



Efficient destops with FVWM

The Chameleon

Computers should adapt to human requirements, and not vice-versa. This is a golden rule of ergonomics. Interestingly, one of the oldest window managers, FVWM, is a shining example of flexibility and configurability. That alone makes FVWM one of the most modern window managers

around. **BY RENE VAN BEVERN**

Virtual Window Manager. The former meaning of the “F” was soon dropped, being inappropriate. There are still rumors that the meaning of the “F” is unknown, although it appeared a lot later as a comment in the source code. This has not stopped users from spreading the legend of the “mysterious F”.

Despite its long history, the star of days gone by has not completely disappeared from the limelight, but gradually made the transition to a modern window manager under the leadership of Dominik Vogt. The fan community continues to grow.

The thing that puts FVWM apart from countless other window managers is its support for a variety of communication channels, such as mouse gestures, which perform actions initiated by specific movements of the mouse. However, to leverage the full power of FVWM, you first have to take the hurdle of editing configuration files. The reward is a unique desktop environment that is unparalleled in its flexibility.

Installation

The best way of getting to know FVWM is to install the 2.5 series developer version. This version proved quite stable and reliable, when tested in our labs. The good news is that you do not need to build the window manager your-

self. There are RPM and Debian packages up for grabs at [2] and [3].

More cautious users, who do not trust developer versions, might prefer to install the stable version of the window manager from the installation media supplied by their distributor. However, the stable version does not have all the features we will be discussing in this article.

If you prefer to build FVWM yourself, check out the `INSTALL.fvwm` file in the source code directory for important details. The install follows the usual three steps, `./configure ; make ; su -c make install`, where the `./configure` script has a lot of optional parameters. You can type:

```
./configure --help
```

to list the options. For example, users who do not intend to run FVWM on multiple screen displays can specify:

```
./configure --disable-xinerama
```

to disable xinerama support.

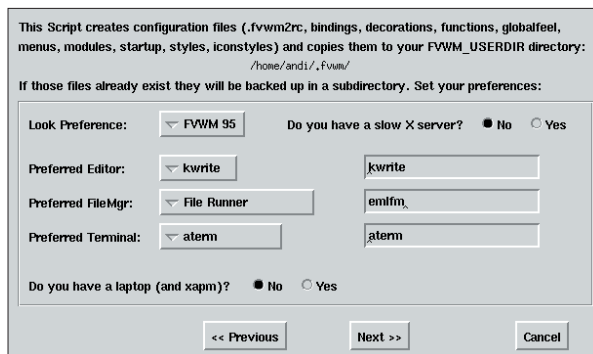


Figure 1: The configuration wizard prompts you for your favorite apps, such as a file manager and an editor.

Every one of us has their own individual idea of a perfect working environment. If you are a programmer, you can simply develop your own, but the rest of us are at the mercy of the configuration options provided by the developers. FVWM [1], the classic window manager created by Rob Nation, allows users a lot of freedom of choice.

History Lesson

FVWM was originally distributed with a version of `rxvt`; the idea was to propagate the software as quickly as possible. This worked, and led to FVWM becoming the standard desktop environment before KDE and Gnome hit the Linux scene. As FVWM was short on features originally, but still capable of managing multiple desktops, the letters were typically interpreted as meaning Feeble

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.

Getting Started

After completing the install from the packages, you should have a FVWM entry in the drop down menu of your **login manager**. Launching FVWM can be somewhat of a shock for users more accustomed to KDE. The empty desktop is a bit frightening. For want of a configuration file, your only option is to press the left mouse button to access the internal menu, and move or rescale windows. So the next step, obviously, is to create a configuration file, before you can get down to any serious work with FVWM.

You can use the FVWM menu to generate a new file. To do so, left click on the desktop and select *Configure with Script95*. A wizard will first prompt you to enter your favorite editor and file manager (see Figure 1), and then generate a basic configuration. You should then restart FVWM to view the new look (see Figure 2).

You can now hold down the left mouse button to move windows around. This action automatically shifts the focus to the foreground, whereas a right click will toggle between the foreground and the background. Left clicking the desktop pops up the main menu, right clicking displays a window list where you can navigate to individual windows.

The buttons in the title bars work just like under Windows – the only difference being that the left button opens a menu with additional window operations, for maximizing or iconifying the window, for example. There is also a keyboard shortcut for the window menu: [Shift-Alt-F2]. [Shift-Alt-F1] for the main menu, and [Shift-Alt-F3] for the window list are also shortcuts to remember.

The next thing to do is to shed some light on the mysterious big gray box on the right hand border of the screen. This is where you can check the system status, launch programs that you need regularly, and toggle between multiple desktops. In the default configuration, FVWM has four desktops which are all

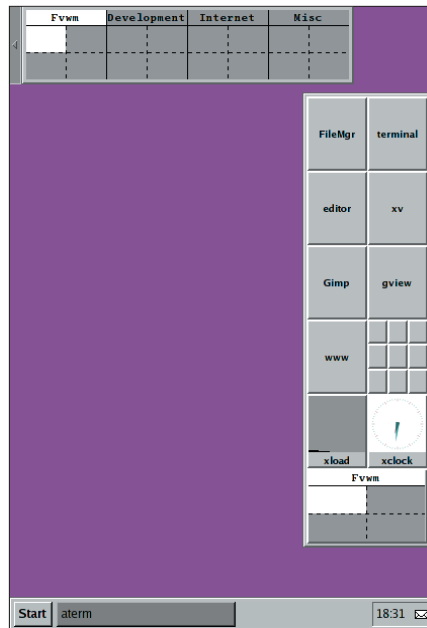


Figure 2: FVWM with the sample configuration has something similar to the Windows taskbar with quickstart buttons for important programs, and a pager for switching between desktops.

subdivided into four panels. The current desktop with its four panels is represented by the bottom pager. You can click on the gray arrow in the top left-hand corner for an overview of the other desktops. To switch panels, press [Shift-Arrow key], to toggle between desktops press [Alt-PgUp] and [Alt-PgDn].

In the default configuration, the accelerator keys mean contortions for the user, but fortunately FVWM does not expect you to be happy with the defaults.

Keyboard Shortcuts

To prevent keyboard shortcuts from colliding with the accelerator keys in other applications, fans of the mouseless desktop are advised to define another **modifier key** besides [Alt] and [Ctrl]. The caps lock key is a useful candidate as almost nobody uses it, although it is quite easily accessible. To use caps lock as a modifier under FVWM, you need to specify an *xmodmap* command when launching the window manager. Enter

the command into the FVWM configuration file, *fwm2rc*. The file will either be located in your home directory, or in a *.fvwm* folder below that level.

The required entry is as follows:

```
AddToFunc InitFunction
+ "I" Exec xmodmap -e "remove
Lock = Caps_Lock"
+ "I" Exec xmodmap -e "add mod4
= Caps_Lock"
```

This removes the caps lock key's normal function ("*remove Lock = Caps_Lock*"), allowing you to use caps lock as an additional modifier in future ("*add mod4 = Caps_Lock*").

To add a keyboard shortcut, you need an entry with the following syntax:

```
Key keyname context
modifier command
```

in *.fvwm2rc*.

"Key" is the keyword that introduces a combination of keys; *keyname* is the description of the required key, such as *A* or *F4*. The context specifies where the keyboard shortcut will apply. For example, should the action only apply if you are working inside a window, or at any time? The latter option is useful if you want to define shortcuts to launch the programs you use most. Table 1 provides an overview of the contexts for FVWM.

Replace *modifier* with the desired modifier key. FVWM recognizes *C* for [Ctrl], *M* for [Alt] and *S* for [Shift]. If you want a key to do something without additionally pressing a modifier key, type an *N* into this field. In contrast, *A* specifies that it does not matter which modifier key the user presses. To use a modifier you defined yourself, simply enter its number here.

For example, we would need to enter a *4* for caps lock. The *command* has to be a command that FVWM recognizes, e.g. *Exec*, which you should be familiar with from the *xmodmap* example. This is fol-

GLOSSARY

rxvt: A terminal emulator similar to "xterm", the KDE Konsole program, or "gnome-terminal"

Login Manager: A GUI-based login window that often has more features than a simple login, for example buttons for shutting down

or rebooting the system. The best known login managers are KDM, GDM, and the classic XDM.

Modifier key: A key that works in combination with other keys. Most Linux systems use [Ctrl] and [Alt] as modifiers.

xmodmap: A keyboard configuration program that allows the user to define what should happen when a specific key is pressed. The application allows users to switch individual keys.

lowed by the program to be launched, ensuring that you use quotes for entries with spaces. For example, the following `.fvwm2rc` entry would launch KMail wherever you are on the desktop by simply pressing [Shift-m]:

```
Key m A 4 Exec kmail
```

FVWM recognizes more commands:

```
Key 1 W C MoveToPage 0 0
```

moves the active window (*W*) to the top left-hand desktop (*MoveToPage 0 0*) when you press [Ctrl-1] (*1, C*). The two zeroes are the values for the virtual X and Y axes with the origin in the top left-hand corner.

Mouse Control

Mouse actions are defined in a similar way to keyboard shortcuts:

```
Mouse key context ⌘
modifier command
```

Mouse specifies that a mouse action follows. The key is an integer between 1 and 5. 1 through 3 are the three normal buttons, 4 and 5 are for mouse wheel movements. The context specification, modifier, and commands are the same as for keyboard entries.

For example, to toggle between desktops:

```
Mouse 4 R A GotoDesk -1 0 0 3
Mouse 5 R A GotoDesk +1 0 0 3
```

Table 1: FVWM Contexts

Shortcut	Area of application
R	Desktop background
T	Title bar of window
S	Side bar of window
F	Window corners
I	Iconified window
W	Whole window
A	Everywhere
Integer 0-9	Buttons in the title bar of the window, which are numbered as follows: 13579 window title 08642

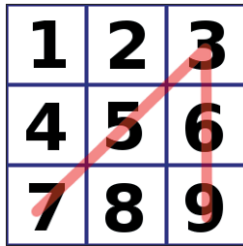


Figure 3: A mouse gesture shown on the phone keypad.

Now, moving the mouse wheel takes you to the previous or next desktop.

In addition to normal mouse actions, FVWM support so-called mouse gestures, just like the Opera browser. The idea is that you can move the

mouse in a specific way to perform an action you defined yourself. For example, dragging the mouse in a straight line from the top to the bottom of the screen could be used to reorganize all the windows. Just like with other keyboard and mouse events, you need to define the mouse button, the modifier and the context for the action.

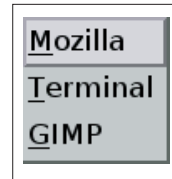


Figure 4: Sample menu with highlighted hotkeys.

The definition of a mouse gesture has the following format:

```
Stroke sequence ⌘
mousebutton ⌘
context modifier command
```

where *Stroke* is the keyword. The *sequence* is described by a series of numbers like on a

phone keypad. A mouse gesture “up diagonally right, straight down” can be described as *75369* (see Figure 3).

Menus

As it is more-or-less impossible to memorize all the key combinations for all the programs you use, FVWM also has menus that are accessible using the

Listing 1: Sample Configuration

```
01 #Use the arrow keys and 23
02 #modifier key # 4 to move to 24 Key Menu R A WindowList
the next window 25
03 #in this direction. The 26 #Left click (1) on the
(Direction) 27 #left title bar button (1) to
04 #is defined by the four points 28 #the window
of the compass. 29
05 #This keyboard combination is 30 Mouse 1 1 A Close
valid everywhere 31
06 #(A). 32 #A gesture with the right
07 mouse button held down (3)
08 Key Left A 4 Direction 33 #from bottom left to top right
West Focus 34 #and then straight back down
09 Key Right A 4 Direction 35 #((75369) reorganizes all
East Focus 36 windows.
10 Key Up A 4 Direction 37
North Focus 38 Stroke 75369 3 A A All
11 Key Down A 4 Direction 39 PlaceAgain
South Focus 40
12 41
13 #Use the Windows key with an 42 #Add entries for Move, Resize,
arbitrary 43 #Iconify, and Maximize to the
14 #modifier to open the window 44 #window menu (AddToMenu
menu WindowOps). The keyboard
15 #(PopUp WindowOps). This 45 #shortcuts for accessing
16 #combination is only valid 46 individual entries are as
inside a window 47 follows
17 #(W). 48 M,
18 #R, y und s
19 Key Super_L W A PopUp 49 AddToMenu WindowOps
WindowOps 50
20 + "&Move" Move
21 # [Menu] used in combination 51 + "&Resize" Resize
with an arbitrary 52 + "Iconif&y"
22 #modifier this opens the 53 Iconify
window list (WindowList) 54 + "Maximize to &screen"
55 Maximize 100 100
```

mouse or the keyboard. For example, left clicking the desktop opens the root menu by default. To help keyboard navigation, you can assign a key to each entry. You could add options for maximizing, iconifying, and closing windows to the Windows key, where the maximize shortcut, [Windows-M], would expand the active window to full screen size. Menu definitions are simple:

```
AddToMenu menuname
+ "first Menu entry" command
+ "second menu entry" command
```

The letter followed by the ampersand (&) is the hotkey and allows access to the menu entry via the keyboard. This letter is underlined in the menu entry, a technique which many applications use. The *menuname* can be an arbitrary name that identifies the menu. This name will be used later to call the menu with the *PopUp menuname* command. You can decide whether a mouse click or a keyboard combination should display your menu. A menu definition for Mozilla, Gimp, and *rxvt* would be as follows (see Figure 4):

```
AddToMenu Applications
+ "&Mozilla" Exec mozilla
+ "&Terminal" Exec rxvt
+ "&GIMP" Exec gimp
```

Besides your own menu definitions, there is also the window list, which contains all the active windows with a serial number. To select a window, you can either click on the window, or press the appropriate number key. FVWM uses the *Window List* command to open the window list, and this makes it easy to assign the event to a keyboard shortcut or mouse action.

Listing 1 shows a sample configuration file with comments. The file defines a number of shortcuts and mouse gestures, and also

has a sample menu.

Room Enough to Share

FVWM will also position windows to your liking. You can even specify the criteria the window manager should apply when positioning each individual window. To define a global method for all windows, you need to add a line with the following syntax:

```
Style "*" positioningmethod
```

to *.fvwmrc*. The asterisk (*) is a wildcard that represents all windows, although you can optionally specify a window name. The name is displayed in the title bar of the window, and you can call the *Identify* command in the window menu just to make sure. The *Class* and *Resource* fields have valid values for window names. The commonest placement methods are as follows:

- *CascadePlacement*: Just like on Windows 3.1, each window will be displayed at a slight offset to all the others, so that the title bars and buttons of each window remain visible.
- *TileCascadePlacement*: Attempts to tile new windows, that is to place them adjacent to other windows. If this does

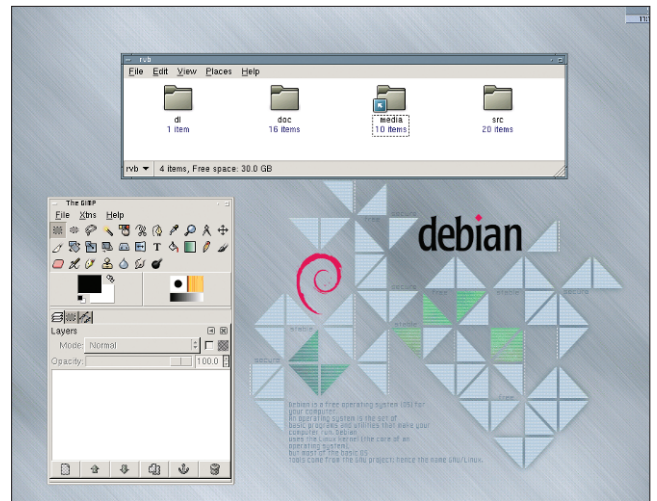


Figure 6: The authors, somewhat plainer configuration.

not work, for lack of room, FVWM will fall back to the *CascadePlacement* method.

- *TileManualPlacement*: Looks for a position for tiling first, and then allows the user to position the window manually, if there isn't enough room.
- *MinOverlapPlacement*: Overlaps windows by as little as possible when opening new windows.
- *ManualPlacement*: Allows the user to place the windows manually. Before a window opens, a yellow frame is displayed. The frame follows the mouse cursor. Users can then click to drop the window on the desktop.

The *MinOverlapPlacement* method is the easiest of all, as it automatically looks for the best position.

Unlimited Options

Both configurations (Figures 5 and 6) are a useful basis for your own file. If you are looking for more inspiration, you can try the sample configuration supplied with FVWM, which is typically located in the */usr/share/fvwm* directory. For more information check out sources such as the exhaustive FVWM manpage, with its command reference, and the FVWM wiki at [4].

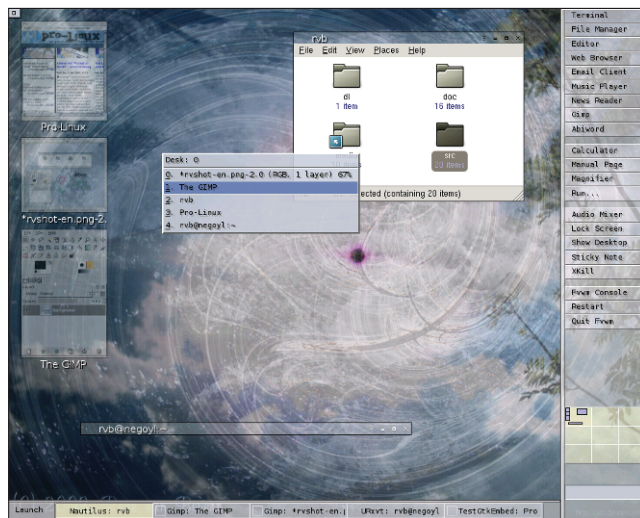


Figure 5: FVWM with transparent menus and miniaturized windows for iconified windows.

INFO

[1] FVWM: <http://www.fvwm.org/>

[2] RPM packages: <http://fvwm-themes.sourceforge.net/rpm/>

[3] Debian packages: <http://fvwm-themes.sourceforge.net/deb/>

[4] FVWM wiki: <http://fvwm.rvb-web.de/>