

The Fluxbox Window Manager

Opening Pandora's Box

Speed is a family trait. Fluxbox inherits the best characteristics of its predecessor, Blackbox, and adds a few useful features to the desktop.

BY ANDREA MÜLLER

Programming skills are a fantastic thing. No matter if your favorite application is too plain, too garish, or lacks a few options, just a few lines of code will provide the functionality you need. This is the situation that Henrik Kinnunen found himself in just a while back. Blackbox [1] was his favorite window manager, but he needed to add a few things to be really satisfied. Instead of re-inventing the wheel, Henrik started to work on the Blackbox 0.61.1 source code, added a few convenient features, and dubbed his creation Fluxbox [2].

Obtaining Fluxbox

The quickest way to install Fluxbox is to use RPM packages, if available. The good news for Suse Linux Version 9.0 users – although Suse does not provide RPMs, the Mandrake packages will work just fine.

For once, users with older distributions have an advantage over users with newer versions. You cannot build Fluxbox with `gcc` version 3.3 or later. If you have Suse Linux version 8.2 or 9.0, or Mandrake Linux 9.2 or 10.0, refer to Box 1 to find out how you can build a new window manager with a modicum of manual steps.

Desktopia

Only you can decide how your desktop looks. With deskTOPia we regularly take you with us on a journey into the land of window managers and desktop environments, presenting the useful and the colorful viewers and pretty toys.



If you have `gcc` 3.2, or an older version (you can type `gcc -version` to discover your version), follow the usual steps to build and install Fluxbox:

```
./configure ; make ; su -c ↵
make install-strip
```

The window manager itself is a frugal character, and only requires the `XFree86` developer packages, `libpng`, and `freetype`. If you regularly use KDE applications, you might like to replace `./configure` with `./configure --enable-kde`. This tells Fluxbox to display KDE programs that

Box 1: Compiling Fluxbox with `gcc` 3.3 or later

If you attempt to compile Fluxbox with `gcc` 3.3 or later, the `make` command will issue an error message and quit when it attempts to compile `src/fluxbox.cc`. `utils/fbrun/FbRun.cc` below the source code directory contains another error.

To build a functional Fluxbox, call `./configure` as usual. The first change you need to make is to the `config.h` header file in the source code directory. This file contains rules for compiling Fluxbox. Open the file in your favorite editor, and look for the `STRFTIME` string. You should find a comment block, as indicated by `/*` and `*/`. Insert the following statement after the line

```
#define HAVE_STRFTIME 1
```

The second change affects `utils/fbrun/FbRun.cc`. You need to add another `#include` line, to allow the `fbrun` tool to compile. This points the compiler to the

file that contains the functions it was previously unable to locate at compilation. The list of files to add to the existing `#include` directives depends on your distribution. If you have Suse Linux, you need to add the following (see Figure 1):

```
#include <g++/cassert>
```

Mandrake stores the `cassert` file in a different directory. The correct statement for Mandrake is as follows:

```
#include <g++-3/cassert>
```

If you have neither of these distros, you will need to run `locate` or `find` to look for the `cassert` file. The `#include` line does not expect the absolute path, but the path relative to the `/usr/include` directory. So if the file is located below `/usr/include/wherever`, you would need to add the following:

```
#include <wherever/cassert>
```

```

andi@ragnarok:~/fluxbox-0.1.14
// DEALINGS IN THE SOFTWARE.
// $Id: FbRun.cc,v 1.8 2002/11/27 21:56:56 fluxgen Exp $
#include "FbRun.hh"
#include "App.hh"
#include "EventManager.hh"
#include <X11/Xlib.h>
#include <X11/keysym.h>
#include <X11/Xutil.h>
#include <unistd.h>
#include <iostream>
#include <fstream>
#include <g++/cassert>
37,22 7%

```

Figure 1: Add an #include line to FbRun.cc to compile Fluxbox with gcc 3.3.

typically slot into the kicker system area in the Slit, a bar for **WindowMaker dockapps**. These apps include the *klipper* clipboard manager, or *kgpg* which manages **GnuPG keys**.

Fluxbox Ready for Takeoff

After installing Fluxbox, the next step is to integrate the window manager into the GUI launch procedure. If you installed a package provided by your distributor, the new entry should be in **Login Manager** pull-down list.

If you installed Fluxbox manually, you will need to edit the configuration menu. For KDM, the KDE login program, launch the Control Center, and then go to *System administration | Login manager | Sessions*. The */usr/local/bin/fluxbox* entry below *New type* allows you to add the window manager to the selection list. To change the KDM settings, make sure that you click the *Administrator mode* button first, to work with the *root* account.

If you log on in text mode, create an entry as follows

```
exec /usr/local/bin/fluxbox
```

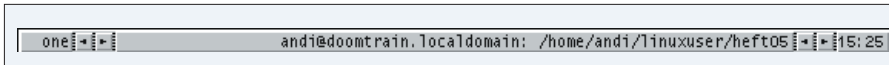


Figure 2: The Fluxbox toolbar with desktop selector, and clock.

in your *~/.xinitrc* file, to allow the *startx* command to launch Fluxbox.

First Steps

Fluxbox is quite straightforward when first launched. There is a toolbar with a clock and a desktop selector tool for a default of four virtual desktops (see Figure 2). Iconized programs are also displayed in this area. You can click the arrow buttons to access the next desktop, or if you are in a hurry, and have a wheel mouse, simply roll the wheel.

To modify the default settings, right click on the toolbar to open the configuration menu. The items below *Placement* define the position of the toolbar. You have a choice of six positions, from *Top Left*, through *Top Center*, to *Bottom right* (see Figure 3).

To prevent windows from hiding the tool bar, you might like to enable *Always on top*. The *Auto hide* option tells the tool bar to disappear if it is not in use, and reappear when you move the mouse cursor to the position normally occupied by the tool bar. You can use a combination of this option and *Always on top* to use the whole screen for application windows. Moving the mouse to the edge of the screen reinstates the tool bar.

To assign an intuitive name to the active desktop, thus allowing you to distinguish it more easily from others, select *Edit current workspace name*. The toolbar entry (*one, two* etc.) will toggle to this name when you select this entry.

My Programs

The **Context menu**, which has a useful selection of applications by default, facilitates program selection (see Figure 4).

The first two entries are for a terminal emulation and a browser. The programs hiding behind these entries will naturally depend on what you have installed on your system.

Fluxbox provides a *fluxbox-generate_menu* tool which searches through menus to find known applications. It saves any apps it discovers in the menu file, which is copied to *~/.fluxbox/menu* when you first launch the window manager. There is one application missing on initially launching Fluxbox, the *fbrun* applet, which pops up a quick start window for programs. This practical tool was not installed when *fluxbox-generate_menu* created the menu. You can type

```
fluxbox-generate_menu -p > /usr/local/share
```

to add *fbrun* to the top level of the menu. The *-p /usr/local/share* option is important if you built Fluxbox yourself. By default, the window manager will expect the directory with its settings and styles to be located below */usr/share*. If you forget to specify the path, Fluxbox will not have any **themes** on offer in the configuration menu.

If you are unhappy with your selection for the default command-line window and browser, you can use the *fluxbox-generate_menu -t* and *-b* parameters, followed by the desired program names to change the settings:

```
fluxbox-generate_menu -t > konsole -b konqueror -p > /usr/local/share
```

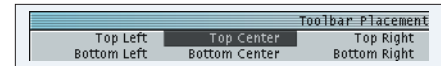


Figure 3: Setting the screen position for the toolbar.

GLOSSARY

gcc: The C compiler from the GNU Compiler Collection GCC. It translates human-readable program source code text into a machine-readable format, thus creating an executable from a text file.

WindowMaker Dockapps: Desktop applications [3] specially designed for the WindowMaker (<http://www.windowmaker.org/>) window manager. These applications run in the background and are accessible via

an icon in the toolbar, which is referred to as the dock.

GnuPG key: The "GNU Privacy Guard" is an application that securely encrypts email messages and other data, allowing decryption by authorized persons only. This protects email messages from spying eyes en route, just like an envelope protects snail mail. GnuPG keys [4] are required for encryption and decryption.

Login Manager: The program that prompts users to enter their usernames and passwords to access the GUI. The window manager will not be launched without these security credentials.

Context menu: A menu that appears when you right-click on a specific area within an application. The menu content changes to reflect the task performed by this part of the GUI.

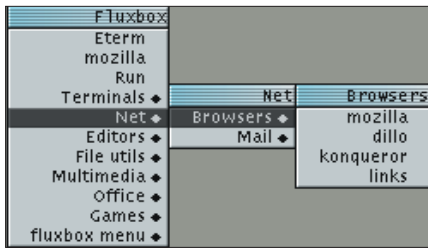


Figure 4: Fluxbox modifies the menu to reflect the system environment during the build process.

will add entries for the KDE command-line window, *konsole*, and the Konqueror browser to the top level of the menu.

Groups

Working with Fluxbox windows is similar to other window managers. A single click will move the focus to a window. The window title bars have a collection of well-known buttons which pin, iconize, maximize, and close the window (from left to right).

The small tabs at the top left of the window bar (see Figure 5) are a special feature that allow users to assign multiple applications to a single container. This works just like under *pwm* [5]. Click on a tab, and hold down the middle mouse button to drag the tab to the target window. Drop the tab in the window to assign a second application to that window. You can click a tab to access the application in the tab. To remove a tab from a window, simply drag it out of the window and drop it on the desktop.

Having to manually reposition applications that you want to assign to a single window can become bothersome. The tabs provide a useful shortcut. Right-click a tab to open the main menu with a list of applications. If you launch a program from the menu, it will not use a window of its own, but simply join the applications in the window from whose tab menu it was launched.

Not easy enough? If you want applications on a desktop to share a window, without the user having to set this up

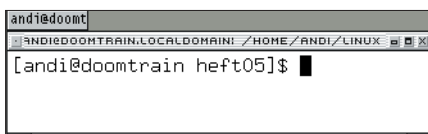


Figure 5: Tabs allow users to group windows on Fluxbox.

explicitly, you can create a *.fluxbox/groups* file in your home directory. Add the window class names of the applications that you want to run in shared windows, one line per group.

The *xprop* tool will tell you the group name for a program. When you type

```
xprop WM_CLASS
```

the mouse pointer changes into a crosshair. Now click the window whose class name you need to discover. The following output shows the class name for the HTML editor, Quanta:

```
WM_CLASS(STRING) = "quanta", "quanta"
```

and for Konqueror

```
WM_CLASS(STRING) = "konqueror", "konqueror"
```

Strangely, Mozilla caused *xprop* trouble, and led to a *class unknown* message. The following line in the *groups* file allows Konqueror and Quanta to automatically share a window:

```
quanta konqueror
```

Now you simply need to tell Fluxbox that the file contains the group definitions. The place to do this is *~/fluxbox/init*. The last line of the file

already reads *sessiongroupFile*: You just need to add the path to the *groups* file you just created:

```
sessiongroupFile: /home/andi/.fluxbox/groups
```

You can use the *fluxbox-menu | Reload Config* command in the Fluxbox menu to tell the window manager to reparse its configuration to apply the changes without rebooting.

Settings

The *fluxbox-menu* entry in the main menu is also the place to set the window manager's behavior. Fans of the sloppy focus model will want to use *Configure | Focus Model | Sloppy Focus* to tell Fluxbox to shift the focus to a window on mouse over. If you want to move windows that are activated in this way to the foreground, you will additionally need to check the *Auto Raise* checkbox.

The *Tab Placement* menu offers a selection of 16 positions for tabs. The most interesting entries are the ones that include the word *Relative*. There is a relative entry for each window border. Selecting *Left Relative* will anchor the tabs to the left-hand window border. If you group two applications, each tab will occupy 50 percent of the window's height.

If you prefer not to use tabs, you can disable *Use Tabs* to do without them.

Listing 1: Shortcuts for *~/fluxbox/keys*

```
01 # Use [Alt-Ctrl-Right arrow] a window
    and [Alt-Ctrl-Left arrow] 13 Mod1 m :MaximizeWindow
02 # to switch between desktops 14
03 Mod1 Control Right 15 # [Alt-s] reduce the window to
    :NextWorkspace its title bar
04 Mod1 Control Left 16 Mod1 s :ShadeWindow
    :PrevWorkspace 17
05 18 # [Alt-d] removes and restores
06 # [Alt-Ctrl-Enter] open an window decorations
    xterm command-line window 19 Mod1 d :ToggleDecor
07 Mod1 Control Return 20
    :ExecCommand xterm 21 # For Emacs fans: Fluxbox can
08 also handle keyboard
09 # [Alt-x] close the active 22 # shortcuts, such as [Alt-y]
    window [Alt-c] to open Mozilla
10 Mod1 x :Close 23 Mod1 y Mod1 c :ExecCommand
11 mozilla
12 # [Alt-m] maximize and restore
```

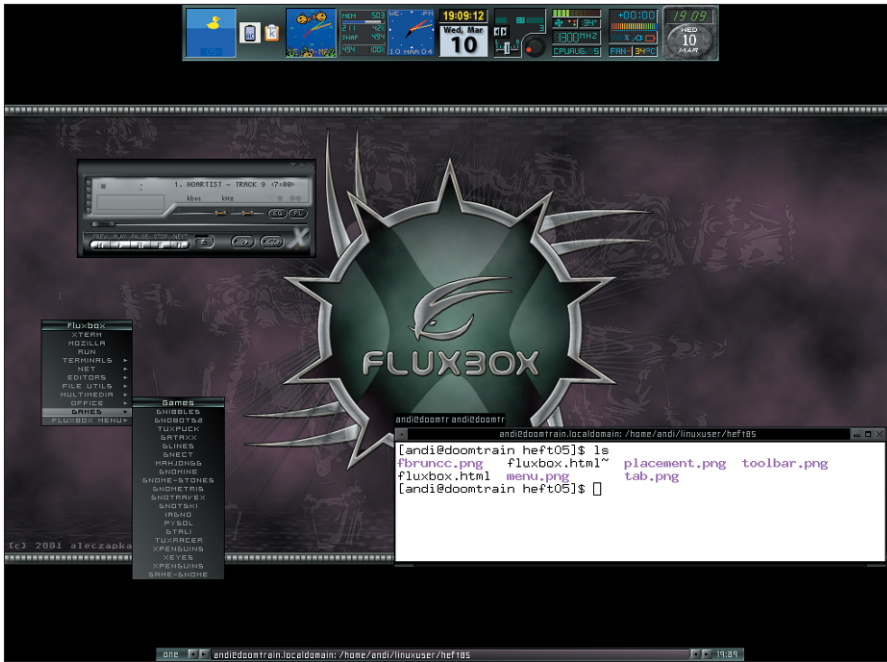


Figure 6: Adaptable thanks to styles. The window manager dressed in the Fluxarnation theme.

The *Opaque Window Moving* option is the right choice, if you want to view window content while moving windows.

Stylish

The entries below *fluxbox-menu | Styles* allow you to change the window manager’s appearance. Fluxbox has a standard collection of over 20 themes. Blackbox fans will enjoy the *Artwiz* theme. *Lemon Space* has a nice fresh feeling of Spring, and *Blue* is a futuristic theme. If you are unable to find a theme to your liking, check out [6] for a collection of really neat themes (see Figure 6).

You may need to create a directory called `~/.fluxbox/styles` for your themes, although some themes off the Web will take care of this automatically when you unpack them. If the theme has its own background image, you additionally need a `~/.fluxbox/backgrounds` directory. After creating a theme directory, you can use the *fluxbox-menu | System Styles* and *fluxbox-menu | User Styles* Fluxbox menu items.

Keyboard Wizardry

Keyboard fans will also enjoy working with Fluxbox. Only a few keyboard shortcuts are defined by default, for example [Alt-Tab] is used to switch windows, and [Alt-F1] through [Alt-F12] to switch desktops. Keyboard shortcuts are stored in the `~/.fluxbox/keys` file. You can use any editor to modify this file, assigning shortcuts to most of your major functions.

Each of these keyboard shortcuts needs to be in a separate line, just like the following one:

```
keyboard shortcut :action
```

The *keyboard shortcut* must comprise at least one modifier key such as [Alt] (the configuration file calls this *Mod1*), or *Control*, and one additional key. Refer to the *KEYS FILE* section in the manpage for Fluxbox to discover what actions the Fluxbox window manager recognizes. The annotated sample configuration in Listing 1 shows a few useful additions to the *keys* file.

GLOSSARY

Theme: A method of changing the general appearance of the desktop and the programs launched on it. Themes are like a collection of clothes for the window manager.
ACPI Monitor: A small monitoring program that can be used to display the battery load

state or the temperature of a laptop, for example. The device needs to support ACPI (“Advanced Configuration and Power Interface”) to do so, and will typically need a modern Linux kernel version, also configured to support ACPI.

Enter the Applets

We referred to the Slit, the bar that provides a home for WindowMaker dock-apps [3], Afterstep [7] and KDE applets, at the beginning of this article. We have not discussed it in detail thus far. The Slit is always there, waiting, usually in the bottom right-hand corner of the screen, for you to launch an appropriate application. For example, when you launch the system monitor, *bubblemon* [8], and the KDE **ACPI monitor**, *akpi* [9], both of these apps will happily move into the Slit as can be seen in Figure 6 to the left.

If you do not like the default position, as a vertical bar on the right, the drop-down menu for the Slit has an applet that allows you to change this. *Direction* allows you to choose between vertical and horizontal. *Placement* has the same choices as previously discussed for the tool bar, and allows you to set the screen position. The practical “*Auto hide*” and *Always on top* features are not available for the Slit

All this convenience makes Fluxbox more than just a good choice for fans of simple window managers. In fact, Fluxbox has enough potential to tempt even the biggest fans of KDE and Gnome.

INFO

- [1] Blackbox: <http://sourceforge.net/projects/blackboxwm/>
- [2] Fluxbox: <http://fluxbox.sourceforge.net/>
- [3] Dockapps: Joachim Moskalewski, “DockApps”, Linux Magazine Issue 3 <http://www.linux-magazine.com/issue/03/Dockapps.pdf>
- [4] GnuPG: Patricia Jung, “Key Signing Party”, Linux Magazine Issue 35, http://www.linux-magazine.com/issue/35/Using_GnuPG_Keys.pdf
- [5] PWM: <http://modeemi.cs.tut.fi/~tuomov/pwm/>
- [6] Fluxbox themes: <http://fluxbox.sourceforge.net/themes.php>
- [7] Afterstep Applets: Andrea Müller, “Docked”, Linux Magazine Issue 41, April 2004, p78
- [8] Andrea Müller, “Aquatic Utilities”, Linux Magazine Issue 31, http://www.linux-magazine.com/issue/31/BubbleMon_WMFishTime.pdf
- [9] Akpi: <http://akpi.cmd.at/>