



We want to use GNU Autotools so that we can easily build, test and distribute bigger projects.

For now, you create a new, empty project directory with one subdirectory called `src` or `mycopyshop` or another appropriate name. It should contain for example an empty file `main.c`. Later, you will create all source files for your project in this directory.

Now you should open a terminal in the main project directory and run the following commands:

```
$ git init
$ git add src/main.c
$ autoscan
```

Using `git`, you can easily manage which files are part of the project. All foreign files can be quickly removed using `git clean -df`. `git status` shows the status of the files in the project directory.

`autoscan` generates a GNU Autoconf file `configure.scan`. You should fill in the project name etc. in the file. Afterwards you should rename the finished file to `configure.ac`. `autoscan` can also be executed later on to show possible improvements for `configure.ac`.

Typically, a file called `autogen.sh` for enabling the project to be configured. It should contain something similar to this:

```
#!/bin/sh
cd $(dirname $0)
autoreconf --force --install --verbose -Wall
```

The following commands can be used to mark the file as executable and then to execute it:

```
$ chmod +x autogen.sh
$ ./autogen.sh
```

Now the project can be configured automatically, i.e. it can easily adapt to the user's operating system and their desired configuration of e.g. the C

compiler. However it cannot be automatically built yet. This is where GNU Automake can help:

```
$ automake --add-missing
```

The following should be added to `configure.ac` to make Automake usable and to allow Automake to build in subdirectories:

```
AC_CONFIG_FILES([Makefile])
AM_INIT_AUTOMAKE([-Wall -Werror -Wno-portability
                 subdir-objects])
```

Together with `autogen.sh`, these are all instructions needed to run the `Makefile.am` which you will need to create now.

To specify that a program called `copyshop` should be built from `src/main.c` and `src/cshopapp.c`, the following code should be inserted into a new file `Makefile.am`. Note that indentation should be done using tabs, not spaces.

```
AM_CPPFLAGS = -DDATADIR=\"$(datadir)\"
bin_PROGRAMS = src/copyshop
src_copyshop_SOURCES = \
src/main.c \
src/cshopapplication.c \
src/cshopapplication.h \
src/cshopwindow.c \
src/cshopwindow.h
src_copyshop_LDADD = $(GTK_LIBS)
src_copyshop_CFLAGS = \
-I$(top_srcdir)/ \
$(GTK_CFLAGS)
dist_data_DATA = \
src/ui/menu.ui
```

The following code in `configure.ac` defines the variables `GTK_LIBS` and `GTK_CFLAGS` needed by the `Makefile.am`:

```
# Checks for libraries.
PKG_CHECK_MODULES([GTK],[gtk+-3.0 >= 3.16])
```

The configuration can be accessed from C using:

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

Now you can configure, compile and install as well as uninstall the program with the default options plus warnings:

```
$ ./autogen.sh
$ CFLAGS="-Wall" CPPFLAGS="-Wall" ./configure
$ make
# make install
# make uninstall
```

The following command generates a file containing all source code (“tarball”) for distribution:

```
$ make dist
```

For serious projects, the manuals of the GNU Autotools should be searched for further recommendations:

- <https://www.gnu.org/software/autoconf/manual/autoconf.html>
- <https://www.gnu.org/software/automake/manual/automake.html>