

Internationalizing the XSupplicant UI

By, Chris Hessing

© Copyright 2009, The Open1X Group

It is important to the developers of XSupplicant that users are able to make use of the supplicant in a language that is comfortable to them. To this end, the supplicant UI was developed from the ground up with the ability to be internationalized. Internationalization of the UI is done through the Qt Linguist tool that is available as part of the Qt SDK. This document is meant as a quick start to get users up and translating quickly. For a full discussion of the Qt linguist program it is suggested that you look at the TrollTech web site at the Qt Linguist Manual.

Getting Started

The first thing you need to do to create a translation of the UI is to determine the name that Qt uses for your locale. To make this easy, the supplicant will display the local string on the about window. If you right click on the supplicant's tray icon, and select “About...” you should get a dialog that looks similar to this :



At the bottom of the dialog is an item for “Current Locale”. In this image I have circled the locale value that my machine is set for. In this case, my machine is set to use the “en_US” locale. (Note: Case is important! So make sure you note the exact case that is used!)

Once you have determined your locale, you need to extract all of the translatable strings from the UI source files. Doing this is surprisingly easy. In the source directory go in to the xsupplicant-ui/xsupptray directory, then run the following command(s):

On Windows :

```
c:\qt\<version>\bin\lupdate XSupplicantUI.pro -ts XSupplicantUI_<locale>.ts
```

The <version> string should be replaced with the version of Qt that you are using. At the time of this writing I am using Qt 4.3.4. The <locale> should be the name of the local you are translating to. For my example, I am going to create a translation for the en_US locale (even though that is the default locale). Based on the Qt version and locale I am using, I need to run this command :

```
c:\qt\4.3.4\bin\lupdate XSupplicantUI.pro -ts XSupplicantUI_en_US.ts
```

This will create a file called “XSupplicantUI_en_US.ts” that is used as the source for internationalization. Assuming everything worked properly you should get a status line similar to this :

Found 808 source texts (808 new and 0 already existing)

And then be returned to your command prompt.

Next, you need to run the Qt Linguist application. This can either be found via your normal OS application launcher (i.e. The Start menu on Windows), or can be run from the command line. In my example, I am running it from the command line with the following command :

```
c:\qt\4.3.4\bin\linguist.exe
```

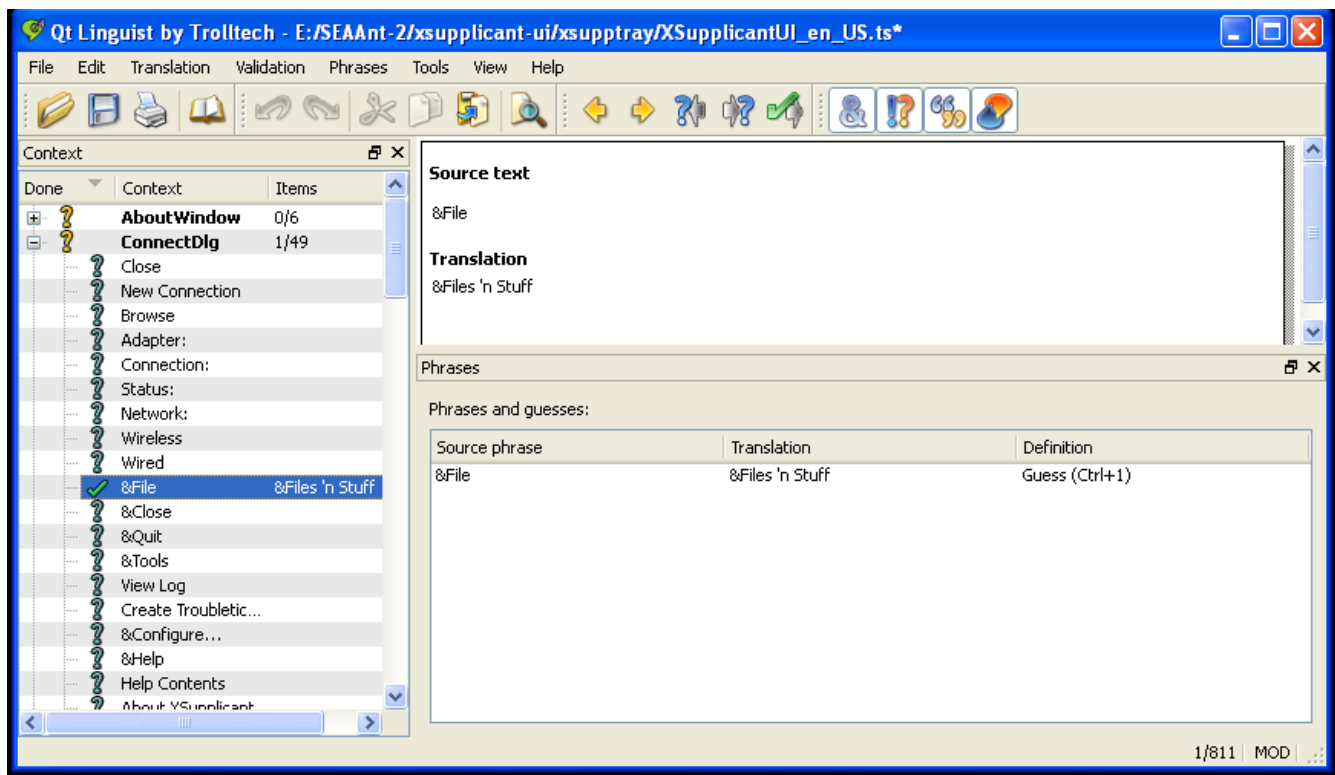
You should now be presented with the Qt Linguist application. Next you want to open the .ts file that we created in the above steps. To do that select File | Open, then locate “XSupplicantUI_en_US.ts”, then click “Open”.

You will now be presented with a list of Contexts that can be translated. For my example, I am going to translate the “File” menu option, so I have expanded the “ConnectDlg” context, and located the string “&File”. Clicking on that item will allow you to enter the translated text in the main form window below the **Translation** line.

Since this is a quick example, I am going to 'translate' the “File” menu to be called “Files 'n Stuff” instead. So under the **Translation** line, I have entered “&Files 'n Stuff”. (The '&' character indicates that the character following it is the hot key for that option. Since this is a main menu the file menu hot key will be Alt-F.)

Once I have entered the translation, I need to tell linguist that I am done translating that item. In the tool bar at the top is an icon that contains a check mark and an arrow to the right. If you hover over that icon it should display “Done and Next”.

The end result should look similar to the screen shot below.



At this point, you would normally finish translating all of the other strings that are displayed. But, for our example we only want to translate the one string. So, save the file and exist linguist.

Back at the command line, change to the same directory that you were in when you created the original file to translate. You now need to create the release file for the language. This is done by running the lrelease command against your translation. For me, the command line looks like this :

```
c:\qt\4.3.4\bin\lrelease XSupplicantUI_en_US.ts -qm XSupplicantUI_en_US
```

After running that command you should get output results similar to this :

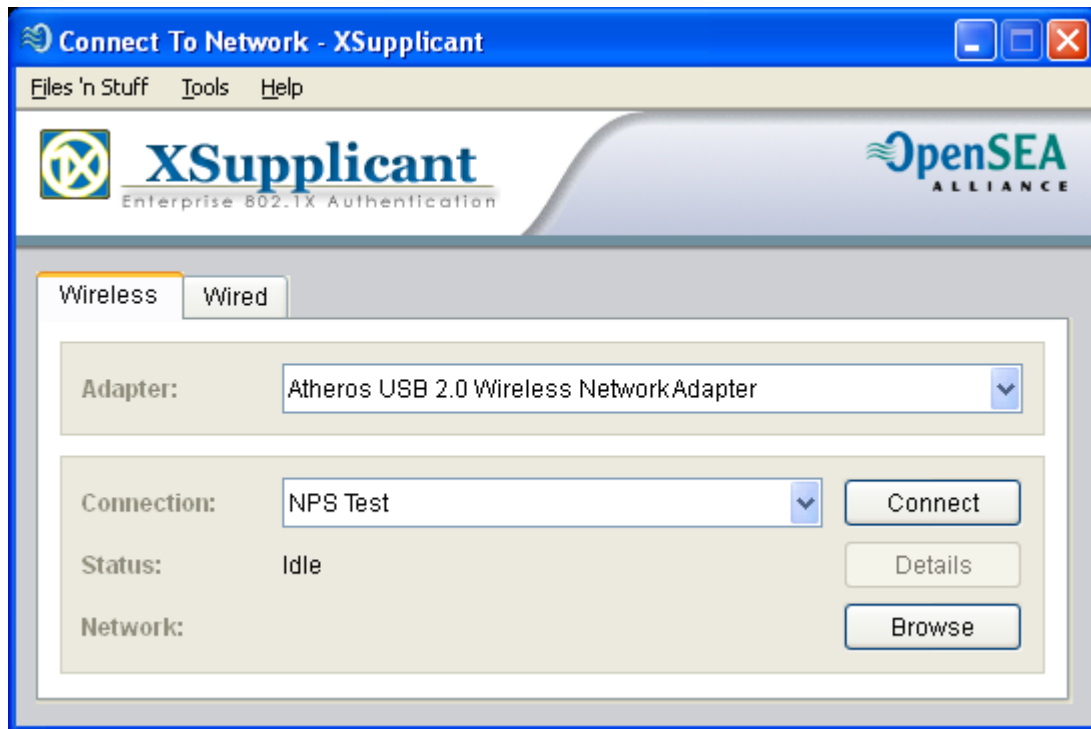
```
Updating 'XSupplicantUI_en_US'...
Generated 1 translation (1 finished and 0 unfinished)
Ignored 810 untranslated source texts
```

And a file called XSupplicantUI_en_US will be created. To use this file, it just needs to be copied to the directory that the XSupplicantUI.exe file is located in. On Windows, this is usually [c:\Program Files\XSupplicant](#).

Once the file has been copied there, you are ready to use your translated version. If the UI is currently running, exit it. Then start the UI back up, and as long as your locale matches the locale file you created all of your translated strings should now show up.

When I restart the UI and double click the tray icon I am presented with the normal “Connect To Network” dialog. However, as expected the “File” menu is now called “Files 'n Stuff” because of translation file.

Pictures speak louder than words, so here is a screen shot :



(Notice in the top left the file menu name has been “translated”.)

Now What?

Once you have completed your translation, the Open1X team would love to include it in our build system for future releases. There are two ways that you can use to get your translation included. You can open a bug on the Open1X SourceForge site, with the .ts file attached to it, or you can join one of the mailing lists and send it to the list.

Please note that we need to unprocessed translation file (the .ts file) in order to include your work in the build. This will allow us to make minor changes to the UI without rendering most of your translation useless.

Also, please include an e-mail address where you can be reached. Once your file has been accepted we will contact you to verify that you are okay with the license that the file will be distributed under, and to see if you want to have any information included in the source tree recognizing your work. (Information may be as simple as your name, or at your option may include a way of contacting you.)