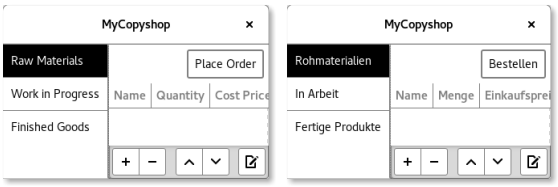

First, add use `git add` to add all important files. Then remove everything that is not a source file:

```
$ 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. Additionally, the `main()` function needs code to initialize Gettext so it uses the translations. This code can be found in the GLib Reference as well.

`.ui` files are marked like this:

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

Once everything is marked, you should go back to the main project directory using `cd` and then execute the following command there:

```
$ gettextize
```

This command adapts `configure.ac` and `Makefile.am` to generate and use translations. It also prints what changes one may want to make manually. Downloading files with `wget` via HTTP is not necessary and generally not a good idea. Adding `gettext.h` is not necessary because we are using GLib. `aclocal` and `autoconf` is already being executed by `./autogen.sh` and therefore need not be used.

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

```
# List of source files which contain
# translatable strings.
src/cshopapplication.c
src/cshopraw.c
src/cshopwindow.c
src/main.c
```

```
src/ui/menu.ui
src/ui/toolbar.ui
```

In `Makevars` this option may need to be set if the source files may contain non-ASCII characters:

```
# These options get passed to xgettext.
XGETTEXT_OPTIONS = --keyword=_ --keyword=N_ --from-code=UTF-8
```

Some compilers have problems interpreting source code with these characters though, so one may want to avoid them.

`configure.ac` should also contain the following:

```
AC_SUBST([GETTEXT_PACKAGE], [$PACKAGE_TARNAME])
AC_DEFINE_UNQUOTED([GETTEXT_PACKAGE],
    ["$GETTEXT_PACKAGE"], [Package name for gettext])
```

`main.c` will need to include this:

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

Once all important files have been added to the project with `git add`, the usual commands can be executed:

```
$ ./autogen.sh # instead of aclocal and autoconf
$ ./configure
$ make
```

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 the projekt can be rebuilt. `msginit` generates the respective files in the `po` directory which may be edited e.g. using `gtranslator` to specify the translations.

Rebuilding the project will now create the translated version as well. It can be executed by entering:

```
$ LANG=en_US.UTF-8 copyshop
```