

WeeWM - Wee Enhanced Environment

Fast and ultra light window manager for XFree
version 0.0.2 - May, 13 2004

FlashCode, Xahlexx and Bounga

This manual documents WeeWM, a fast and ultra light window manager for XFree.
This manual is part of WeeWM.

Copyright (C) 1991, 1992, 1993, 1994, 1997 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA.

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to process this file through TeX and print the results, provided the printed document carries copying permission notice identical to this one except for the removal of this paragraph (this paragraph not being relevant to the printed manual).

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Foundation.

You may contact the authors by e-mail:

- FlashCode: flashcode@flashtux.org
- Xahlexx: xahlexx@weeland.org
- Bounga: bounga@altern.org

Table of Contents

1	Introduction	1
1.1	About this documentation	1
1.2	About WeeWM	1
1.2.1	What is WeeWM ?	1
1.2.2	Features	1
1.2.3	Command line options	2
1.2.4	Ask questions and find help	2
2	Installation	3
2.1	Prerequisites	3
2.2	Installation from sources	3
2.2.1	Getting source code	3
2.2.2	Compiling	3
2.3	Other packages	4
2.3.1	Debian package	4
2.3.2	Mandrake/Redhat package	4
2.3.3	BSD package	4
2.3.4	Gentoo package	4
2.3.5	Sorcerer package	5
2.3.6	Slackware package	5
2.4	Start WeeWM	5
3	Configuration	6
3.1	General options	6
3.1.1	A config.rc file example	7
3.2	Key bindings	7
3.2.1	A key.rc file example	9
3.3	Backgrounds and wallpapers	10
3.4	.xinitrc / .xsession	10
3.4.1	.xinitrc utility	10
3.4.2	Script syntax	11
4	Debugging	12
4.1	General informations	12
4.2	Informations to be specified	12
4.3	What about core-dumps ?	12
4.3.1	Compiling WeeWM with the debugging options	12
4.4	Report a bug	13
5	Authors / Support	14
5.1	Authors	14
5.2	Get support	14

1 Introduction

1.1 About this documentation

This is the WeeWM documentation, a window manager for XFree. It was generated from a .texi file (texinfo format) available with the WeeWM sources and in all the package available. We are using this format to make this documentation easily adaptable to the various output formats.

For the moment, Nicolas “Bounga” C. is the maintainer of this documentation. You can contact him at this address: bounga@altern.org. So feel free to contact him for all the question in touch with the WeeWM documentation. You can, as well, try to get in touch with him on [irc.freenode.net](https://www.irc.freenode.net), [#weewm](https://www.slack.com/channel/weewm) channel (Nick: Bounga). Nicolas maintains the english and french version of this documentation.

Contributions / suggestions: Obviously, we accept the suggestions about the documentations if this one is well write. Particularly about the subjects that are not handled here. As well, translations are welcome. For example, we need the most common languages (Italian, Spanish, German, etc). If you have any question on the way to help or if you have documentation that you want to give to us, you just have to send an email to the address given before.

1.2 About WeeWM

1.2.1 What is WeeWM ?

WeeWM is a window manager for XFree that is designed to be very fast and light. All can be done on the keyboard. Fully customisable, it handles the dock, the smart windows placement, the virtual desktops, and a lot more...

1.2.2 Features

Done:

- light windows decorations
- smart windows placement
- up to 256 virtual desktops
- all can be done with the keyboard
- with one key you’ve got a lot of associated command
- config file for bindings
- very light binary

Under development:

- smart dock management
- hot configuration reading

1.2.3 Command line options

WeeWM takes the following options from the command line:

```
weewm [ -cdfhklmov ] [long-options]
```

-c, --config

Choose alternate config file
(default: \$HOME/.weewm/config.rc)

-d, --display

Choose X display.

-f, --font

Choose X font (for messages, like window size when resizing).

-h, --help

Display the usage and exit.

-k, --keys

Choose alternate key bindings file
(default: \$HOME/.weewm/keys.rc)

-l, --licence

Display WeeWM licence and exit.

-m, --list-commands

Display WeeWM commands (for keys.rc) and exit.

-o, --list-options

Display WeeWM options (for config.rc) and exit.

-v, --version

Display WeeWM version and exit.

1.2.4 Ask questions and find help

Even if WeeWM is easy to use and to configure, it is possible that some questions or problems subsist. First of all, before coming on the mailing lists or on the WeeWM IRC channel to ask your question, don't forget to read the FAQ available with the sources.

2 Installation

A lot of WeeWM binary package are or will be available for the most common distributions. Those package should help you to install WeeWM. However, you might need for a reason or another to compile the source code. For example, the most recent version of WeeWM could not be available in binary package. You could also want to compile WeeWM with certain options that are not set in the binary packages.

This part of the documentation should allow to the beginners to be able to compile and to install the WeeWM window manager.

2.1 Prerequisites

You must have these programs correctly installed in order to run WeeWM:

- XFree, release 6 or newer (X11R6)

2.2 Installation from sources

2.2.1 Getting source code

You can download a source code package (recommended) or retrieve CVS base.

- Source package installation (**recommended**)

Download the source package, following on of these two links:

- <http://www.weewm.org/download.php>
- <http://savannah.nongnu.org/files/?group=weewm>

- Getting sources from CVS (**advanced users only**)

WARNING! CVS sources may be unstable and may not compile!

Create an empty directory “weewm” where you want, go into this directory and issue these commands:

```
cvs -d:pserver:anoncvs@subversions.gnu.org:/cvsroot/weewm login
cvs -z3 -d:pserver:anoncvs@subversions.gnu.org:/cvsroot/weewm co weewm
```

2.2.2 Compiling

To extract the archive, depending on the file you get, issue one of the two following commands:

```
tar xvzf weewm-x.y.z.tar.gz
tar xvjf weewm-x.y.z.tar.bz2
```

where x.y.z is the version (for example 0.1.2)

Then, go into the directory with the source code and issue this command:

```
make
```

So your source code is now compiled.

Then, as root user:
make install will install the files in the good directories.

2.3 Other packages

2.3.1 Debian package

Issue this command as root user:
dpkg -i weewm_x.y.z_i386.deb

You can create a .deb package yourself as well using the source code. To do this, you just have to go in the directory where is the source code and do:

```
debuild -rfakeroot -sd
```

A file named *weewm_x.y.z_i386.deb* should be available in the parent directory.

An apt repository is also available to you. To take advantage of this feature, you have to add the following line to your */etc/apt/sources.list* file:

```
deb http://apt.bounga.ath.cx/ unstable main
```

After what, as root, you can type the following commands:

```
apt-get update
```

```
apt-get install weewm
```

With this method, the upgrade is really easier. Indeed, when a new version of WeeWM will be out, you will just have to do:

```
apt-get update
```

```
apt-get upgrade
```

to upgrade to the new version.

2.3.2 Mandrake/Redhat package

Issue this command as root user:
rpm -i weewm_x.y.z-i386.rpm

2.3.3 BSD package

***** Not developed! *****

2.3.4 Gentoo package

***** Not developed! *****

2.3.5 Sorcerer package

***** Not developed! *****

2.3.6 Slackware package

***** Not developed! *****

2.4 Start WeeWM

Now that the installation has been successful, you have to start WeeWM.

There is generally two ways to start X (so WeeWM). The traditional way is to use the `startx` command. The other way is to use a graphical login display (as `gdm`, `xdm` or `kdm`).

If X is started with the traditional way (with `startx`), the important file is named `.xinitrc` and is located in your home directory. If you start X with a graphical login display, the file is `.Xsession` and is located in your home directory too.

The following step is to find the WeeWM binary. For most of the users, it's located in `/usr/bin/weewm`. Now, you have to edit (or create) the file mentioned before. You just have to add this line:

```
exec /usr/bin/weewm
```

Edit the path `/usr/bin/weewm` to put the good path of WeeWM on your system. If you don't find it, trying *type weewm* should give you the answer. When this step is done, you can save the file and close your editor.

If you start XFree doing *startx*, you have to make you `~/.xinitrc` executable, so you have to type:

```
chmod 700 ~/.xinitrc
```

If you're using a graphical login display, you don't have to do it.

You can now start WeeWM by doing:

```
startx
```


3 Configuration

3.1 General options

General options are stored in the `config.rc` file (default: `$HOME/.weewm/config.rc`). You can modify the available option values in this file to make WeeWM correspond to your wishes.

Detail of options:

<code>border</code>	width of the window borders (in pixels) min: 1, max: 3, default: 1
<code>space</code>	space between window and its border (in pixels) min: 0, max: 5, default: 1
<code>desktops</code>	number of desktops available min: 1, max: 256, default: 5
<code>minwidth</code>	minimum window width min: 1, max: 4096, default: 15
<code>minheight</code>	minimum window height min: 1, max: 4096, default: 15
<code>move_offset</code>	offset for moving windows (in pixels) min: 1, max: 256, default: 16
<code>color.border.active</code>	color of the active window border default: 'green'
<code>color.border.inactive</code>	color of the inactive window border default: 'black'
<code>color.sticky</code>	color of border for sticky windows default: 'blue'
<code>info.width</code>	width of info window - NOT DEVELOPED! min: 10, max: 4096, default: 100
<code>info.height</code>	height of info window - NOT DEVELOPED! min: 10, max: 4096, default: 50
<code>dock.use</code>	dock activation (1=use, 0=no dock) min: 0, max: 1, default: 1

dock.size
dock size (in pixels)
min: 8, max: 512, default: 64

dock.position
dock position (1=top, 2=left, 3=right, 4=bottom)
min: 1, max: 4, default: 3

dock.align
dock alignment (0=top/left, 1=centred, 2=bottom/right)
min: 0, max: 2, default: 0

3.1.1 A config.rc file example

Here is an extract from the config.rc file example given with the WeeWM package. You can find the full version in the `/usr/share/doc/weewm/examples/` directory.

```
border 1
space 1
desktops 5
minwidth 15
minheight 15
move_offset 16
color.border.active green
color.border.inactive black
color.sticky blue
info.width 100
info.height 50
dock.use 1
dock.size 64
dock.position 2
dock.align 0
```

3.2 Key bindings

Key bindings are stored in the keys.rc file (default: `$HOME/.weewm/keys.rc`). This file will allow you to choose which keys combination will launch a defined command.

Detail of key bindings:

All key names can be found in file
`/usr/include/X11/keysymdef.h` (key names without “XK_” prefix)

Modifiers are:

- “Mod1”: alt key
- “Mod4”: windows key
- “Control”: control key
- “Shift”: shift key

Detail of WeeWM internal commands for key bindings:

`next_win` focus next window

`prev_win` focus previous window

`pos_top_left`
move window to the top left corner

`pos_top` move window to the top of the screen

`pos_top_right`
move window to the top right corner

`pos_left` move window to the left of the screen

`pos_center`
center window on screen

`pos_right`
move window to the right of the screen

`pos_bottom_left`
move window to the bottom left corner

`pos_bottom`
move window to the bottom of the screen

`pos_bottom_right`
move window to the bottom right corner

`move_win_left`
move window horizontally (-X pixels)

`move_win_right`
move window horizontally (+X pixels)

`move_win_up`
move window vertically (-X pixels)

`move_win_down`
move window vertically (+X pixels)

`prev_desktop`
go to previous desktop

`next_desktop`
go to next desktop

`desktop[1-256]`
go to specified desktop

`send_prev_desktop`
send window to previous desktop

`send_next_desktop`
send window to next desktop

`win_max_horizontal`
maximise window horizontally

```

win_max_vertical    maximise window vertically
win_max             maximise window
win_fix             fix/unfix window
win_memo_pos        memorise position and size of window
win_memo_desktop    memorise desktop of window
win_forget_pos      forget memorised position for window
win_forget_desktop forget memorised desktop for window
win_kill            kill current windows
reconfigure         reload config - NOT DEVELOPED!
quit                quit WeeWM

```

3.2.1 A key.rc file example

Here is an extract from the example file given with the WeeWM package. You can find the full version in the `/usr/share/doc/weewm/examples/` directory.

```

Mod1|Control Left prev_desktop
Mod1|Control Right next_desktop
Mod4 Left send_prev_desktop
Mod4 Right send_next_desktop
Mod4 F1 desktop_1
Mod4 F2 desktop_2
Mod4|Shift Up pos_top
Mod4|Shift Down pos_bottom
Mod4|Shift Left pos_left
Mod4|Shift Right pos_right
Mod1|Control Escape quit
Mod4 Return run aterm 0 7913 xterm
Mod4 c run1 xclock 1 0 xclock

```

In this example, we have:

```
Mod1|Control Left prev_desktop
```

so using the keys Mod1 + Control (Mod1|Control) + left arrow key (Left) will show us the previous desktop (prev_desktop).

```
Mod4 Return run aterm 0 7913 xterm
```

will launch (“run” parameter) an aterm terminal to a given position.

If you're searching for a key name, you could use `xev` to find it. For example, you want to know the name of the “right arrow” key. So launch `xev`, go to the new window and press the “right arrow” key. You should see something like this in your terminal:

```
KeyPress event, serial 24, synthetic NO, window 0xa00001,
root 0x3e, subw 0x0, time 11617881, (122,71), root:(126,88),
state 0x0, keycode 102 (keysym 0xff53, Right), same_screen YES,
XLookupString gives 0 characters: ""
```

The interesting part is: `keycode 102 (keysym 0xff53, Right)` where we can see the keyword “Right”. So the name of the key we were searching for is “Right”.

3.3 Backgrounds and wallpapers

First of all, you have to know that the backgrounds and the wallpapers are not handled by WeeWM itself but by an external software. We will introduce here the most well-known and we will explain how to use it the basic way.

xsetroot:

`xsetroot` is a background software included in the XFree package. It can handle basic backgrounds like a plain color background. It can handle a bitmap image too. For more information about this software, see its manual (`man xsetroot`).

Esetroot:

`Esetroot` as for him is more dedicated to handle wallpapers and transparency. Indeed, if you load your wallpaper using `Esetroot` and that you're using the Eterm terminal emulator, you will be able to activate the transparency for this one. To display a wallpaper on the whole screen, you just have to do `Esetroot -f your_file.png`.

To automate the loading of your favourite wallpaper when you start WeeWM, you can for example add this line (if you don't use a graphical login display):

```
Esetroot -f /path/to/my/image.png
```

at the end of your `~/.xinitrc` file

3.4 .xinitrc / .xsession

3.4.1 .xinitrc utility

You need to know that a window manager is a software for X like mozilla, gimp or aterm. We can think that X starts a window manager and that's the softwares are launched by the window manager. However, in fact, the things aren't like this. It is XFree that launches the softwares and the window manager, as its name indicates it, does nothing more than managing the windows. You can, for example start a windows manager, launch your softwares, then kill the window manager and start another one, then continue to work.

When X is started, it reads your `.xinitrc` or `.xsession` script. Once that the whole of the orders of this file have finished their execution, X shuts down. And it's not the fact of quitting the window manager that makes X shut down as we can think.

3.4.2 Script syntax

To understand the following, you need to have some basic knowledges about the shell. For example, when you enter a command in a shell, you can't do another thing in this shell before that the first command has end-up. The scripts (and so `.xinitrc` or `.xsession`) has the same behaviour.

Obviously, there is a way to launch a command in a shell and to immediately regain the hand. This method will allow you to launch automatically some softwares that you want on the X startup. Those softwares must be put on the background. We can do that, adding a “&” at the end of the command line. Thus, if you want that `xclock` runs at every XFree starting, add this line to your `.xinitrc` :

```
xclock &
```

When all the commands that you want to automatically launch are added, you can go to the window manager starting step. To do this, we will use the `exec` command.

A `.xinitrc` example file could be :

```
# we start some softwares  
xmms &  
gkrellm -w &  
idesk &  
  
# then we start our window manager  
exec weewm
```

4 Debugging

4.1 General informations

The debugging process can appear as a reserved task for developer but you can easily acquire the basis. Two good addresses to learn to do a good bug report are : <http://www.mozilla.org/quality/bug-writing-guidelines.html> and <http://www.chiark.greenend.org.uk/~sgtatham/bugs.html>.

4.2 Informations to be specified

Informations that you must absolutely transmit to us when you write us a bug report bug are mainly :

- Your OS / distribution and its version.
- WeeWM version, and if it's a CVS (or development tarball), its date
- When does it happen ? Have you done something particular ? Can you reproduce it ?
- Your settings. It can be found in `~/.weewm/`

4.3 What about core-dumps ?

If a crash has happen while you were using WeeWM and that you've got a core-dump, you should come on `#weewm` channel on `irc.freenode.net` server and say to an op that you've got a problem and a core-dump. He should asks you some informations.

To get interesting informations using core-dumps, you must have a WeeWM version compiled with the appropriate options and that is not the case of the release versions that are offered on the website. So, you will have to compile WeeWM by yourself.

4.3.1 Compiling WeeWM with the debugging options

To compile WeeWM with the debugging capabilities, you will have to download the WeeWM code sources. When you will want to launch the `"make"` command, be sure to had the following options :

```
$ OPTIONS=-Wall -W -pedantic -pipe -O2 -g3 make
```

When the compilation is over, do this in your shell :

```
$ ulimit -c unlimited
```

it will enable the creation of core-dumps with unlimited size.

You can now start X and start the debugging. Try to reproduce the action that had crashed WeeWM the last time to get the core-dump. Then start the debugging :

```
$ gdb weewm core
```

Then type this command in gdb :

```
(gdb) where
```

A lot of messages should be printed. WeeWM developers will be involved by all the messages beginning with a #.

When you've got all of this, you can send an e-mail describing the way to crash WeeWM, the messages that you've just get and your WeeWM configuration files available in `~/weewm/`.

Please don't delete your core file and the sources directory of WeeWM immediately. The developers should ask you some other details and you will probably have to use this files. **Don't** send any file to the developers without having exposed your problem to them. You can do this on the IRC channel, or by IM (AIM, ICQ, Jabber) with one of them. Your problem could be specific to your system, so it will probably be useless for the developers to have your core file that is, in most case, really huge.

4.4 Report a bug

Before reporting a bug, be sure to check if the problem has not been already report. To know if it has been report you can look at the BUGS file in the last source package or on the CVS.

If you've find a new bug, please submit it to this webpage :
<http://savannah.nongnu.org/bugs/?func=addbug&group=weewm>

5 Authors / Support

5.1 Authors

WeeWM is written by:

- FlashCode:
 - *e-mail*: [flashcode AT flashtux DOT org](mailto:flashcode@flashtux.org)
 - *web*: <http://www.weewm.org>
 - *irc*: server: irc.freenode.net, channel: #weewm
 - *jabber*: flashcode@jabber.org
 - *icq*: 160677660
 - *aim*: FlashCode AIM
 - *yahoo*: FlashCode_Y
- Xahlexx:
 - *e-mail*: [xahlexx AT weeland DOT org](mailto:xahlexx@weeland.org)
 - *web*: <http://www.weeland.org>
 - *irc*: server: irc.freenode.net, channel: #weewm
- Bounga:
 - *e-mail*: [bounga AT altern DOT org](mailto:bounga@altern.org)
 - *web*: <http://bounga.ath.cx>
 - *irc*: server: irc.freenode.net, channel: #weewm
 - *jabber*: Bounga@jabber.org
 - *icq*: 178297842

5.2 Get support

Support with IRC:

- server: **irc.freenode.net**
- channel: **#weewm**

WeeWM forum: <http://forums.flashtux.org>

Support with mailing list: You can subscribe and use the users mailing list for weewm support:

- to subscribe: please visit <http://mail.nongnu.org/mailman/listinfo/weewm-support>
- to send a mail to all subscribers: send mail to weewm-support@nongnu.org

Or you can contact developers, See [Section 5.1 \[Authors\]](#), page 14.