Translation Guide to FileWave

The translation files for FileWave are stored in a git repository on the http://www.github.com website. You will need to know the basics about using the git source control system. There are many helpful sites to help you learn git. Here is one example, http://git-scm.com/documentation

You will need a free github account to get started. After you've got your account, send an email request to support@filewave.com and request access to the translations repository. Be sure to include your github account name in your email request.

Overview of the Translation Process

The first step you will need to perform is a "fork" of the translations repository – which creates your own private copy of the translations, ensuring your work is isolated from other translators.

You will then perform the translations and testing using a local copy of the forked repository.

When your changes are complete, you can commit the changes locally and push them into the forked repository on github.

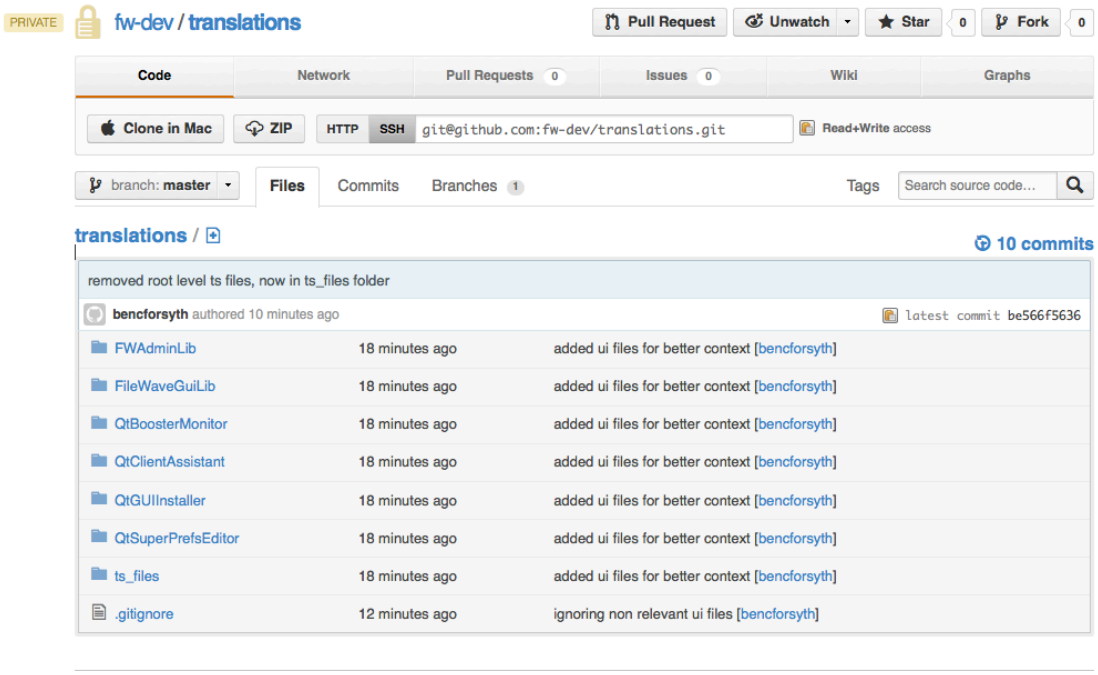
Then you make a "pull request" to have the changes integrated into FileWave's main repository.

Each of the above steps is outlined in more detail below.

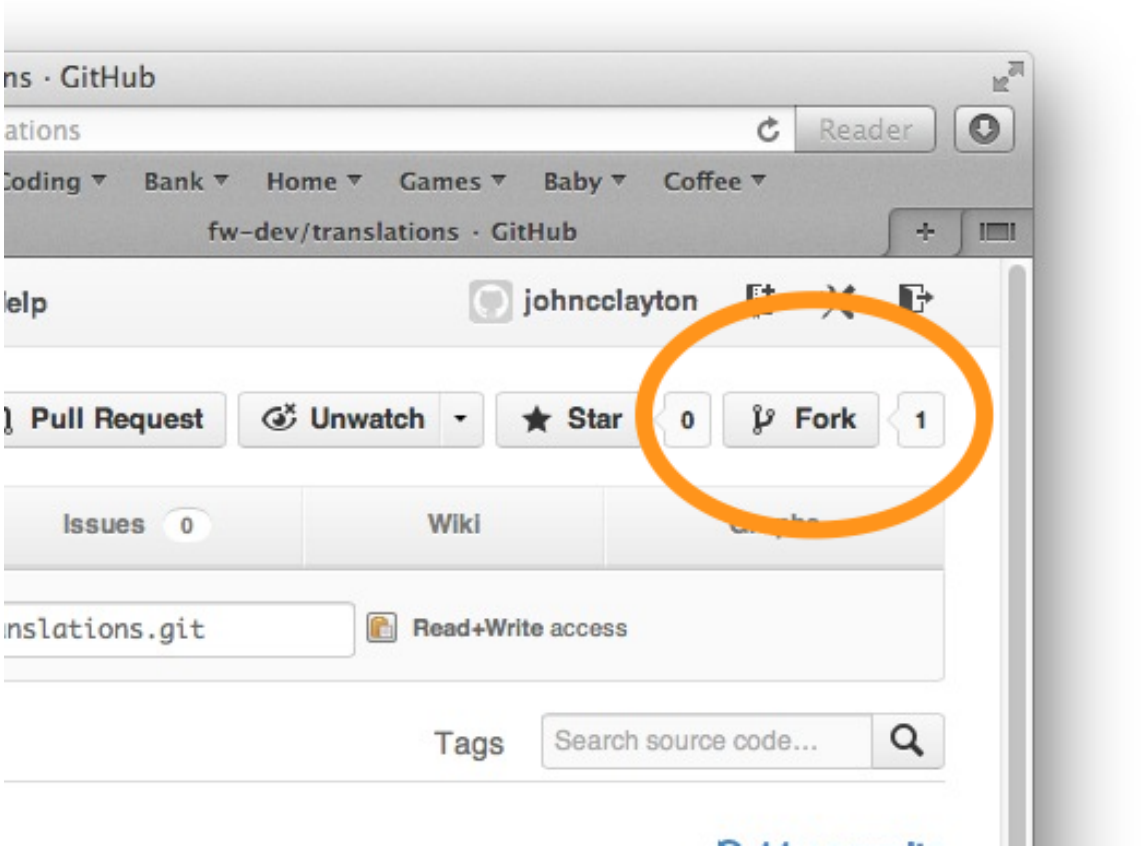
Creating your own Fork

After you've been given access to the FileWave/translations repository, go to this web address
https://github.com/fw-dev/translations

You should see a view similar to the one below



You'll want to click on the "Fork" button to make a copy of the translations.



When the fork process is complete, you will have your own copy of the translations repository on github.

Now you need to bring a copy of this newly forked repository to your local computer.

Obtaining a local copy of the fork
Next step is to make a copy of the github repository to your local computer. We recommend using a GUI client to do this.

There are many free git clients available. A good one for Mac OS is called 'Source Tree' and it is available from the Mac App Store.

A windows git GUI tool is 'Git Extension' it is available from:
http://code.google.com/p/gitextensions/

The site also has demo videos of how to clone a repository.

For example, assuming I have a github account called fancy_new_translator, then I would want to clone the repository called:
git@github.com: fancy_new_translator /translations.git

Doing so will result in a local copy of the translations repo on your disk drive.

Performing the translation

Now you can start the job of translating. The files you checked out of github are xml documents that can be edited by the QtLinguist application.

Using the QtLinguist application makes it easy to understand where the translations will appear in the product, and which context they have (i..e does the translation string appear in the login window or some other part of the UI?).

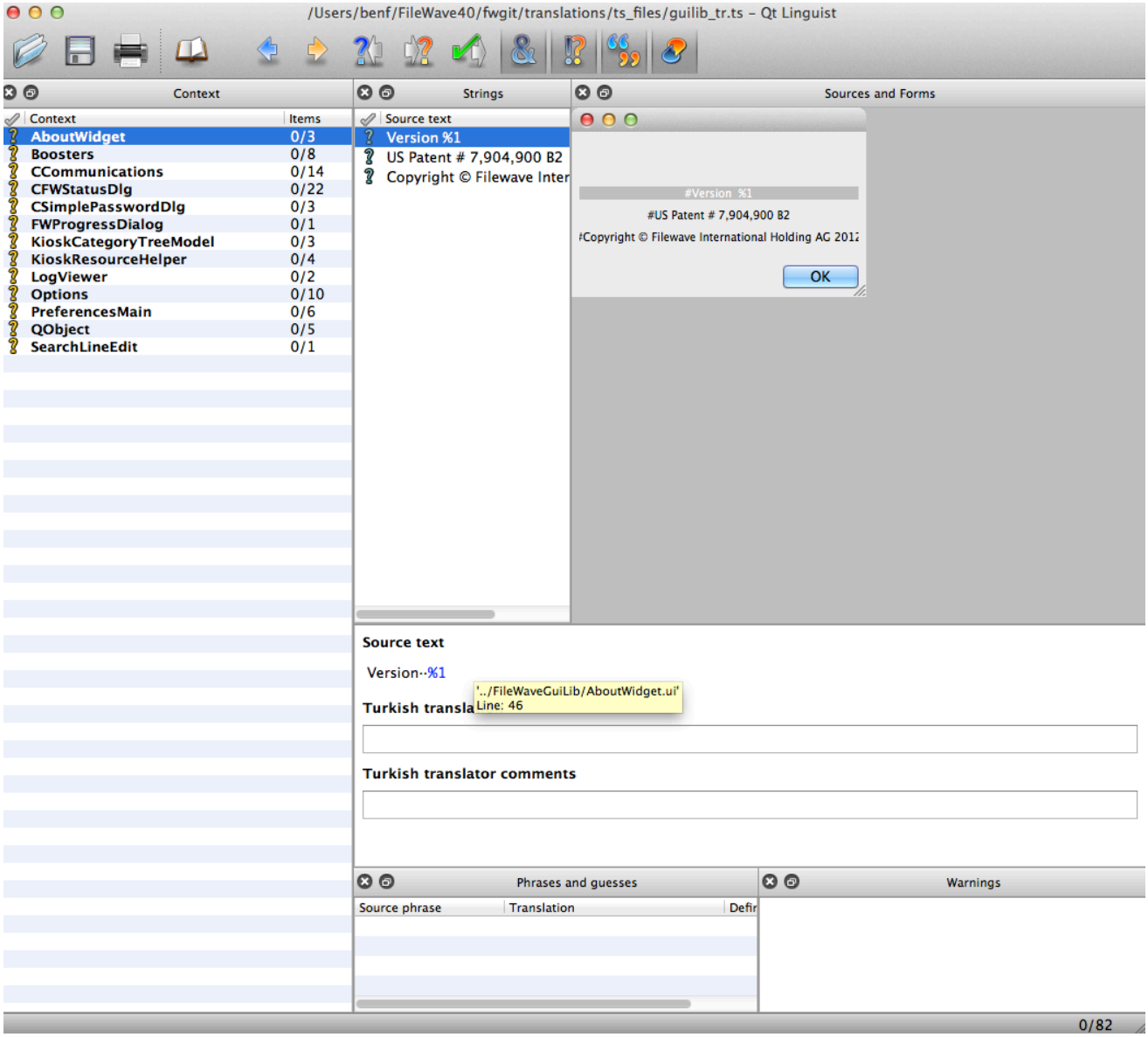
QtLinguist is available for download from:
http://qt-apps.org/content/show.php/Qt+Linguist+Download?content=89360

Linguist has a user manual accessible from the Help menu. You should read through the manual to become familiar with the Linguist application before proceeding.

Launch Linguist and open one of the ts files from the ts_files folder.

Language	ts_file	Application
Turkish	admin_tr.ts	FileWave Admin
Turkish	guilib_tr.ts	FileWave Admin
Turkish	booster_monitor_tr.ts	Booster Monitor
Turkish	client_assistant_tr.ts	Client Installer
Turkish	client_tr.ts	Client Monitor
Turkish	super_prefs_tr.ts	SuperPrefs Editor

Opening the guilib_tr.ts file should give you a view similar to this



The relevant user interfaces are displayed to help give context to the translation.

After a ts file is completed for translation you can test it by doing the following:

- Go to the File menu of Linguist and select “Release” this will create a file that ends in “qm”
- Create a folder (if it doesn’t already exist) named “translations” in the same location as the FileWave Admin application
- Copy the qm file you created into the translations folder and launch the FileWave Admin. You should see your translated user interface
- The qm files must be named as below

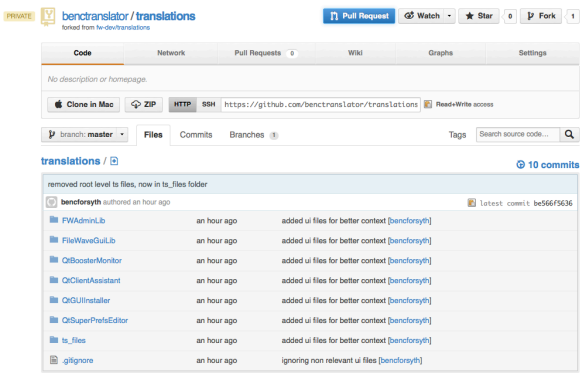
Language	ts_file	qm file name
Turkish	admin_tr.ts	admin_tr_TR.qm
Turkish	guilib_tr.ts	guilib_tr_TR.qm
Turkish	booster_monitor_tr.ts	booster_monitor_tr_TR.qm
Turkish	client_assistant_tr.ts	client_assistant_tr_TR.qm
Turkish	client_tr.ts	client_tr_TR.qm
Turkish	super_prefs_tr.ts	super_prefs_tr_TR_qm

Check Translations into Github

When you have finished translating you will need to commit and push your changes to the ts files into your github translations repository.

Once you have pushed your changes into github, you will make a “Pull Request” to have those changes integrated into FileWave’s translation repository.

Login to github with your web browser and click on the “Pull Request” button. This will send a notification to the FileWave Developers that you want to have your changes committed to the translations repository.



Clicking the Pull Request button will bring up another web ui. It should automatically have the FileWave translations repository set as the destination.

Then chose your master or branch with your translations to pull.

After your pull request is approved, FileWave's developers will "pull" your changes into the translations repository. It could be that you changed a file incorrectly or that there was some other problem. A FileWave developer will reply back with any comments or changes that are needed.

Logos and Pictures

There are 2 different corporate graphics in use throughout the FileWave suite of applications. They are the red "sail" icon that is used as the main icon of the FileWave product (the sail icon comes in 4 sizes 256, 128, 64 and 32) and there is a 2nd logo used in the FileWave Admin's Server Status dialog which is the sail icon combined with the filewave text.

The Booster monitor uses its application icon as the application icon in its connect dialog.

Pixmap locations
The pixmaps will be loaded from the translations directory located in the same directory as the FileWave Admin.app or FileWave Admin.exe.

For the "sail" icon used by FileWave Admin the names must be:
main_logo_32.png
main_logo_64.png
main_logo_128.png
main_logo_256.png

where the number (32, 64, etc..) is the size in pixels of the artwork

This is the logo used in the Server Status window. The size should be 400 pixels wide:
main_logo_text.png

Application Icons and Name

Mac OS

You will need to edit the each applications Info.plist. The plist is an xml document. You will need to edit the values for the keys

CFBundleName
CFBundleIconFile

CFBundleName will change the name that is displayed to your end users. CFBundle icon is the name of the icon file. The icon file must be placed in the Resources folder inside the .app bundle. The icon file must be in icns format and should contain 32, 64, 128 and 256 pixels.

Windows

You will need a resource editor to edit the icon inside the application. There is a free editor available from http://www.resedit.net/

You will need to replace the existing icon resources with an ico icon file containing the icons in 32, 64, 128 and 256 pixels.