

# Compile

## From x11-basic

This page describes how to **compile** X11-Basic for different platforms. The sources are found in the file X11Basic-1.20.tar.gz.

### Contents

- 1 Linux
  - 1.1 32bit
  - 1.2 64bit
    - 1.2.1 Debian Wheezy Installation auf amd64 mit dem gcc 4.7.1
  - 1.3 ARM linux
- 2 MS WINDOWS
- 3 other

## Linux

In order to compile X11-Basic, you must install the following development packages.

The actual name of these development packages depends on your distribution, so please refer to the distribution specific pages above for more details.

### 32bit

In order to compile X11Basic under linux or UNIX, you will need the following:

- A C compiler, preferably GNU C or tcc (but some ANSI C compilers, like DEC's will do)
- X11/Xorg libraries
- the readline library
- the lapack library (optional)
- the gcrypt library (optional)
- optionally the alsasound library

To build it, install the following packages (assuming you use a debian linux system):

```
sudo apt-get install libx11-dev libreadline-dev libc6-dev
```

These will suffice to get you started.

Unpack X11Basic-1.20.tar.gz with

```
tar xzf X11Basic-1.20.tar.gz
```

go into the X11Basic-1.20 directory and do a

```
./configure  
make  
sudo make install
```

That's all you will have to do (on a 32bit system) (for more detailed installation instructions read the file INSTALL, which comes with the package).

If a library or a development package is missing, then you will be warned by ./configure that some components are disabled.

If you want x11basic and its components installed in /usr instead of /usr/local do a

```
./configure --prefix=/usr
```

If the `configure' script fails, please contact me (kollo@users.sourceforge.net) and send me the output it generated (config.log). I am going to try to help you to fix the problem.

## 64bit

### Debian Wheezy Installation auf amd64 mit dem gcc 4.7.1

System ist eine Debian Wheezy Installation auf amd64 mit dem gcc 4.7.1. Tatsächlich ist make dann durchgelaufen und ich konnte xbasic ausführen.

## ARM linux

like the Raspberry PI ([http://en.wikipedia.org/wiki/Raspberry\\_Pi](http://en.wikipedia.org/wiki/Raspberry_Pi)) .

One can use two strategies to compile X11-basic for the Raspberry PI

1. use an ARM crosscompiler
2. set up the full development environment including gcc on the PI

In the latter case: To build it, install the following packages:

```
sudo apt-get install libreadline-dev tcsh libx11-dev libc6-dev libtool
```

After ./configure ; make, do:

```
sudo mkdir -p /usr/local/share/man/man1
make -n install | grep libx | sudo sh
sudo make install
```

The second line pre-installs the libraries so that also xbc will build.

For the graphics on the Raspberry pi, there is an interesting option. If you do not want to install and use the X11-libraries, you can compile a framebuffer-version of X11-basic with:

```
make fb
```

A file xbasic.framebuffer will be created. You can execute this and the graphics goes directly to the screen without X-Windows or such. At the moment, the Mouse-Input does not work and Keyboard input has to be done via stdin, but for purly graphics display applications this is already great, because this way, very little resources need to be used/installed onto the Pi except for X11-Basic of course.



Using X11-Basic on the Raspberry Pi without X11

## MS WINDOWS

The WINDOWS Version can be crosscompiled using mingw32 on linux. The make directives are included in the Makefile, so you could do a simple

```
make windows
```

## other

If the standard make dies not work, you should try a minimal compilation first:

```
make static
```

Should only produce one file: xbasic, the interpreter, statically linked. This is the less complicated case and should be fixed first.

Retrieved from "<https://sourceforge.net/apps/mediawiki/x11-basic/index.php?title=Compile>"

- This page was last modified on 27 April 2014, at 16:56.