

# A beginners guide to wmii

Steffen Liebergeld

February 25, 2006

## Contents

<b>1</b>	<b>What you need to know</b>	<b>3</b>
1.1	Wmii - the second generation of window manager improved . . . . .	3
1.2	What this document is all about . . . . .	3
1.3	Who should read this document, or what you need to know before . . . . .	3
<b>2</b>	<b>Configuring and installing wmii</b>	<b>3</b>
2.1	Obtaining wmii . . . . .	3
2.2	Configuring . . . . .	3
2.3	Install . . . . .	4
2.4	Make it fit your needs . . . . .	4
<b>3</b>	<b>Terminology</b>	<b>4</b>
3.1	Pages . . . . .	4
3.2	Frames . . . . .	4
3.3	Clients . . . . .	4
3.4	Layout . . . . .	5
3.5	Actions . . . . .	5
<b>4</b>	<b>Getting started</b>	<b>5</b>
4.1	Trying different layouts . . . . .	5
4.2	Tabbed frames . . . . .	6
<b>5</b>	<b>Further concepts</b>	<b>6</b>
5.1	Zoom . . . . .	6
5.2	Virtual workspaces . . . . .	6
5.3	Maximize . . . . .	6
5.4	How to move frames around . . . . .	7
5.5	To wrap it up . . . . .	7
5.6	Flashback . . . . .	7
<b>6</b>	<b>Looking under the hood</b>	<b>7</b>
6.1	Tools . . . . .	7
6.2	wmibar . . . . .	8
6.3	wmimenu . . . . .	8
6.4	wmikeys . . . . .	8
6.5	wmifs . . . . .	8

6.6 wmir . . . . .	8
<b>7 A tour through the ixpfs</b>	<b>8</b>
<b>8 Copyright notice</b>	<b>8</b>

# 1 What you need to know

## 1.1 Wmii - the second generation of window manager improved

wmii is a lightweight window manager for the X Window System. It provides a synthesis of conventional, tiled and tabbed window management based on layouts. Several roots of these window management capabilities have been introduced by the Ion and LarsWM window managers. Apart from this, it implements a socket-based filesaver, which is accessed to configure and interoperate with wmii. The idea behind this file-based approach is derived from the plan9 operating system and can be found in the Acme programming environment. wmii consists of the core window manager itself and several utilities, such as wmibar, wmifs, wmimenu, wmikeys, wmiplumb, wmir and wmiwarp.

## 1.2 What this document is all about

This document will teach you all you need to know to make a good living with wmii. You will learn how to configure and install wmii, and how to start using it. On the way you will get to know about certain design decisions. After reading this document, you will have fallen in love with wmii and be able to work with it much more effectively than you could before.

## 1.3 Who should read this document, or what you need to know before

This text is written for people, who already have lots of experience using Unix and who have quite good knowledge of some of the basic things in the X Window System. It is intended for people, who are eager to get to know new methods of interaction with their computer, and who are open minded with new technology. Maybe the term *Hacker* describes the audience of this document best.

# 2 Configuring and installing wmii

## 2.1 Obtaining wmii

Wmii is to be licensed under the terms of the MIT/X Consortium License. That means, everybody is allowed to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so. Please note, that the authors provide the software "as is", without any warranty of any kind.

Of course you can download the wmii sources from <http://wmi.modprobe.de>. For most major Linux distributions precompiled packages are available on the net already.

If you would like to get the bleeding-edge, always freshest sources, you can easily check out the darcs repository.

## 2.2 Configuring

The wmii sources come with a file called "config.mk", which contains all the things one might want to adjust prior to compilation.

The most important setting is the PREFIX, which tells where wmii is to be installed to. In this file you can also tell if wmii is to be built with cairo support. However cairo seems to be pretty unstable, so you are not encouraged to use it. You won't need it anyways.

## 2.3 Install

After you have configured `wmii`, you just have to issue `make` to build `wmii`.

Now you will be able to install `wmii` as superuser using the `make install` command.

After installation you should copy the files from `PREFIX/etc/wmii` to a local directory `.wmii`.

Note that `PREFIX` is `/usr/local/` in the default install.

To make `wmii` your default window manager, you add can write the following lines into your `/.xinit(` or `/.xsession` if you run a display manager like `xdm`):

```
#!/bin/sh
exec wmii
```

## 2.4 Make it fit your needs

The directory `/.wmii` contains a set of files, which allow you to set your preferences for `wmii`.

When `wmii` starts up, it executes a file called `rc` in your `.wmii` directory. Please note, that this can be any executable, be it a shell script or a binary. The `rc` program needs to know two parameters, `start` and `stop`. On `start` it has to set up the internal virtual filesystem of `wmii`.

It is important for you to know that `wmii` uses a new concept of inter process communication for set up and usage. The basic approach is to use a filesystem-alike representation of the data. This idea is taken from the Plan9 operation system and its ACME-programming environment.

Please note, that this internal filesystem will be mountable into a unix directory tree in the future, bringing you all the power of your standard shell tools to operate `wmii`.

For this time, we should stick to the standard configuration, which is already very powerfull. I will teach you the basics of file manipulation and the filesystem later on.

# 3 Terminology

## 3.1 Pages

The whole visible area of your monitor is called a page. This is important for you to remember, because you will be able to apply layouts only to full pages later on.

## 3.2 Frames

Frames are exactly what the term *suggests*: Frames are rectangular areas on pages, that contain X clients (e.g. an Xterm). In `Wmii`, default config, frames have a titlebar with a string identifying the contained client.

## 3.3 Clients

Any X program, or -so to say- any program that draws a window on a X Server is called a client. You will be able to do various manipulations on those clients later on. For now it is important for you to know that only one client at a time gets the users input. In our terminology this client is the "selected"<sup>1</sup> client.

---

<sup>1</sup>In the X Window System, this is called the *focussed Client*

## 3.4 Layout

A layout defines the behaviour of a page. It defines how frames are to be arranged on a page.

## 3.5 Actions

Actions are all the commands you can use to manipulate the window management. Maximizing a window or launching programs are actions, for example.

# 4 Getting started

Before you fire up `wmii` for the first time, I would like you to write down some of the commands on a sheet of paper. Of course you could start right in without, but chances are that you will get stuck very soon. So here is the list:

- `MOD+Tab` = cycle frames
- `MOD+shift+Tab` = cycle nested frames (tabbed frames)
- `MOD+ctrl+a` = open up actions menu
- `MOD+ctrl+p` = open program menu
- `MOD+t` = open a Xterm

I have to remind you, the `MOD` key is usually your `ALT` (left) key, but can be any other if you played with `xmodmap`.

The shipped default layout is *tiled*. So maybe it is a good start to open up a few Xterms now. You do this simply by pressing `MOD+t`. So after you opened 4 Xterms, you have quite a good playground for your first navigation experiments.

Did you notice that `wmii` automatically placed and sized the new windows on creation? This is a feature and should help you to even further free your mind.

Okay, type a few characters now. In one Xterm these characters should appear. This one is the *selected* client.

How can you move the selection<sup>2</sup> now? It is very easy: you just press `MOD+Tab` and you will be pointed to the next Xterm. Do this a few times to get a feeling for it.

## 4.1 Trying different layouts

*tiled* layout is already good and powerful, but `wmii` has some more layouts. It is now time to try them all. To do so, please press `MOD+shift+m` now. You will see that all the windows (frames) are now maximized on the page. Well, in fact you won't see that but only the one frame in front of you. This is what the *max* layout is all about: maximized frames.

Now press `MOD+Tab` again to cycle through the maximized frames. See how easy it is? In fact, the maximized layout is the one I use most since I like to use as much screenspace as possible. In fact it makes my life easier in one further aspect: it keeps all the other disturbances and dialogs out of my view so I can concentrate on my current work.

But there are still other layouts for you to try. Press `MOD+shift+g` to enter *grid* layout. Note that your Xterms now got aligned like those little rectangles on a chessboard. The philosophy with this is, that every frame should get as much screenspace as all the others. Note that you can now use `MOD+Tab` again to cycle through the frames. As you can see the titlebar of

---

<sup>2</sup>or focus

the focussed frame is highlighted in a different color than the others. This is useful when you are using nested frame - which you will learn about later on on.

Let us just try another layout. Press `MOD+shift+f` to enter the *float* layout. You will see that all the xterms are now floating without any order around on the page. This layout resembles the old fashioned window management of all the other boring window managers.

You are free to change back to your favourite layout now.

- `MOD+shift+m` = max layout
- `MOD+shift+t` = tiled layout
- `MOD+shift+g` = grid layout
- `MOD+shift+f` = float layout

## 4.2 Tabbed frames

Up to now we only had one client in one frame each. One could work with this already, but if you open up a lot of clients your workspace will get cluttered. To help avoid this `wmii` has a technique called *tabbed frames*. It enables you to put more than one client in one frame. The clients are still using all the space in the frame each, but they will be arranged in tabs -just like in your favourite browser-. The tabs will be made visible in the titlebar of the frame, which is then tiled in as many parts, as you have clients in that frame, each part resembling one client.

So far for the theory. Now lets try it out. Cycle to one xterm of your choice and press `MOD+u` to unlock the frame. All the clients you open up from now on will be placed in that frame. Press `MOD+t` to open up a Xterm. Note that this one is now in that old frame you are still in. The titlebar has split up in two. That ss good already, but how can you access the now hidden xterm? This is done with the nested cycling shortcut `MOD+shift+Tab`. Press it now to cycle trough the nested clients. If you press `MOD+Tab` again, you will see that you can still cycle through all the frames of the page just as you did before.

Now cycle back to the frame with the two xterms and press `MOD+shift+u` to lock the frame. If you open up windows now, they will be put in new frames as usual.

You have now learned two key concepts of `wmii`: *locking* and *unlocking* of frames.

## 5 Further concepts

### 5.1 Zoom

`Wmii` has a feature called *zoom*, which tries to rearrange the managed frames to give the focussed the most screenspace. This is the most helpful in tiled layout. You can invoke *zoom* by pressing `MOD+Enter`.

### 5.2 Virtual workspaces

You have only used one page now, but `wmii` offers you to have as muc of them as you want. You can create a new page by pressing `MOD+ctrl+y`. Feel free to cycle through all the pages by pressing `MOD+l`.

### 5.3 Maximize

Sometimes you want to escape your cluttered page and see only one frame maximized. Well this is possible by pressing `MOD+m`. `Wmii` will create a virtual page and show only that one

frame maximized on it. You can still get back to all the other pages with MOD+l. To get the maximized frame back into its previous place, just press MOD+m again.

## 5.4 How to move frames around

You may have asked yourself, how you could take an existing client and put it somewhere else, on another page or into another frame. In wmii this is done through detaching and attaching the frame. I will now guide you through that process:

First of all, you will have to cycle to the frame you want to be placed somewhere else. Once you are there, press MOD+d to detach the client. You will see, that the client instantly disappears. Stay calm, the clients still exists in wmii's memory.

You can now go to another frame, unlock it (MOD+u) and attach the detached client to it by issuing MOD+a. As you can see, the client is now part of that frame. You may now lock the frame again using MOD+shift+u.

You can also take a nested client and put it into a frame of its own. Just cycle to the client you want and detach it (MOD+d). Now make sure all frames are locked, so it won't get attached to any existing frame by accident and press MOD+a for attach. You will see, that the client is now on a frame of its own.

## 5.5 To wrap it up

- MOD+Enter = zoom a frame
- MOD+ctrl+y = create new page
- MOD+l = cycle pages
- MOD+m = maximize a client
- MOD+u = unlock a frame
- MOD+shift+u = lock a frame
- MOD+d = detach a client
- MOD+a = attach a previously detached client

## 5.6 Flashback

You might want to get a list of all the configured keystrokes of wmii. This can be easily generated using a tool called `wmirefcard`.

Congratulations. You have now learned all you need to know to work with wmii. In the next chapter I'll give you some further insights into the configuration interface of wmii.

# 6 Looking under the hood

## 6.1 Tools

Wmii consists of a number of tiny tools, each doing only one specific Task. Those tools are able to communicate with each other through a virtual filesystem. This design follows closely the main Unix rules - the *Rule of Modularity* and the *Rule of Composition*.

You may want to read about it in *The Art of Unix Programming*<sup>3</sup> by Eric S Raymond.

---

<sup>3</sup>The book *The Art of Unix Programming* by Eric S Raymond can be read online at <http://www.faqs.org/docs/artu/>

## 6.2 **wmibar**

Wmibar is a little tool that has the purpose of drawing a little bar with some labels on your screen.

## 6.3 **wmimenu**

Wmimenu is a tool that creates menus. It is highly configurable through its virtual filesystem, so that one can easily create his own menustructures and behaviours.

## 6.4 **wmikeys**

Wmikeys does all that has to do with keys. Basically it grabs the keys and executes actions. Again, this can be configured through its own filesystem.

## 6.5 **wmifs**

Wmifs has the purpose to bind all the different virtual filesystems of the wmii tools into one namespace, which acts like one virtual filesystem to the outside world. In other words, wmifs brings together all the interfaces of the wmii tools and let the look like one united interface.

## 6.6 **wmir**

Wmir is used to read and change the virtual filesystem exported by wmifs. You can use wmir to control virtually anything in wmii if only it has a representation in the wmifs.

# 7 **A tour through the ixpfs**

If you fire up `wmir read /` you will see the root node of the ixpfs. You can then just browse through this filesystem just like you would do in any other filesystem.

# 8 **Copyright notice**

Beginners guide to wmii Copyright (C) 2006 by Steffen Liebergeld

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.