Subject: A proposal: Adding a Translation & Localization GUI to TheIDE
Posted by Oblivion on Sun, 05 Jun 2011 13:49:20 GMT
View Forum Message <> Reply to Message

Hello guys,

I started using TheIDE, and eventually learning the U++ of course, in 2007, and since then it became my first choice of C++ framework and IDE. As far as I can see, and from my own experience I can confirm, that one of the major downsides of U++ and theIDE is that both have a higher learning curve compared to other traditional frameworks (such as QT/GTK, even MFC), and this may and sometimes does scare beginners/newbies/newcomers. While theIDE is very very powerful and effective, I believe it needs some polishing and simplifying on the UI side. So as the title suggests, my main target here is the internationalization & localization tools of TheIDE. I was for some time studying the internals of theIDE and I found out that given the current (U++ daily builds) codebase of TheIDE it is very easy to come up with a nice GUI which will help the developers to translate their projects into other languages. I have already written an experimental, barebone UI (which is inspired from KDE's Lokalize)for project translations and patched my local copy of theIDE.

Here is what I propose:

Primary Goals:
1. Creating a 'Translator' package & ctrl under Ide/Translator, and encapsulating the current 't.cpp' code within it. Current code is really easy to use and manipulate, so this is no big deal. Imo, a seperate Translation app won't be useful, for this will need the code to handle packages which is already in the TheIDE codebase to be rewritten. Also, integral translation UI will give user the impression that TheIDE is cooler than ever

2. Using the new so-called 'Translator' codebase create a decent GUI analogous to layout editor, which will give the developer the chance to handle t. files both in a new and KDE's Lokalize alike UI integral to TheIDE or in the current text based t[ranslation] editor of TheIDE.

3. To simplify export/import/sync operations, using a translation "wizard". Imho, the current export/import/sync dialog is somewhat obscure for a newcomer.

Secondary Goals

4. Adding "optional" dictionary (dict, or maybe... google translate ??) and internal database support (as in Lokalize).

5. Extending t. file structure using XML (from my experience on large projects I know that translation strings need to be individually revised, rejected, commented, etc. In some cases handling these flags become important.

Now, this is not a GsoC project, nor a wish. I am currently working on this project simply because I need a better and faster way of translating projects (since I use U++ a lot) and I want to contribute to U++ and theIDE and this community, which simplifies my life.

I would like to hear your thoughts, critisicm and suggestions on this project?

Regards.
