



First, use `git add` to add all important files. Then remove everything that is not a source file:

```
$ git clean -df
```

Now you will have to mark all strings that should be translated. C source files containing such strings will need to include this header file:

```
#include <glib/gi18n.h>
```

For marking, replace for example the string "My Text" with `_("My Text")`. If you want to make additional contextual information available to the translators, you need to use other markings; see the explanations in the GLib Reference.

.ui files are marked like this:

```
<property name="..." translatable="yes">My Text</property>
```

Additionally, the `main()` function needs code to initialize Gettext so it uses the translations. Therefore we need to declare a configuration file in `meson.build`. The configuration file shall set a `GETTEXT_PACKAGE` constant.

```
cdata = configuration_data()
cdata.set_quoted('DATADIR',
  join_paths(get_option('prefix'), get_option('datadir')))
cdata.set('GETTEXT_PACKAGE', 'mycopyshop')
configure_file(output : 'config.h', configuration : cdata)
```

In `main.c` the following must be included:

```
#include <locale.h>
#include <config.h>
```

Then the following function calls must be made. This code can be found in the GLib Reference as well.

```
setlocale (LC_ALL, "");
bindtextdomain (GETTEXT_PACKAGE, DATADIR "/locale");
bind_textdomain_codeset (GETTEXT_PACKAGE, "UTF-8");
textdomain (GETTEXT_PACKAGE);
```

Now Meson needs to transform the markings into files with the not yet translated texts. For these we create a `po` directory in the main project directory and include it in `meson.build` by calling `subdir('po')`.

We create a file `po/meson.build` in it:

```
i18n = import('i18n')
i18n.gettext('mycopyshop', preset : 'glib')
```

In `po/POTFILES` all files containing strings to be translated should be listed like this:

```
# List of source files which contain
# translatable strings.
src/cshopapplication.c
src/cshoppraw.c
src/cshopwindow.c
src/main.c
src/ui/menu.ui
src/ui/toolbar.ui
```

This concludes the project's internationalization. For the German localization, a line `de` needs to be added to the file `po/LINGUAS`. English needs a line `en`, U.S. English needs `en_US` etc.

The final `po/LINGUAS` looks like this:

```
de
en
```

Its lines should be sorted alphabetically. Now a foundation for the translation can be built.

```
$ mkdir builddir
$ cd builddir
$ meson ..
$ ninja mycopyshop-pot
```

Copyright information and "Report-Msgid-Bugs-To:" should be added with a text editor in the header of the just created `po/mycopyshop.pot` file.

`msginit -l en --no-translator` or `msginit -l de --no-translator` then creates the corresponding files in the `po` directory in which the translations can be added with e.g. `gtranslator` (both German and English have "nplurals=2; plural=(n != 1);" according to the Gettext manual). All new files other than `builddir` should be added to the Git repository with `git add`.

Rebuilding the project now creates the translated version as well. It can be run with e.g. `LANGUAGE=en_US.UTF-8 LC_ALL=en_US.UTF-8 copyshop`.