# How to download and install StumpWM Ubuntu

The easiest route would be:

```
sudo aptitude install clisp
sudo aptitude install stumpwm
```

The following description applies only if you want to install a more recent version of it.

## Prerequisites

```
sudo aptitude install git-core
sudo aptitude install autoconf
sudo aptitude install build-essential
sudo aptitude install texinfo
```

## Installing Common Lisp

You should first decide which Common Lisp (from now on, CL) implementation you want to use:

- CLisp (2.44.1+) or
- SBCL (1.0.24+)

and then install the version packaged with Ubuntu or their latest release. I would recommend using the latest versions because earlier issues have been resolved.

My advice is to use SBCL (with `cl-clx-sbcl`, as you definitely need to install CLX support before building StumpWM).

### SBCL with CLX

Normally, you could just do:

```
sudo aptitude install sbcl sbcl-doc
sudo aptitude install cl-clx-sbcl
sudo aptitude install cl-ppcre
```

To install CLX, start `sudo sbcl` and use `asdf-install` to download and install Lisp packages:

```
(require 'asdf)
(require 'asdf-install)
(asdf-install:install 'clx)
(asdf-install:install 'cl-ppcre)
```

When prompted "GPG warns that the key id [...] is not fully trusted", choose "[SKIP-GPG-CHECK] Don't check GPG signature for this package".

```
(quit)
```

### CLisp

Get CLisp from CVS:

```
cvs -d:pserver:anonymous@clisp.cvs.sourceforge.net:/cvsroot/clisp login
cvs -z3 -d:pserver:anonymous@clisp.cvs.sourceforge.net:/cvsroot/clisp co -d clisp-cvs -P clisp
```

or download the latest stable release of CLisp (2.48, as of 2009-07-28).

Switch to the directory you want your CLisp directory to be downloaded into (`/usr/local/src`, meant for source files that you downloaded yourself) and type:

```
wget ftp://ftp.gnu.org/pub/gnu/clisp/latest/clisp-2.48.tar.gz
tar -xvzf clisp-2.48.tar.gz
cd clisp-2.48
```

Install a few libraries on which CLISP relies:

GNU libsigsegv
    Library for handling page faults in a portable way
GNU libffcall
    Foreign Function Call Libraries

- Install libffcall

```
mkdir tools; cd tools
wget http://www.haible.de/bruno/gnu/ffcall-1.10.tar.gz
```

```
tar xfz ffcall-1.10.tar.gz
cd ffcall-1.10
./configure
make
make check
sudo make install
cd ../..
```

- Install libsigsegv

```
cd tools; prefix=`pwd`/i686-pc-linux-gnu
wget http://ftp.gnu.org/pub/gnu/libsigsegv/libsigsegv-2.6.tar.gz
tar xfz libsigsegv-2.6.tar.gz
cd libsigsegv-2.6
./configure --prefix=${prefix} && make && make check && make install
cd ../..

rm -f src/config.cache
```

### Install the XLIB development libraries

They're just packages you have to install. On Ubuntu, you can install `xorg-dev` and get what you need.

```
sudo aptitude install xorg-dev
```

Also, you have to make sure CLisp is compiled with`new-clx` support (`mit-clx` is not supported).

Otherwise, you'll get the error "there is no package with name `XLIB'" when building StumpWM.

```
# cd tools; prefix=`pwd`/i686-pc-linux-gnu
# cd ..
./configure --with-libsigsegv-prefix=${prefix} --with-module="clx/new-clx"
cd src
vi config.lisp
ulimit -s 16384   # increase default stack size
make
make check
sudo make install
cd ../..
```

Once the software has been installed, you can find further information about using `clisp` with these commands:

```
man clisp
apropos clisp
which clisp
locate clisp
```

And, finally, install CL-PPCRE from your package manager:

```
sudo aptitude install cl-ppcre
```

### Other approach

The other approach is to install version 2.44.1, by typing:

```
sudo aptitude install clisp
```

## Installing StumpWM

### Fetch StumpWM

Get StumpWM from Git. Type:

```
git clone git://git.savannah.nongnu.org/stumpwm.git
```

### Configure StumpWM

Now `cd` to your `stumpwm` dir and type `autoconf` to get an executable `configure` script:

```
cd stumpwm
autoconf
```

### Build StumpWM

Then type (as normal user):

- If you're building with CLisp:

```
./configure --with-lisp=clisp --with-ppcre=/usr/share/common-lisp/source/cl-ppcre
sudo make
make       #! this one succeeds (after a first failure)
```

- If you're building with SBCL:

```
./configure
make
```

#### Install StumpWM

Copy the compiled binary in your `PATH` somewhere (e.g. `/usr/local/bin`) by typing:

```
sudo install <full-path-to-stumpwm>/stumpwm /usr/local/bin/
```

```
sudo cp <full-path-to-stumpwm>/contrib/stumpish /usr/local/bin/
```

### Debugging

clisp -K full -x "(load \"asdf.lisp\") (load \"stumpwm.asd\") (load \"/usr/share/common-lisp/systems/cl-ppcre.asd\") (asdf:operate 'asdf:load-op :stumpwm) (stumpwm::stumpwm)"

You can run these expressions in the repl. Maybe then you'll get the debugger. The goal is to get a backtrace for the error.

### Installing StumpWM as the default window manager

Add a new entry to GDM Sessions menu, so that we can easily switch between StumpWM and Gnome.

To do so, copy any of the `*.desktop` files in`/usr/share/xsessions` to`/usr/share/xsessions/stumpwm.desktop` and edit it to look like this:

```
[Desktop Entry]
Encoding=UTF-8
Name=StumpWM
Comment=This session logs you into StumpWM (a minimalistic window manager)
Exec=/usr/local/bin/stumpwm
Icon=
Type=Application
```

Making StumpWM the *default session* will update your`.dmrc` file:

```
[Desktop]
Session=stumpwm
```

Logging out of StumpWM can take you to Gnome, and conversely, by choosing the appropriate session.

## StumpWM manual

### How to get Network Manager working with StumpWM

StumpWM does not have a system tray. And Network Manager needs one to be controlled. So, you need to install one. No further discussion.

I use `stalonetray` (instead of `gnome-panel`) which works great.

Currently, the system tray is put in the middle of the screen. It is not docked to the bottom of the screen, and it is overlapped by other windows, but that's not a real problem — to the contrary, in fact, as I save some space for the other windows.

### How to change keyboard language in StumpWM

To switch to another keyboard layout without restarting the X server, the easy answer is to use `setxkbmap` (for example, `setxkbmap fr` or `setxkbmap us`).

You can bind the two layout switching commands in your `.stumpwmrc`. That's what I've done with the Caps Lock key.

And to get a visual feedback of the current layout, I use a program called numlockx that turn on the Num Lock led — what has no negative side effect. The led is off in the default keyboard layout, on for the other one.

### How to get new mail notification with StumpWM

For the email notification, a powerful customization is to set the email alert as being a blinking led of the keyboard.

I use `mail-notification` and the command `xset led 3` to turn on the Scroll Lock led when new mail arrives. When all mail has been read, the led is turned off with `xset -led 3`.

For a blinking led, use `blink -s -r 1` and `blink` to turn the led on and off… but I've found this to interact badly with my Caps Lock: the status of the Caps Lock led is changed whenever the Scroll Lock led blinks…

### How to get Impressive working fine

- Try it with fullscreen mode disabled (option `-f`)
- Try `-g 1280x1024` (or whatever your resolution is) for black margins

## Fonts

You could add to your X defaults a setting like this:

Xft.dpi: 96

or a value closer to your display's resolution. It might also help to run your X server with your display's resolution. Some X servers evaluate the contents of a ~/.xserverrc file.

## Print Screen

## Monitor

A handy little application is the xosview client, which provides load, CPU, memory and swap usage, disk I/O usage and activity, page swapping information, network activity, I/O activity, I/O rates, serial port status, and if APM is enabled, the battery level (such as for a laptop).

## Automount

Un environnement de bureau comme Gnome ou KDE fait aussi tourner ds tas de démons pour genre auto-monter les CD ou clés USB, et faire des tas de trucs.