

Building Bforartists with Ubuntu 17

Downloading the repository.....	1
Downloading and building the dependencies.....	2
Cmake.....	3
Building.....	4
Desktop shortcut and start menu entry.....	5
Updating the repository.....	7

In this short tutorial we will go through the build process under Linux Ubuntu 17. I will show one working way. But will not go into too much detail. I am no Linux user. I prefer the graphical UI where ever available. You can do all the steps completely at console too.

I have unfortunately just a german Ubuntu here. Sorry for the german screenshots.

Downloading the repository

We need to download the repository. This can be done in the console. First we make a directory. Then we tell GIT to clone the repository into this new created directory. Note that i am SU at the moment. You might need to sudo ...

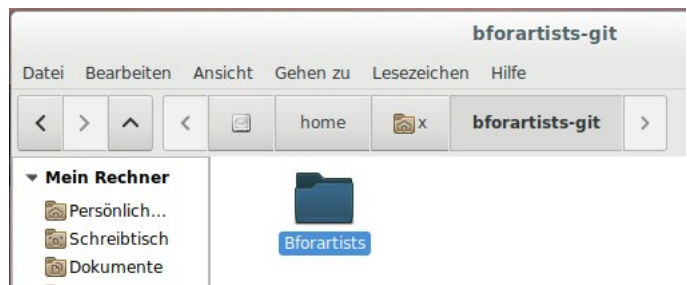
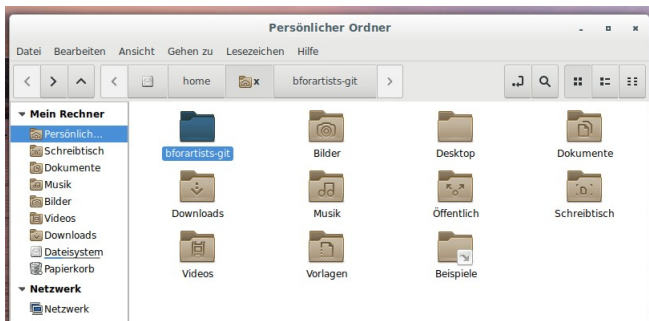
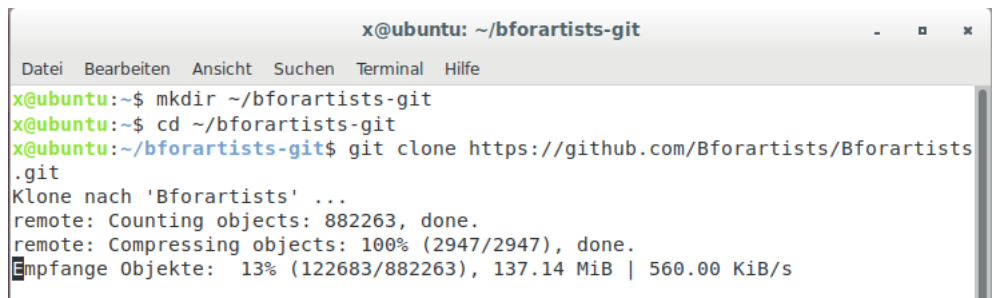
The commands are:

```
mkdir ~/bforartists-git
cd ~/bforartists-git

git clone https://github.com/Bforartists/Bforartists.git
```

Bforartists does not have submodules like Blender! So we don't need to care about them.

So we have now created the directory bforartists-git in the Home directory of your username.



And inside you can now find the repository folder with the code.

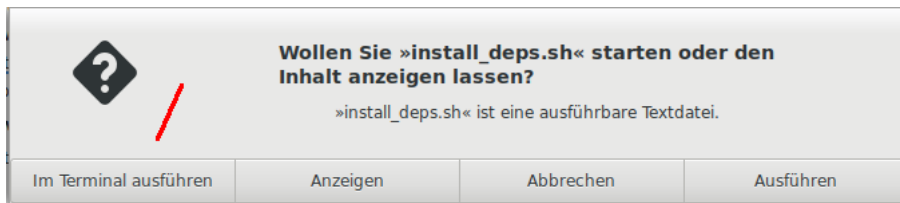
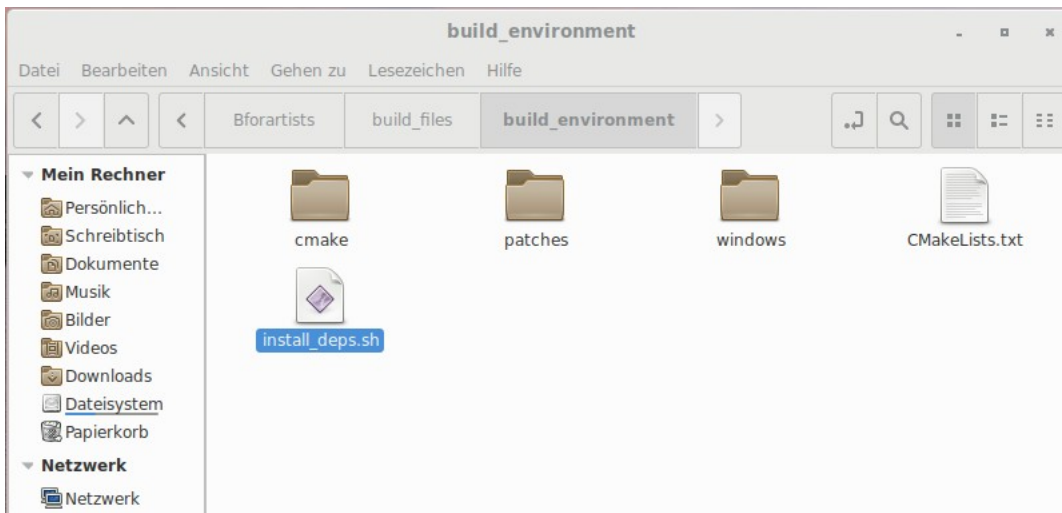
Downloading and building the dependencies

The next thing we need is the dependencies. Bforartists uses the very same dependencies than Blender.

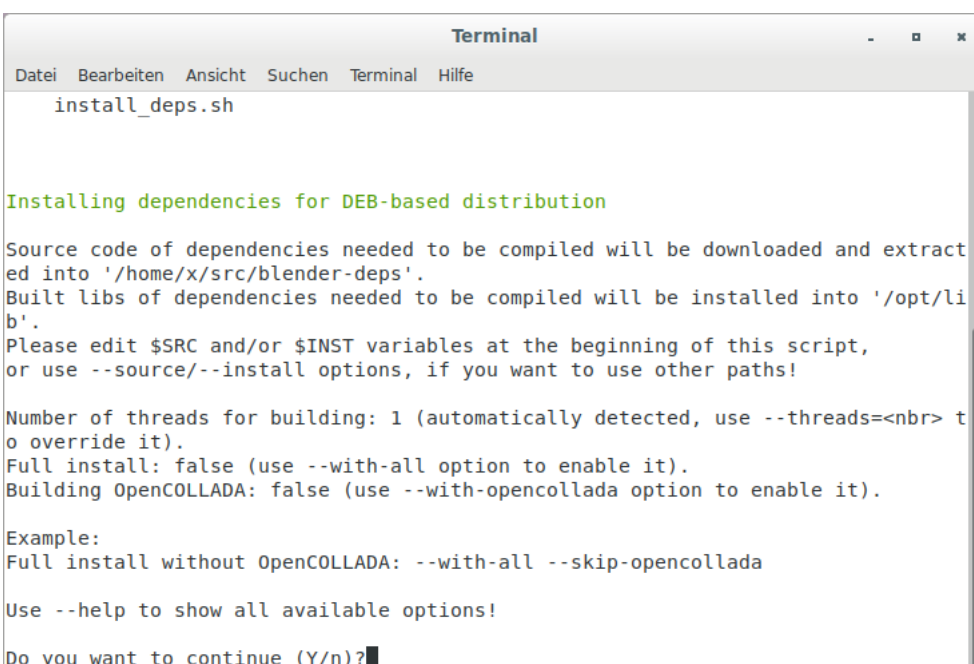
There is a script in the repository, called `install_deps.sh`. It is located in

```
./Bforartists/build_files/build_environment/
```

Run this script. Either from the console. Or by simply double clicking at it and click at run from console.



In the console you will be asked if you want to continue. Click yes.



The process to download, to compile and to install the dependencies may take a pretty while. Bforartists requires quite a few things.

Note

You can of course also install all dependencies manually. That's this monster term in the console then ...

```
git, build-essential, libx11-dev, libxi-dev, libsndfile1-dev, libopenexr-dev, libopenjpeg-dev, libpng-dev, libjpeg-dev, libopenal-dev, libalut-dev, python3-dev, libglul-mesa-dev, libSDL-dev, libfreetype6-dev, libtiff-dev, libavdevice-dev, libavformat-dev, libavutil-dev, libavcodec-dev, libswscale-dev, libx264-dev, libxvidcore-dev, libmp3lame-dev, libspnav-dev, python3
```

Beware, the python versions may change over time. So you may need to adjust them ...

install_deps.sh options

This script accepts some optional command line arguments (use --help one to get an exhaustive list), among which:

--source <path>

Where to store downloaded sources for libraries we have to build (defaults to ~/src/blender-deps).

--install <path>

Where to install the libraries we have to build (defaults to /opt/lib).

--with-all

Include some extra libraries that are by default not considered mandatory (main effect of this one is building OpenCollada).

Cmake

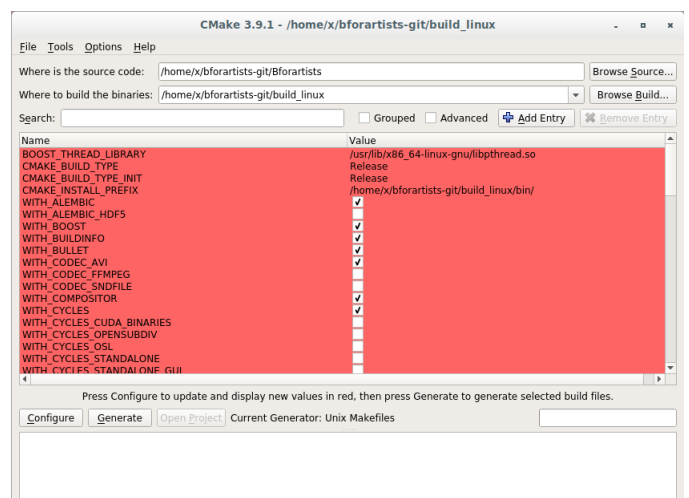
You need Cmake to configure the compiling process. So if not installed already, install Cmake.

Cmake also allows you to change some options manually. But this step is not required for building! This is for advanced usage, and will not be covered here.

The important part is that Cmake is installed. It gets used in the background.

But in case you want to do it manually, the different configurations can be found in the repository in build_files\cmake\config

The blender_release.cmake comes with all settings from the official build. To make it short, that's how we currently compile the official versions:



```

WITH_BUILDINFO      OFF
WITH_CODEC_SNDFILE  ON
WITH_CYCLES_CUDA_BINARIES ON
WITH_CYCLES_DEVICE_OPTIX ON
WITH_MOD_OCEANSIM   ON
WITH_OPENIMAGEDENOISE      ON
WITH_OPENVDB        ON
WITH_OPENVDB_BLOSC   ON

```

Building

We have downloaded the repository, we have downloaded and builded the dependencies. Now it's time to build Bforartists.

Open the path to the repository folder in the terminal.

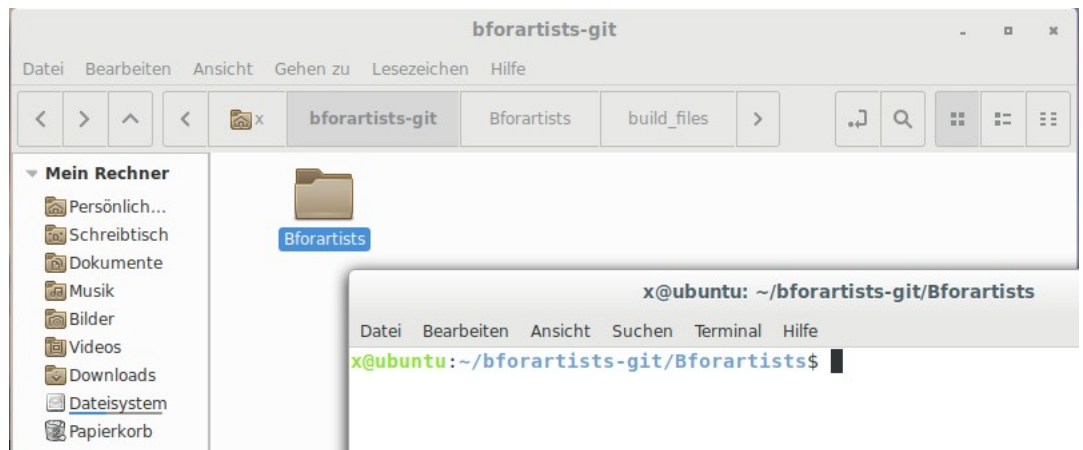
You can of course do all the configuration steps manually. Bforartists comes already with a preconfigured make file.

So open the repository path in the console. Then type:

```
make
```

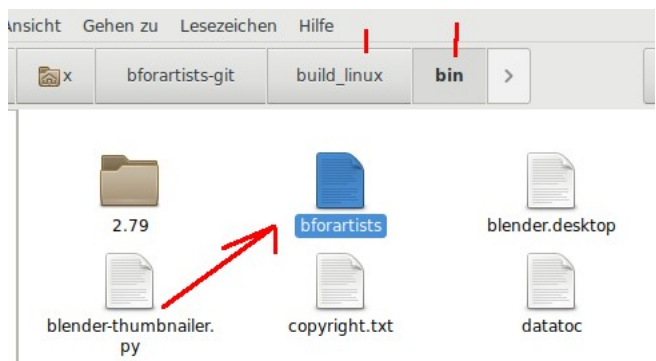
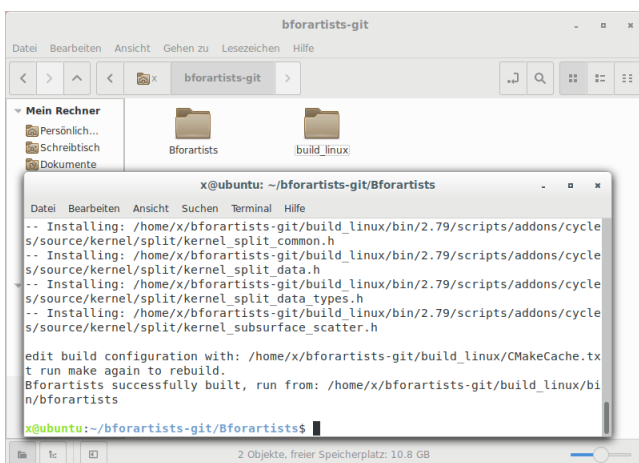
Or

```
make install
```

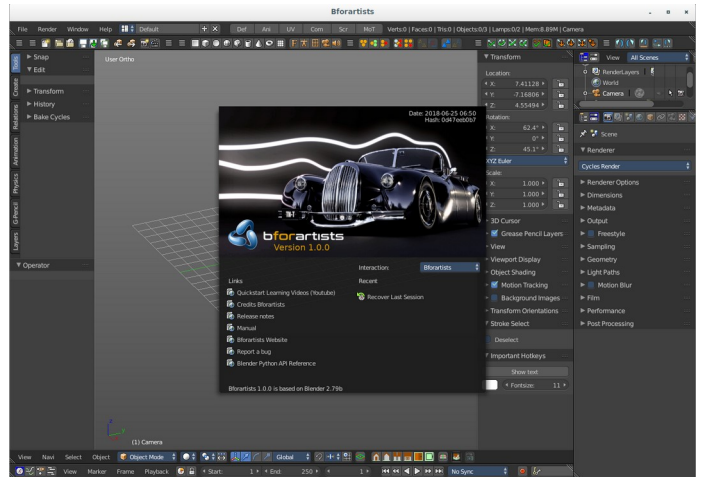
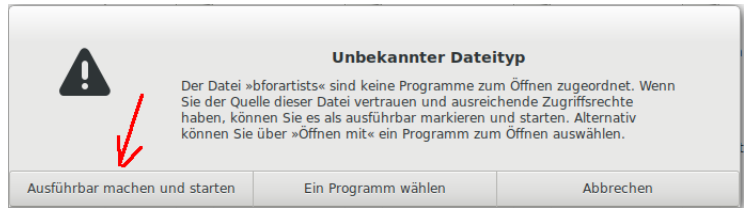


This will start the compiling process.

When it's done you will have a folder called build_linux. Inside this folder you will find a folder called Bin. And that's where you can find the bforartists executable now.



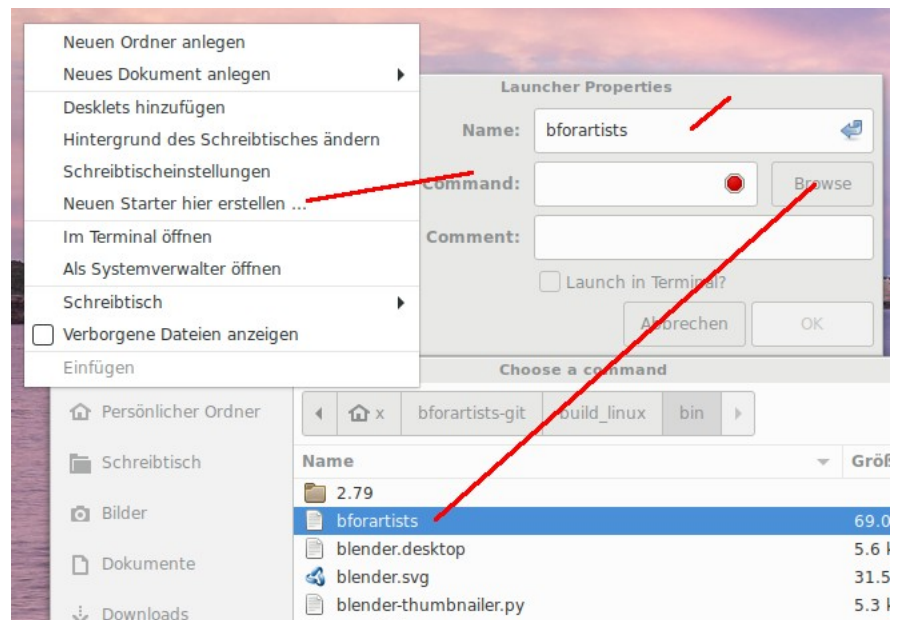
Double click at bforartists, allow to make it executable, and Bforartists should start.



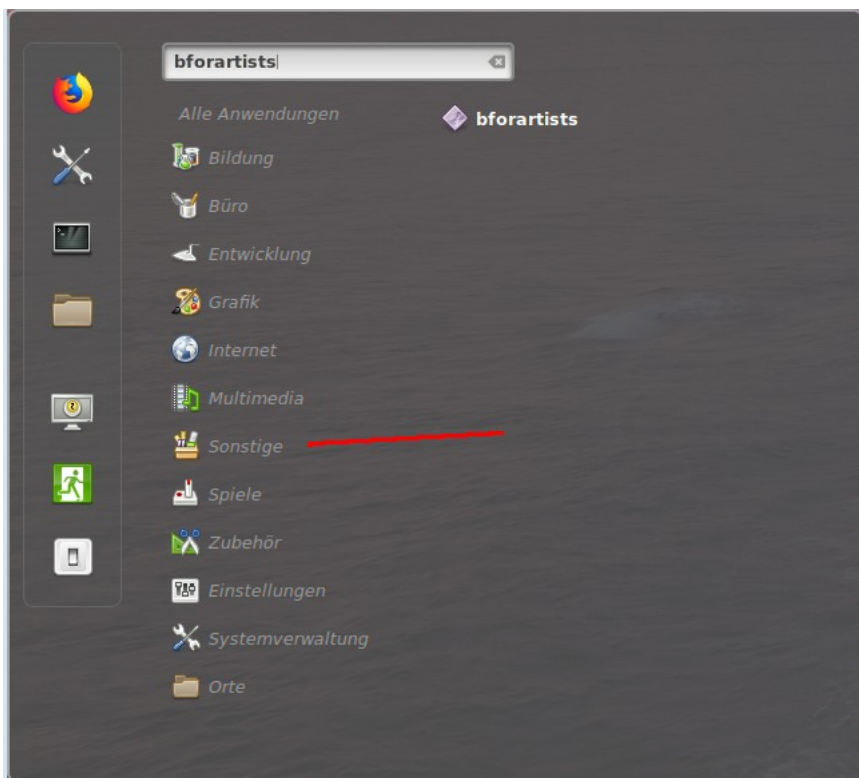
Desktop shortcut and start menu entry

When you have created the build with "make" and not with "make install" then the desktop icon is missing.

You can create a starter at the desktop. Right click at the desktop, choose create new starter here, type in the name, and in the Command field navigate to the bforartists executable.



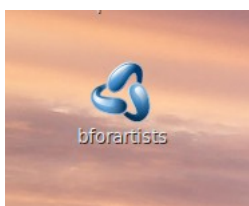
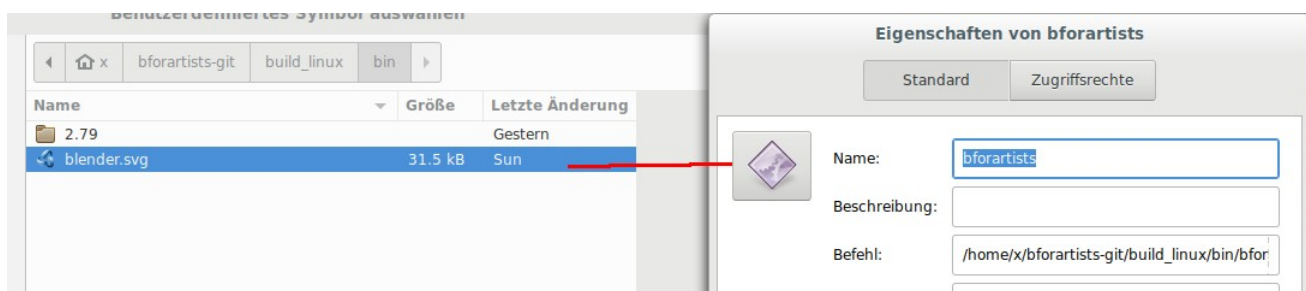
You will get a dialog if you want to add it to the start menu too. Click yes, and bforartists will be in the start menu too.



Note

The console term `sudo make install` in the `build_linux` folder will do the same, it will place the installer in the start menu. But will not create a shortcut at the desktop.

Finally you could add an icon to the desktop icon. Right click at the desktop icon, click at the icon, and navigate to the `blender.svg` in the `build_linux` folder. This is the Bforartists icon file. Choose it and click OK.



Updating the repository

To update your repository open the console, navigate to the bforartists-git/Bforartists folder, and type.

```
make update
```

This checks for the newest changes in the official repository and in the dependencies and downloads it. Therefore there should not be a need to rerun `install-deps.sh` when you rebuild Bforartists. `make update` should update the svn files too.

And it creates the new files in the `build_linux` file, and builds the new binary.