to create an account. Take note of your username and password, as they will be required to configure the command-line tool.

You can find the Bitcoin translation project at <https://www.transifex.com/bitcoin/bitcoin/>.

Installing the Transifex client command-line tool

The client is used to fetch updated translations. If you are having problems, or need more details, see <https://docs.transifex.com/client/installing-the-client>

```
pip install transifex-client
```

Setup your Transifex client config as follows. Please *ignore the token field*.

```
nano ~/.transifexrc

[https://www.transifex.com]
hostname = https://www.transifex.com
password = PASSWORD
token =
username = USERNAME
```

The Transifex Bitcoin project config file is included as part of the repo. It can be found at `.tx/config`, however you shouldn't need to change anything.

Synchronising translations

To assist in updating translations, a helper script is available in the [maintainer-tools repo](#). To use it and commit the result, simply do:

```
python3 ../bitcoin-maintainer-tools/update-translations.py
git commit -a
```

Do not directly download translations one by one from the Transifex website, as we do a few post-processing steps before committing the translations.

Handling Plurals (in source files)

When new plurals are added to the source file, it's important to do the following steps:

1. Open `bitcoin_en.ts` in Qt Linguist (included in the Qt SDK)
2. Search for `%n`, which will take you to the parts in the translation that use plurals
3. Look for empty `English Translation (Singular)` and `English Translation (Plural)` fields
4. Add the appropriate strings for the singular and plural form of the base string
5. Mark the item as done (via the green arrow symbol in the toolbar)
6. Repeat from step 2, until all singular and plural forms are in the source file
7. Save the source file

Translating a new language

To create a new language template, you will need to edit the languages manifest file

`src/qt/bitcoin_locale.qrc` and add a new entry. Below is an example of the English language entry.

```
<qresource prefix="/translations">
  <file alias="en">locale/bitcoin_en.qm</file>
  ...
</qresource>
```

Note: that the language translation file **must end in** `.qm` (the compiled extension), and not `.ts`.

Questions and general assistance

If you are a translator, you should also subscribe to the mailing list,

<https://groups.google.com/forum/#!forum/bitcoin-translators>. Announcements will be posted during application pre-releases to notify translators to check for updates.