
Arabeyes Translator Guide

Arabeyes Core Team , Arabeyes Project <contact (at) arabeyes dot org>

\$Revision: 1.27 \$

This document is meant to serve as a guide for Arabeyes' [<http://www.arabeyes.org>] translators and those wanting to join the Arabeyes translation team.

Table of Contents

License	1
Introduction	2
Credits	2
Translations	2
Feedback	2
First Steps... ..	2
The Translation Process	3
Online Translation References	3
Common Guidelines and Recommendations	4
'PO' and 'POT' Files	4
'PO' Files Structure	5
Comments within 'msgid' ('_:')	7
Special keywords within 'msgid' ('Comment=' or 'Name=' or others)	7
HTML tags within 'msgid'	7
Carriage Returns & Punctuation within 'msgid'	8
Multi-listing 'msgstr' for plural forms	8
Shortcuts & underscores in 'msgid'	8
Names and Acronyms within 'msgid'	8
Special Characters within 'msgid'	8
Use of the Imperative Form for Actions	9
UTF-8 Encoding	9
Sanity checking (any errors ?)	9
Commit check-list	10
Translation Tools	10
Text Editors	10
KBabel	11
gTranslator	12
poEdit	12
New Translation Projects and Arabic Documentation	12
New Translation Projects	13
Writing Arabic Documents	13
What Coordinators Do (Duties, Tasks)	13

License

Copyright (c) 2003, Arabeyes Project, Youcef Rabah Rahal.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation;

with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Introduction

This document will serve as a guide to all translation projects and procedures on Arabeyes.

It will detail required information you might find useful about registration and translation and will hopefully answer the many commonly asked questions you may have during this process. You can refer to this document at any moment while translating.

Arabeyes is eager to hear from you and welcomes you aboard !!

Credits

Here's the list of people who contributed actively to this document:

- Isam Bayazidi
- Mohammed Elzubeir
- Youcef Rabah Rahal
- Nadim Shaikli

Translations

There are currently no known translations of this document.

Feedback

Please inform us of any information that might be lacking in this document as well as any areas which you deem require more clarifications. We've attempted to be as thorough and as complete as possible and tried to include *all* the information translators might need. If you have any questions, suggestions, corrections or concerns, we'd highly appreciate your feedback.

First Steps...

If you are fortunate enough to have good language skills in both Arabic and English (i.e. speak them fluently and can write them without difficulty) and you've decided to participate in the translation effort ? We will then show and outline for you what you're required to do in order to become a member and an active translator within the Arabeyes Project and its team of enthusiasts !!

1. Subscribe to the 'doc' mailing list [<http://lists.arabeyes.org/mailman/listinfo/doc>] then inspect, read through and peruse the doc list archives [<http://lists.arabeyes.org/archives/doc/>].
2. Post a message to the 'doc' list (doc at arabeyes dot org) introducing yourself and stating where your interests lie and which project piques your curiosity (See the Projects List [http://www.arabeyes.org/proj_trans.php] for choices). The project's coordinator will assign you a task (if need be): a file, a group of files or a module to translate.

¹Minimal computer knowledge is required.

3. Register [<http://www.arabeyes.org/register.php>] for a CVS account. This will ultimately give you the ability to share your work within Arabeyes (Read the CVS HOWTO [<http://www.arabeyes.org/download/documents/howto/cvs-howto-en/>]).

You can, at any stage of course, participate in the discussions on the 'doc' mailing list and post your thoughts, suggestions or questions. A coordinator or some well-versed translator(s) with appropriate knowledge will all be more than happy to respond to your concerns.

The Translation Process

After registration or while waiting for your CVS account information email (which ought to be sent in 2-3 days), you can proceed down the following steps:

1. You must have Arabic support setup on your system. If you don't already have Arabic support on your Linux installation, then please refer to the Arabic HOWTO [<http://www.arabeyes.org/download/documents/howto/arabic-howto-en/>] in order to properly configure matters.

If you are using an MS-Windows Operating System machine, we *highly* recommend you start using Linux instead (consider a partition install in which you can use both). Linux will open many new doors and will offer you the possibility to discover the fabulous world of Open Source Software :-). You are also very unlikely to get good answers to your questions if you insist on using MS-Windows (as most, if not all users on Arabeyes, use Linux or a unix derivative).

2. Once you've decided on which project you are going to work on, proceed to check out its files. If you have a fast internet connection (DSL or better or are willing to stay online for prolonged periods of time with a dialup), it's recommended that you checkout the entire project's files (i.e. if you will work on GNOME translation for example, then checkout `translate/gnome/` directory). The massive checkout will help you in building your KBabel database/dictionary among other things. If, on the other hand, you have a more restrictive Internet connection and are more limited on your time, you can download only the file(s) that you will be actively working on and translating. Please note, you don't need to have a CVS account to checkout the files; check the CVS HOWTO access section for further details.
3. Continue reading this document with special attention to specific translation guidelines (here) as well as the translation tools. You should also read the Arabic Documentation Standards [http://www.arabeyes.org/download/documents/misc/doc_standards-en/] for better familiarity.
4. Start translating the file(s) :-)
5. Please commit your file(s) regularly. You don't have to wait to finish a file's translation in order to commit it into CVS either. Refer to the CVS HOWTO guidelines section concerning committing. The "committing often" is crucial in order to avoid disasters (the mistaken removal of a file for instance) and conflicts (though conflicts are supposed to be unlikely since normally there should only be ONE person working on a same file at any one time). If for any reason you are not able to commit your file(s) via CVS, you can send the files as email attachments to `translate (at) arabeyes dot org` and post that you've done that to 'doc'.

Online Translation References

Here is a list of some usual references that you may find useful while translating and which you can use in parallel with any printed dictionary that you may have (specific technical/computer terms dictionaries are preferable, and especially those edited by ALECSO [<http://www.slis.uwm.edu/alecso/Default.htm>]). The already translated PO files are also a good reference. If you still have problems with a word or an

expression that you can't translate, then expose the issue on the 'doc' list.

1. Ajeeb website [<http://www.ajeab.com>]. You can directly use the dictionary [<http://dictionary.ajeab.com>] and/or the translation section [<http://tarjim.ajeab.com>].
2. Almisbar website [<http://www.almisbar.com>].
3. Qamoose dictionary [<http://www.arabeyes.org/cgi-bin/qamoose/index.cgi>].
4. Onelook website [<http://www.onelook.com>]. Which is an english online dictionary that groups together many online dictionaries. Use it to find english definitions of words you are not sure of.
5. Multidisciplinary Dictionary [<http://www.arabicdictionaries.com/>].
6. Foreign Word [<http://www.foreignword.com/>].

Common Guidelines and Recommendations

Here we will review some guidelines and make recommendations that are independent of the tools you will use to translate files. If you use a simple text editor or a more sophisticated tool like KBabel, do please read the following section as it pertains to both usage means. The intent here is educate what needs to be done to translate your files. A common structure and procedure is needed from and by all to result in coherent translation practices.

'PO' and 'POT' Files

Graphical Desktop Environments like KDE and GNOME and the various applications that ship with them are written using C/C++ programming languages. Translating these environments means translating the tens of thousands of user visible strings from their native language (i.e. English) to a target language (i.e. Arabic). This is NOT done by copying the source code and replacing all the English strings with Arabic ones, its done by translating already extracted strings into their various files in order to ease the overall process.

The user visible strings are, thus, extracted from the source code with a program called 'gettext'. They these strings are extracted then are stored in a text file with a '.pot' extension. The '.pot' extension means that this file has NOT been translated yet (i.e. its in its original form). If you want to start a 'POT' file translation, then you have to first rename the file with the same name but with a '.po' extension. Let's say that you want to start translating the file 'filename.pot', you would then:

```
$ mv filename.pot filename.po
```

Work on translating the file and when satisfied with the amount of work you have accomplished, do the following:

```
$ cvs remove filename.pot
$ cvs add filename.po
$ cvs commit2
```

The '.po' extension means that the translation has started on this file, without necessarily meaning that it is finished. Please remember to never translate 'POT' files. Either the files you are translating already

have a '.po' extension or you'll be forced to rename them according to what has been described above. You are not required to change the extension of 'POT' files that you will NOT be translating in the very near future. This renaming has come to sorta mean that the person that renamed them owns them and will be working on them. If you have any doubts/questions, post to the 'doc' mailing list.

At run-time, the desktop environment will load the user visible strings from the target language translated 'PO' files and display them accordingly without any apparent change in the program behavior. If the translation is incomplete, the environment will simply display the original English strings where the translation is missing.

The sections that follow will give some guidelines with regards to how to translate 'PO' files. Remember that you can always use the 'doc' mailing list whenever you have *any* concerns (it would be unrealistic of us to pretend that these guidelines treat all issues that a translator may face). Translators are also encouraged to look at already translated 'PO' files in the CVS repository in a possible bid to learn further the various issues noted.

'PO' Files Structure

The Header

All files carry relevant info with regards to who the translator(s) were, what encoding the file uses, various date info (creation, revision, etc). This info is all stored within the header of the file (i.e. in the beginning of the file). The information is of course editable since the info is liable to be changed.

Here's a sample header of a newly created file:

```
#, fuzzy
msgid ""
msgstr ""
"Project-Id-Version: desktop files\n"
"POT-Creation-Date: 2002-08-14 03:34+0200\n"
"PO-Revision-Date: YEAR-MO-DA HO:MI+ZONE\n"
"Last-Translator: FULL NAME <EMAIL@ADDRESS>\n"
"Language-Team: LANGUAGE <LL@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=CHARSET\n"
"Content-Transfer-Encoding: ENCODING\n"
```

With all newly created files you are to remove the "#, fuzzy" line and add at the beginning of the file the following string "# translation of filename.po to Arabic" (replace "filename.po" with the appropriate filename of the file you are translating), add the copyright information string "# Copyright (C) 2003 Free Software Foundation, Inc." and then add your name and email after the copyright line (this will be the translators list). Then change the, "Last-Translator:", "Language-Team:", "Content-Type:" and "Content-Transfer-Encoding:" fields so that they will look as shown below (don't bother with any other existing or non-existing fields for the time being). If you will use a text editor in order to translate the files (as apposed to a translation application like KBabel), remember to update also the "PO-Revision-Date:" field every time you save your files.

²The CVS commit log ought to be something akin to, 'renamed filename.pot to filename.po' (the translation statistics need to be included as well as is noted here). If you renamed more than one 'POT' file, do please commit them all via a single command. In other words, after moving, adding and removing all the files, do 'cvs commit' and all will be picked up. If ever in doubt, refer to the CVS HOWTO 'Committing Multiple Files' section.

Here's a sample of the header while its being translated (or finished):

```
# translation of desktop_kde-i18n.po to Arabic
# desktop.po - Arabic Translation.
# Copyright (C) 2001,2002,2003 Free Software Foundation, Inc.
# Name1 Surname1 <A_address/@domain_1.com>, 2001.
# Name2 Surname2 <B_address/@domain_2.net>, 2002
# Name3 Surname3 <C_address/@domain_3.org>, 2002,2003
#
msgid ""
msgstr ""
"Project-Id-Version: desktop_kde-i18n\n"
"POT-Creation-Date: 2003-03-02 20:47+0100\n"
"PO-Revision-Date: 2003-02-26 21:27+0200\n"
"Last-Translator: Name3 Surname3 <C_address/@domain_3.org>\n"
"Language-Team: Arabic <support at arabeyes.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=UTF-8\n"
"Content-Transfer-Encoding: 8bit\n"
"X-Generator: KBabel 1.0.1\n"
```

The lines starting with the '#' character are comments. If you use a translation application (like KBabel), some header fields are automatically created like the "X-Generator:" field. If you are working on a 'PO' file already started by someone else, and you have made *significant* changes, then just add your name to the translators list and put your name and email in the "Last-Translator:" field (after checking that the person who was in the "Last-Translator:" field is already in the translators list, otherwise add him/her to that list). Check also that the above cited fields have the correct/complete values entered in them. If not, correct them.

Always remember to add your name to the translators' list. It's very important to know who had worked on what file plus it will ensure that you will get the credit you deserve for your work :-)

The Body & Strings

Beyond the header, a 'PO' file is a mere succession of 'msgid' and 'msgstr', preceded sometimes by comments about the exact location of where those strings appear within the source code³. The 'msgid' is already filled with the English string that needs to be translated. What translators need to do is to fill 'msgstr' with the Arabic translation of what appears in the 'msgid' string (do NOT delete the double quotes ' ').

Sample excerpts from different 'PO' files follow,

```
#: finddialog.cpp:55
msgid "Galaxies"
msgstr "#####"

msgid ""
"_: do not use a target symbol\n"
"No Symbol"
msgstr ""

#: src/acme.h:80
#, fuzzy
msgid "Eject key"
msgstr "#####"
```

³These types of comments are initiated by "#:" at the beginning of the line.

The first 'msgid' is simply translated, the second string is not translated yet (that's why the 'msgstr' is empty). The third string is translated but the translator has put a "#, fuzzy" statement before the block. The "fuzzy" indicator means by which a translator indicates that he/she are not sure of the term and that a second pass is required. The "#," character is of key importance, its a special character sequence. It is important to note that the "#, fuzzy" statement must come *before* the 'msgid'.

Comments within 'msgid' ('_:')

'msgid' strings starting with '_: ' indicate a comment to help the translator or to give him/her more information about the message to translate is included. The comment ends with a '\n' (new line) sequence. Do NOT translate the help comment and simply put in the 'msgstr' the translation of only what follows the comment's '\n' (new line) sequence according to what you understood the comment was directing you to do.

Example:

```
#: cupsdconf.cpp:808 cupsdconf.cpp:831 cupsdconf.cpp:847
msgid ""
"_: Base\n"
"Root"
msgstr ""
"#####"
```

Special keywords within 'msgid' ('Comment=' or 'Name=' or others)

Translate only what comes after the '=' character while keeping the keyword intact (i.e. don't translate or modify "Comment" or "Name"); keep the keywords in *English*.

Example:

```
#: kde-i18n/vi/messages/entry.desktop:1
msgid "Name=Vietnamese"
msgstr "Name=#####"
```

HTML tags within 'msgid'

Some applications use Rich Text features (for paragraphs, colors, bold fonts, etc) to make their strings look nicer. This is accomplished while using HTML tags such as the "<p>" tag for instance. It is *very* important that the translated message keep all those tags. This is also the case for C print characters like "\n" or "\t" etc or "%s". Translate the text between the tags only. Also, remember that when you are in Right-to-Left input mode (i.e. when your are typing Arabic), the "\n" sequence may look odd - "n\". In either mode, remember that you need to enter first the "\" character and then the "n" irrespective of how they look on screen.

Example:

```
#: kio/kio/global.cpp:451
msgid "</p><p><b>Details of the Request</b>:"
msgstr "</p><p><b>#####</b>:"
```

Carriage Returns & Punctuation within 'msgid'

Keep the same punctuation in the translated messages as those found in the original strings. Carriage returns (or new lines) need to also be included. So if there multiple strings within a single 'msgid', the resulting 'msgstr' ought to mimic the original message in line number count (lines beginning with "_" ought to be ignored as noted above since they are mere comments). See the example of comments.

Multi-listing 'msgstr' for plural forms

The plural forms feature offers the possibility to display the correct word form depending on the quantity. They are distinguished by the "%n" character.

Sample:

```
msgid "_n: one cat\n"
"%n cats"
msgstr "####<\/p><p><b>#####</b>\n"
"#####\n"
"%n ###\n"
"%n ####"
```

The first 'msgstr' is displayed when there's a single cat. The second is displayed when there are 2 cats. The third is displayed when there are 3 to 10 cats. The last entry is displayed when the number of cats is equal-to or greater to 11. There is no need to report the "_n" in the translated message. Don't forget to use the "\n" character to separate between the different entries.

Shortcuts & underscores in 'msgid'

Characters preceded by an underscore character ("_") are called shortcuts or accelerators. The shortcut is used as a means to access directly a menu entry instead of selecting it via the mouse or cursor. The accelerators need to be translated in order to fit with an Arabic keyboard. That means that you will have to choose a letter from the word (or the sentence) as the accelerator. Try not to assign the same letter for two accelerators which are in the same menu (you can guess that from the location of the 'msgid' for example). As a rule of thumb, assign the first letter of the word to be the accelerator, if that letter is already used as shortcut in the current context, assign the second letter, etc. Do please try to be consistent.

Names and Acronyms within 'msgid'

Names (people, locations, some programs) can be transliterated to Arabic characters, others can even be translated totally or partly to Arabic. Acronyms like ASCII, HTTP, C++ etc should be kept in Latin characters. This is the same for program directives, class names, variables etc. Remember to read the Arabic Documentation Standards [http://www.arabeyes.org/download/documents/misc/doc_standards-en/]. When in doubt, post on the 'doc' list.

Special Characters within 'msgid'

Some special characters like "&" need to be entered twice to tell the parser to keep it as is. Please trans-

late and take this in account when needed.

Sample:

```
#: rc.cpp:19
msgid "Color && Animation"
msgstr "##### # #####"
```

Use of the Imperative Form for Actions

Menu actions like "Edit", "Save", "Quit" etc and other related cases must be translated to Arabic in the imperative form. This is done for a standardization purpose (follow with the existing) and in order to give a more lively user/machine interaction. This is of EXTREME importance.

Sample:

```
#: ktouch.cpp:239
msgid "Save file..."
msgstr "#### #####..."
```

UTF-8 Encoding

Please remember to ALWAYS save your files in UTF-8 encoding ⁴. Otherwise, it can cause the file to be unreadable by others. UTF-8 is a global all-encompassing encoding (created and maintained by Unicode [<http://www.unicode.org>]). UTF-8 enables the user to encode all world's languages as well as various symbols (mathematical ones for instance) in a single file. Before Unicode and UTF-8, it was nearly impossible to exchange documents between localized systems. So it is *VERY* important that all work be saved in this UTF-8 encoding (posting Arabic messages to 'doc' should also only occur in UTF-8). KBabel, once configured properly (more on that later), will automatically save all your files in UTF-8. If you use a text editor to do your translations, on the other hand, please check that your editor of choice supports UTF-8 and that you are indeed saving your work in that encoding. A check can be done with the 'file' command:

```
$ file filename.po
```

This command will give you filename.po's encoding.

If you start to work on a file and you find that it isn't saved in UTF-8 or that it has an encoding problem (displays boxes instead of regular characters for example) immediately stop working on this file and report the issue/problem to the 'doc' list. Once the problem is reported and prior to resolution, do work on another file instead of waiting by idly :-)

Sanity checking (any errors ?)

In order to check that the 'PO' file that you've translated does not contain any syntactical/syntax errors, you need to,

⁴This document is UTF-8 encoded, set that encoding to view it properly.

```
$ msgfmt -c --statistics filename.po
```

'msgfmt' being a program from GNU's 'gettext' package.

If there are any errors, you will explicitly be warned about them with the appropriate line number(s) so you can easily locate and correct issues (missing double quotes ' ' are very common). Barring any errors, this "msgfmt" command will give you statistics about what's been translated, what's not translated and how many fuzzy strings are in the file. Please, report these statistics in the commit log (i.e. when committing the file(s) to the CVS repository).

Commit check-list

After translating an entire or part of a 'PO' file which you deem ought to be sent back to the repository, you need to commit the file(s) back to the repository. Here's a check-list of things you **MUST** do before committing a file:

1. The file has a '.po' extension.
2. You have entered the correct information in the file header.
3. The file is saved and is encoded with UTF-8.
4. You checked that there are no errors with the 'msgfmt' command.

If one or more of these requirements are not fulfilled, then please refer to what have been discussed above in order to remedy any problems. Otherwise, your files are eligible to be put on CVS :-)

If you had worked and translated several files at once, do please do a 'cvs commit' in order to send all those files grouped rather than committing them separately (saves bandwidth, less headaches, etc). While committing, and this is **VERY IMPORTANT**, remember to put the files' translation statistics in the commit log (how many strings on a per file basis were translated, untranslated and fuzzy). In case of a conflict (very unlikely), please refer to the CVS HOWTO.

Translation Tools

This section will present the various common tools that you may use to assist you in your translation process. While we recommend and encourage the use of tools like KBabel, since they will ease spectacularly your work, it is also possible and plausible to use simple text editors. Make sure you have read the Common Guidelines thoroughly before starting your translation work.

Text Editors

If you read the Common Guidelines carefully and thoroughly, than you can begin translating through the use a UTF-8 capable text editor. However, we recommend that you consider some of the tools noted below since they will ease your process and increase your enjoyment of work at hand. A text editor, unlike a translation application will require you to pay special attention and care to those pesky double quotes ' ' (they're often deleted by mistake), you'll be more likely to make syntax errors, you won't have a database to assist you and you will be unable to make a so-called "rough" translations. In short, consider a translation application and evaluate tools such as KBabel or gTranslator (their whole purpose in life is to ease translator's work :-)

If you are faced with the fact that you are only able to use a text editor, then make ensure that you text

editor is UTF-8 capable. We recommend that you use the latest version of the Arabeyes' text editor Katoob [<http://www.arabeyes.org/project.php?proj=Katoob>]. All other UTF-8 friendly text editor are acceptable as well.

On the Windows platform, we recommend the use of BabelPad [<http://uk.geocities.com/BabelStone1357/Software/BabelPad.html>].

KBabel

If you have KDE, you should already have KBabel installed. Otherwise, you can download the latest stable release from the KBabel Website [<http://i18n.kde.org/tools/kbabel/>].

Personalization

When you run KBabel for the first time, it will open the Preferences-KBabel dialog. You will need to fill-in some fields as this will ease the process and you will not be required to enter that info into the various 'PO' headers. If the Preferences dialog didn't get initiated, then you can open this dialog via Settings->Configure KBabel . Look at KBabel's Figure to see what kind of information you will need to enter in this dialog. Check that *all* your options look like those in the screenshots (except for your personal info of course: Name, email, etc).

Figure 1. KBabel configuration

You are still required to edit the 'PO' header *before* you save any newly opened file, in order to add the copyright info and move the last translator's name to the translators list (if it is not already there) and put your info in the translators list too. Do this from the Edit->Edit Header menu. All the other fields of the header will be filled automatically by KBabel, if you configured it correctly.

Creating a Database

Creating the database (also sometimes called a "dictionary") is important, as this will permit you to make "rough" translations. The database will also enable KBabel to propose to you possible translations for various 'msgid' strings (hints if you will). When you are to build your database, it is recommended that you have the greatest number of 'PO' files you can have (all those from the Arabeyes' CVS repository concerning KDE and/or GNOME for instance) so that you can build a rich and full database.

Go to Settings->Configure Dictionary->Translation database. Choose the "Database" tab and click on the "Scan directory and subdirectories" button. Choose the directory containing the 'PO' files and click on the "OK" button. The operation will take about 10 minutes. Click on the "Search" tab and mimic what is noted in this figure. Then select Settings->Configure KBabel and check the "Automatically start search" checkbox. You will also want to choose "Translation database" in the "Default Dictionary" pull-down list.

Figure 2. Database automatic search

Translating

KBabel creates two windows, the upper one contains the 'msgid' and the lower one is where you will enter your translations (KBabel figure 3). Take into account everything that was said about punctuation, the "_" and "=" characters, plural forms, etc. You will only be able to forget about the double quotes that surround the messages since KBabel assigns them automatically. If you are not sure about the trans-

lation you just made for a message, you can mark it "fuzzy" by typing "#, fuzzy" (without the quotes) in the "Comment" window on the right side of the "msgid" window. If you make a mistake as part of the punctuation or by adding an unneeded accelerator for example, KBabel will mark the string as "faulty". This is very useful since it will warn you about any ongoing mistakes and will reduce the possibility of faults. If the 'msgid' is empty, the string is automatically marked "untranslated". As soon as you type any character in that window, the string gets marked "translated". Do please double-check the correctness of your translation before moving on to next entries.

Figure 3. KBabel interface

Rough Translation

In order to use rough translations, you will need to have created a database as was noted earlier. Rough translation is a means for you to save time by having KBabel do the re-typing and copy-n-pasting of strings that already reside in the database (or other 'PO' files). Rough translation will also propose strings for the 'msgid' strings which are NOT in the database as well. In other words, it will try to do some guessing for you.

Once you have created the translation database, you can invoke rough translation from Tools->Rough translation... menu. Before pushing the "Start" button, make sure the dialog looks like the one in here, so that newly translated strings get marked as "fuzzy". This will force you and others to review those rough ("guessed") translations and to correct any possible errors or rewordings.

Figure 4. Rough translation

gTranslator

If you are a GNOME user or you're unwilling to use KBabel or just want to try something new, try gTranslator :-). If you don't have it already installed, you can download the latest stable version from the gTranslator Website [<http://www.gtranslator.org/>].

poEdit

On the Windows platform, we recommend the use of poEdit [<http://poedit.sf.net>].

New Translation Projects and Arabic Documentation

It is of utter importance that we encourage and set forth a plan by which we realize more translations of various existing Open Source documents to not only encourage more Arab users to consider Linux and Open Source but to educate them of its benefits. We are in dire need of a systematic manner in which the many existing Open Source documents get translated to Arabic. This importance and urgency stems from the fact this work needs to be done in the Arabeyes scope and under its umbrella since Arabeyes will ensure these resulting translations existence in one location in very similar formats. This "central" location of these translated documents will lift a heavy burden off of the Arab Open Source enthusiast in that it will not force him/her to search the entire web for bits of information here and there. One of Arabeyes' many missions is to centralize all the Arabic knowledge about Open Source and as such our collective means will be our strength.

The required procedure with regard to proposing new translation projects and how best to proceed as well as how to go about writing Arabic documentation is described in the First Steps. We are very eager to hear about your thoughts and comments !!

New Translation Projects

For the time being and until current translation projects get staffed to reasonable levels, Arabeyes is option to stick with KDE and GNOME. The reason those two environments were selected was simple - we wanted to reach the maximum number of potential Arab Linux users and enable them to be affective regular/daily Linux users. However, we realize that other aspects of the open source world may have the same impact and that some effort should be invested in other areas. So, if you are of that opinion and you have existing applications/programs, documentation or projects that you believe ought to be looked into and contribute to in their translation to Arabic, feel free to note them and we'll gladly discuss the matter. Arabeyes is here to help and help we will.

Again, please share with us your thoughts/ideas/comments !!

Writing Arabic Documents

The open source scene is unfortunately lacking Arabic documentation. Arabeyes is trying to fill this gap by documenting in Arabic several aspects with regards to the *n*x world. This documentation is *very* important since it will permit others to learn from what Arabeyes members has gone through and experienced.

Examples of what have been achieved so far include, Arabic LyX HOWTO [<http://www.arabeyes.org/download/documents/howto/lyx-howto-ar.pdf>] and the Arabic CVS HOWTO [<http://www.arabeyes.org/download/documents/howto/cvs-howto-ar/>]. Unfortunately this is a very time consuming process and it requires many man-hours to do correctly.

If you think you have sufficient knowledge about anything related to the open source world: Linux/Unix, programming (C/C++, python, HTML, etc - the list is long :-)) or that you don't have enough knowledge on a specific topic that interests which you'd be willing to a abit about or browse the web about, then please let us know. We could instruct you on how to compile a list of the necessary information which you could mold into an Arabic HOWTO. Your participation is much needed and is required *now* !! Let us hear about your intentions and we will be very happy to help you and to encourage you and of course to host these documents on Arabeyes website.

What Coordinators Do (Duties, Tasks)

Here is a general guideline or description of what a translation coordinator's job entails. The coordinator will have to:

- Assess current and future capabilities of the people present and their commitment level (know what volunteers will accomplish and make a distinction between the guy/gal putting 30-hours aweek vs. the one putting 2-hours). This assessment is to be revisited at least once a month.
- Set a plan in motion detailing how the intended work will be completed with the current man-power (or lack thereof). The plan is to be revisited and refined once a month.
 1. The plan will note priority of files (which ones need to be translated first).
 2. The plan will divide up the work so that if/when people volunteer they can hit the ground running and know what to tackle and what not to touch.

3. The plan will also note milestones that should be met to account for the project deadlines (ie. KDE-3.2 or GNOME-2.4 releases, etc).
 4. The plan will conclude with a schedule of tasks (broadly). This has to be reported on the corresponding task list on Arabeyes website, so that people know who is working on what and thus newcomers can choose easily a task that fit them.
- Keep the respective webpages (on Arabeyes.org) upto date with regards to info and priorities as well as status.
 - Status will be sent to the 'doc' list once a month (exact date is project specific). The status file will be of the following format (to ease reading),

Done past month (Feb 2003 - Mar 2003):

 - Did this.
 - Did that.
Todo next month (Mar 2003 - Apr 2003):

 - Finish this.
 - Start on that.
Issues/Concerns/Deadlines:

 - Need a script for this.
 - Need CVS account for such-n-such person.
 - Deadline: KDE release on May 6, 2003.
 - To publicly recruit for their respective projects. The coordinators main tasks beyond planning, executing and formulating a path to success will be to staff their teams. This means the coordinators will be asked to recruit people to help them (this means they'll be expected to post solicitations on various sites for help and for people's assistance). Coordinators are also responsible for motivating their staff and making life interesting for them :-)