



Zunächst sollten alle wichtigen Dateien mit `git add` aufgenommen werden und anschließend alles gelöscht werden, was keine Quelldatei ist:

```
$ git clean -df
```

Nun müssen zu übersetzende Zeichenketten markiert werden. C-Code-Quelldateien, die Zeichenketten enthalten, welche übersetzt werden müssen, müssen dazu diesen Header einbinden:

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

Zum Markieren wird z.B. "My Text" ersetzt durch `_("My Text")`. Wenn zudem Kontextinformationen für die Übersetzer mitgegeben werden sollen, dann können andere Markierungen verwendet werden; diese sind in der GLib-Referenz erklärt. Ebenso muss die `main()`-Funktion nun Code aufrufen, um die Übersetzungen zu benutzen; dieser ist auch in der GLib-Referenz zu finden.

.ui-Dateien werden so markiert:

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

Wenn alles markiert ist, sollte man sich mit `cd` in das Grundverzeichnis des Projekts begeben und dort folgenden Befehl ausführen:

```
$ gettextize
```

Dieser passt `configure.ac` und `Makefile.am` so an, dass eine Übersetzung generiert und genutzt wird. Er gibt auch aus, was man manuell ändern wollen könnte. Das Herunterladen von Dateien mit `wget` über HTTP ist allerdings nicht notwendig und auch nicht zu empfehlen. Das Hinzufügen von `gettext.h` ist nicht nötig mit GLib und auch nicht zu empfehlen. `aclocal` und `autoconf` wird schon von `./autogen.sh` ausgeführt, muss also nicht benutzt werden.

In der Datei `po/POTFILES.in` sollten alle Dateien mit zu übersetzenden Zeichenketten aufgelistet werden, zum Beispiel so:

```
# 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 muss u.U. folgende Option gesetzt werden, wenn die Quelldateien Nicht-ASCII-Zeichen beinhalten können:

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

Manche Compiler haben aber ohnehin Probleme mit Quelldateien mit solchen Zeichen, deswegen will man sie vielleicht ohnehin vermeiden.

In `configure.ac` sollte außerdem folgendes stehen:

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

In `main.c` muss folgendes eingebunden werden:

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

Wenn alle wichtigen Dateien mit `git add` eingebunden sind, kann wie üblich folgendes ausgeführt werden:

```
$ ./autogen.sh # statt aclocal und autoconf
$ ./configure
$ make
```

Damit ist die Internationalisierung des Projekts abgeschlossen. Zum Lokalisieren ins Deutsche muss in die Datei `po/LINGUAS` eine Zeile `de` eingefügt werden, für Englisch eine Zeile `en`, U.S.-Englisch `en_US` u.s.w.

Am Ende sieht `po/LINGUAS` so aus:

```
de
en
```

Die Zeilen sollten alphabetisch sortiert sein. Nun kann das Projekt neu gebaut werden. `msginit` erzeugt dann entsprechende Dateien im `po`-Ordner, wo z.B. mit `gtranslator` die Übersetzung eingetragen werden kann.

Neubauen des Projekts erzeugt nun auch die übersetzte Fassung. Diese kann ausgeführt werden z.B. wie:

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