

On GameFAQs: [Halo 3](#)

- CNET Networks Business:
- [BNET](#) |
- [TechRepublic](#) |
- [ZDNet](#)

SolutionBase: Get familiar with alternative Linux desktops

by [Jack Wallen](#) | Nov 15, 2007

Takeaway: Desktop customization in Linux is very flexible; from the ultra-modern KDE and GNOME window managers to with the likes of Fluxbox and AfterStep, there's a Linux desktop to suit everyone. Jack Wallen covers some of your Linux desktop options.

Related Content

- [right click not responding in xp](#)
- [getDirectories in smart device app](#)
- [taskbar](#)
- [Annoying pop ups](#)
- [help files open when xp pro boots](#)

Linux has proven amazingly flexible: after nearly 10 years of use, I'm still impressed by how the Linux operating system does exactly what I want on any type of hardware. Desktop customization is no exception; from the ultra-modern KDE and GNOME window managers to with the likes of Fluxbox and AfterStep, there's a Linux desktop to suit everyone.

Author's note

Like the majority of window managers (WMs), Fluxbox and AfterStep work on most Linux distributions. Your particular distribution may not have additional WMs in their base packages, so you may need to download additional files to get

them to work. The environment I'll be using is a fresh install of Ubuntu 7.0.4 (Feisty Fawn) with the following (significant) specs:

- AMD 2800+ processor
- 528 MB of RAM
- Via Technology integrated VT8378 video

Obviously this machine is no powerhouse, but these alternative desktops do not require much iron to run them faster than a KDE, GNOME, OS X, or Windows desktop on a machine twice as powerful.

A bit about window managers

A fundamental difference between KDE and GNOME and alternative WMs is that desktop environments such as KDE and GNOME are fully-integrated environments, which means that they include many applications that also depend upon a WM to draw windows for them. A WM, on the other hand, only draws windows and menus, and so on. A WM by itself is smaller, faster, and lighter, but less powerful in most cases.

Essentially, a WM is perfectly suited for less powerful machines or users that do not want the bloat of a full desktop environment (DE).

None of this is to say that a WM is less usable; in many instances, a WM is the simplest of all GUIs to use. Their simplicity makes them absolutely perfect for people who have trouble getting around the sometimes complex world of a full DE.

With that in mind, let's take a look at two very nice WMs.

Fluxbox

Fluxbox is based on Blackbox. The Fluxbox developers saw that Blackbox had seriously stagnated and wanted to see it come back to life, so they helped it out by improving the code, adding features (some of them *amazing*), and planning ahead. Here's the list of current features:

- Configurable window tabs
- Simple Icon Bar (for minimized and iconified windows)
- Multiple workspaces

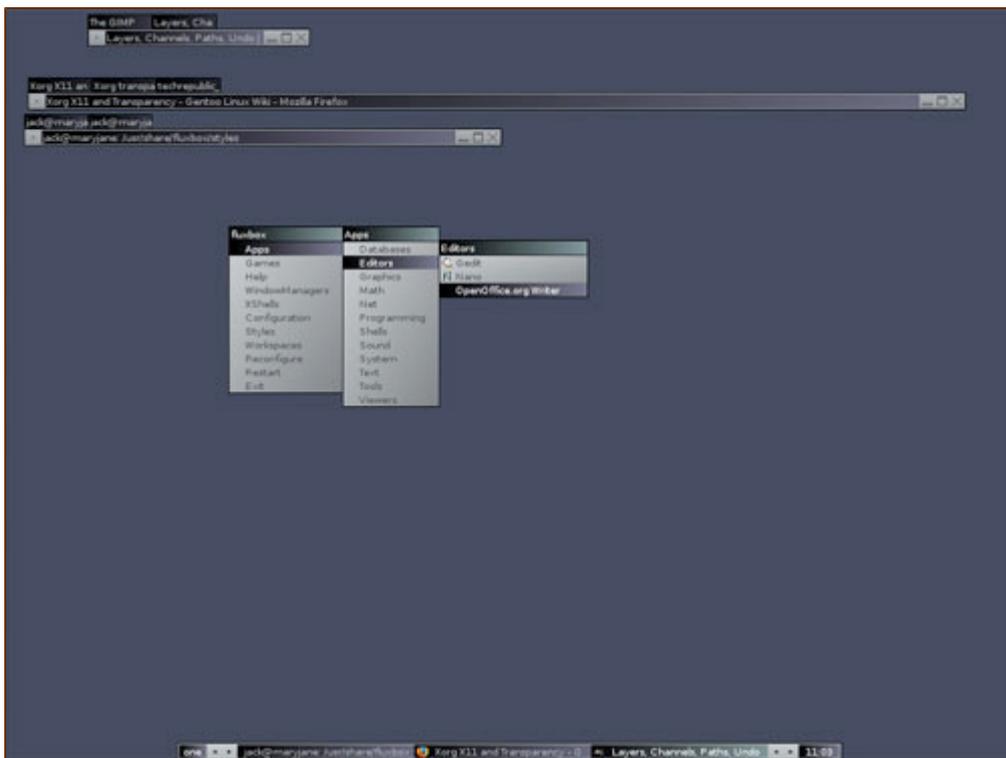
- Configurable titlebar
- KDE support
- New native integrated KeyGrabber (with support for emacs like keychains)
- Partial GNOME support
- Extended Window Manager Hints support

Ahead for Fluxbox:

- Session Management
- Windows Snapping
- Configurable toolbar

Let's view what Fluxbox really looks like. **Figure A** shows the basic Fluxbox desktop using the Spiff style. A total of 22 styles come with Fluxbox, so you have many other looks to choose from.

Figure A



As you can see, there are shaded windows open, a mouse menu, and the toolbar.

Let's take a look at the individual components. The first is the toolbar, a fairly minimalistic tool; basically, this is a bar where applications appear, whether they are minimized or not. Other things on the bar include the workspace

switcher and a clock; each item can be configured via a right-click on the toolbar. When you right-click the toolbar, you'll see the menu shown in **Figure B**.

Figure B



You can hide the toolbar from this menu by selecting Hide.

Here are the toolbar configuration descriptions:

- **Auto Hide:** Auto-hides the toolbar when it is not being used.
- **Toolbar width percent:** The percentage the toolbar takes up.
- **Maximize Over:** When enabled, maximized windows will cover the toolbar.
- **Layer:** The layer of the desktop the toolbar will be on.
- **Placement:** Location of toolbar.
- **Alpha:** Transparency of the toolbar.
- **Iconbar Mode:** The iconbar (within the toolbar) settings.
- **Clock:** Format of the clock.
- **Edit Clock Format:** Customize the clock format.

Aspects of the toolbar that can not be configured via the toolbar are the workspace names. A workspace is basically a desktop, and the Linux desktop is capable of having multiple desktops. By default, Fluxbox has four workspaces. To configure the workspaces, middle-click the desktop; this brings up the Workspace menu.

When you bring up the Workspace edit menu and select Edit Workspace Name, a small text area will appear where you enter the new name of the workspace. Now that you've renamed the workspace, it shows up on the toolbar, as shown in **Figure C**.

Figure C



Keep the names short if possible.

Fluxbox Menu

One of the drawbacks (for users at least) is that the Fluxbox Main Menu must be hand-edited in a text editor. By default, the menu to edit is `/etc/X11/fluxbox/fluxbox-menu`. You need root access to do this. To make this easier, copy this menu to the user's `~/fluxbox/` and rename it `menu`.

A typical menu entry looks like this:

```
[exec] (OpenOffice.org Writer) {/usr/bin/oowriter}
</usr/share/icons/gnome/32x32/apps/openofficeorg-21-writer.xpm>
```

Here's the breakdown of the entry.

- **[exec]:** Tells Fluxbox this is a command.
- **(OpenOffice.org Writer):** What the entry will be named in the menu.
- **{/usr/bin/oowriter}:** The actual command (must be inside of {}).
- **</usr/share/icons/gnome/32x32/apps/openofficeorg-21-writer.xpm>:** The location of the icon image.

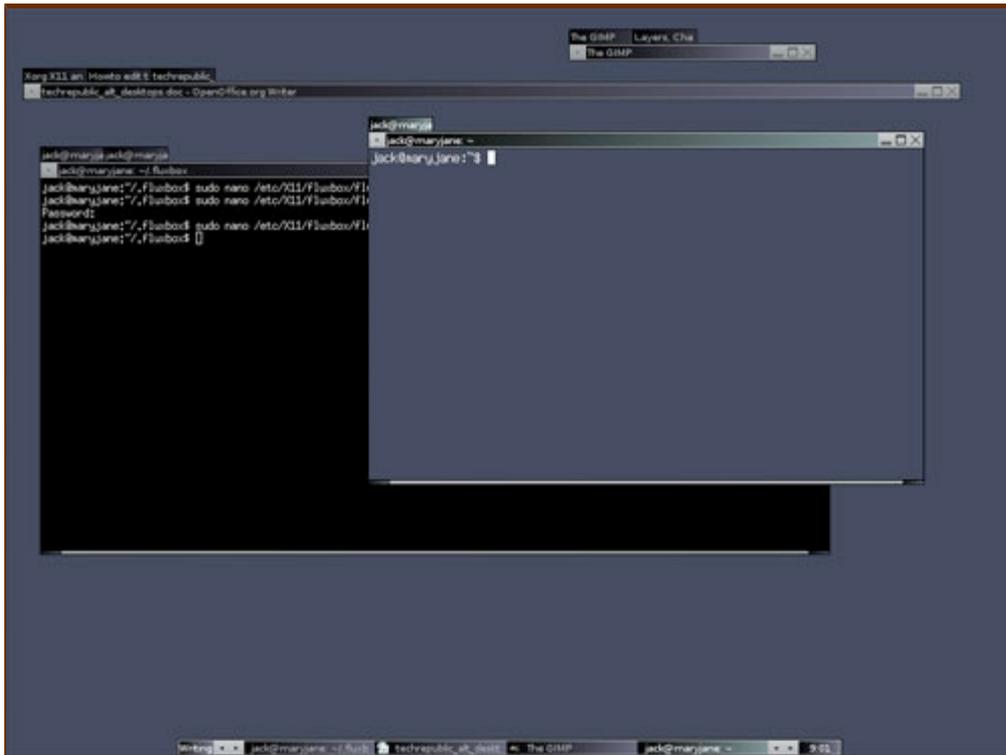
What good is just adding a bunch of entries into the menu without categorization? Let's break it down into submenus. Suppose you want to have a group of tools for the Internet. Here's what that would look like:

```
    [submenu] (Internet) {}                [exec] (Evolution)
{/usr/bin/evolution} </usr/share/pixmaps/evolution.xpm>
    [exec] (Firefox Web Browser) {firefox}
</usr/share/pixmaps/firefox.xpm>        [exec] (Gaim)
{/usr/bin/gaim} </usr/share/pixmaps/gaim-menu.xpm>
[end]
```

As you can see, individual entries are bookended with the `[submenu] (NAME OF SUBMENU) {} [end]`. The brackets are empty because there's no command to execute.

Tabbing together Windows

Even though Fluxbox isn't as robust as other environments, it does have some cool features. Tabbed Windows is one of those little features you wish every desktop could have. Check out **Figure D**:

Figure D

You can't see the X-term because of the A-term.

If you kept opening shells, you could wind up with a mess on your hands. Fluxbox, however, fixes that. If you right-click the little tab at the top left of the titlebar and drag it so that it is on top of the tab of a window you'd like it to join, those two windows will be linked together and function similarly to tabbed browsing in Firefox.

It doesn't matter which applications you join together. If you look carefully back at Figure D, you will see a titlebar with three tabs, one of which says "techrepublic_". Two of those are browser windows, and the final tab is this article, which I am writing in OpenOffice.org Writer.

AfterStep: A step back in time

When I began using Linux, my first experience was with Fvwm95. Although this was somewhat familiar, it was a bit on the ugly side'; since I've always been fond of aesthetics, that particular window manager didn't suit me. My search for an eye-pleasing GUI brought me to AfterStep -- based on the look and feel of the NeXTSTEP interface -- which was growing popular with the Linux crowd. It was special because it was doing things most other desktops weren't doing. AfterStep was a far cry from anything the general public had seen.

Unfortunately, AfterStep became a victim of its own uniqueness. Take a look at **Figure E**: As you can see, there are a lot of elements on the desktop.

Figure E



It's pretty, but there are a lot of elements, which could confuse a new user.

Included in the default desktop:

- **The Wharf (bottom-right vertical and top-left horizontal, under the winlist of Figure E):** A free-floating application launcher that can contain menus and applets, as well as applications.
- **Winlist (top):** A small toolbar that holds icons for minimized and maximized applications. This is the fastest way to switch between applications.
- **Icons (bottom-left):** These are not standard icons. When an application is minimized without having been switched via the winlist, the application will minimize to a small icon. To open the application, click the icon.
- **Pager (top-right vertical):** Where you switch between desktops.
- **Menu (center):** To get the application menu, click the left mouse button.

I've always tended to use AfterStep with a minimal amount of components. To get rid of the pieces you don't want, you have to hand-edit the

/usr/share/afterstep/autoexec text file. In that file, you'll see entries that look like:

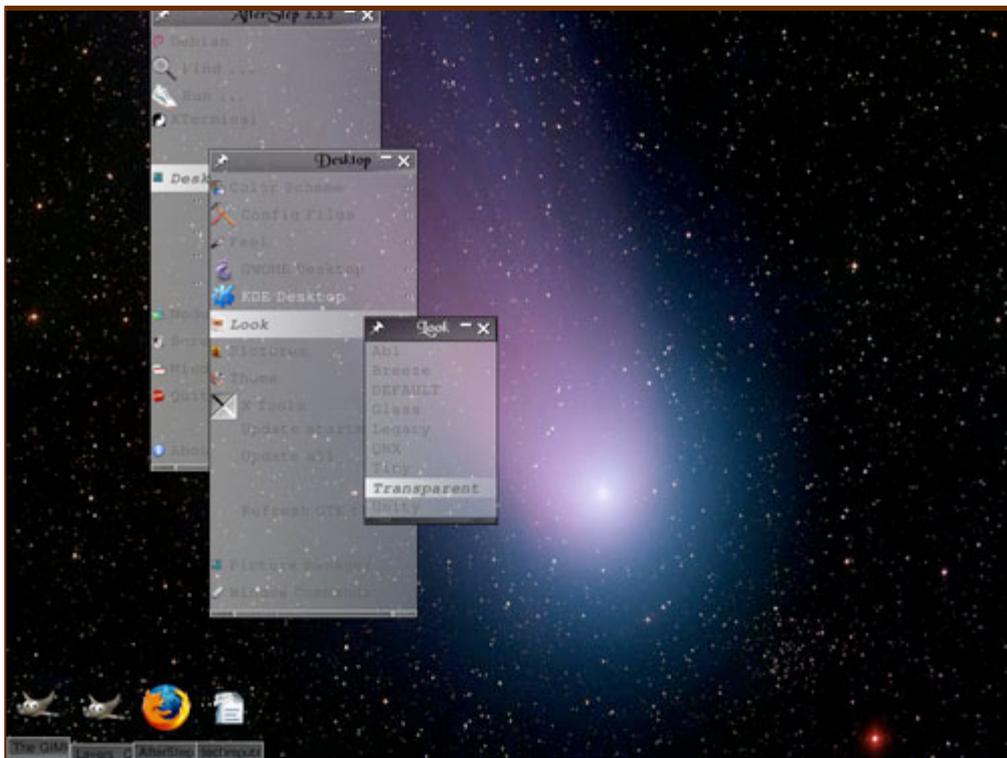
```
Module "I"      WinList
Wait    "I"      WinList
Module  "I"      MonitorWharf
Wait    "I"      MonitorWharf
Module  "I"      Wharf
Wait    "I"      Wharf
Module  "I"      Pager 0 3
```

In order to disable a feature you don't want, comment the feature out, like so:

```
#      Module  "I"      WinList
#      Wait    "I"      WinList
#      Module  "I"      MonitorWharf
#      Wait    "I"      MonitorWharf
#      Module  "I"      Wharf
#      Wait    "I"      Wharf
#      Module  "I"      Pager 0 3
```

After disabling all the desktop elements, changing the theme, and adjusting the color scheme, I have a throwback AfterStep desktop, as shown in **Figure F**.

Figure F

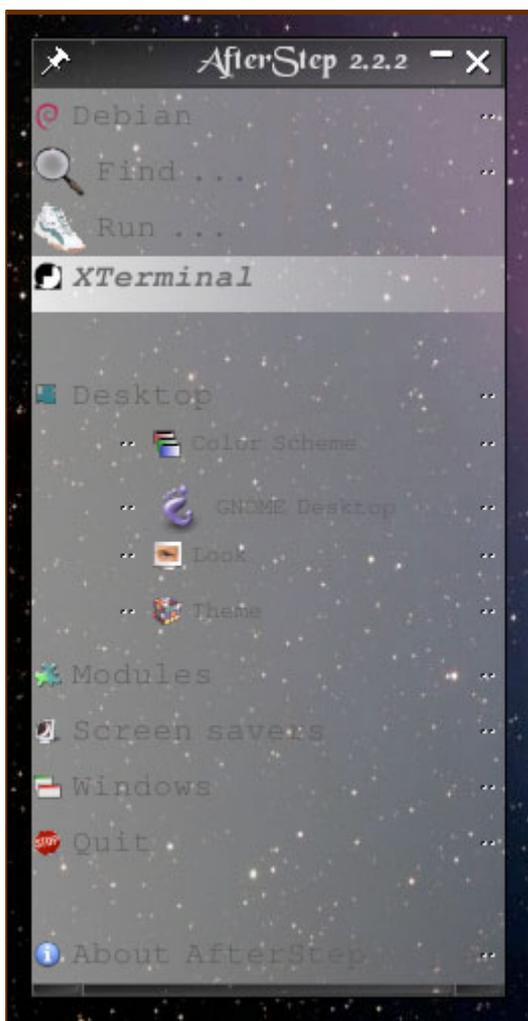


Nothing but a mouse menu and minimized icons: A minimalist's dream.

Mouse menus

Let's take a look at the three mouse menus available. The left-click menu, shown in **Figure G**, is the primary menu. This menu opens up applications, configures the desktop, exits from the session, and gets you help with AfterStep.

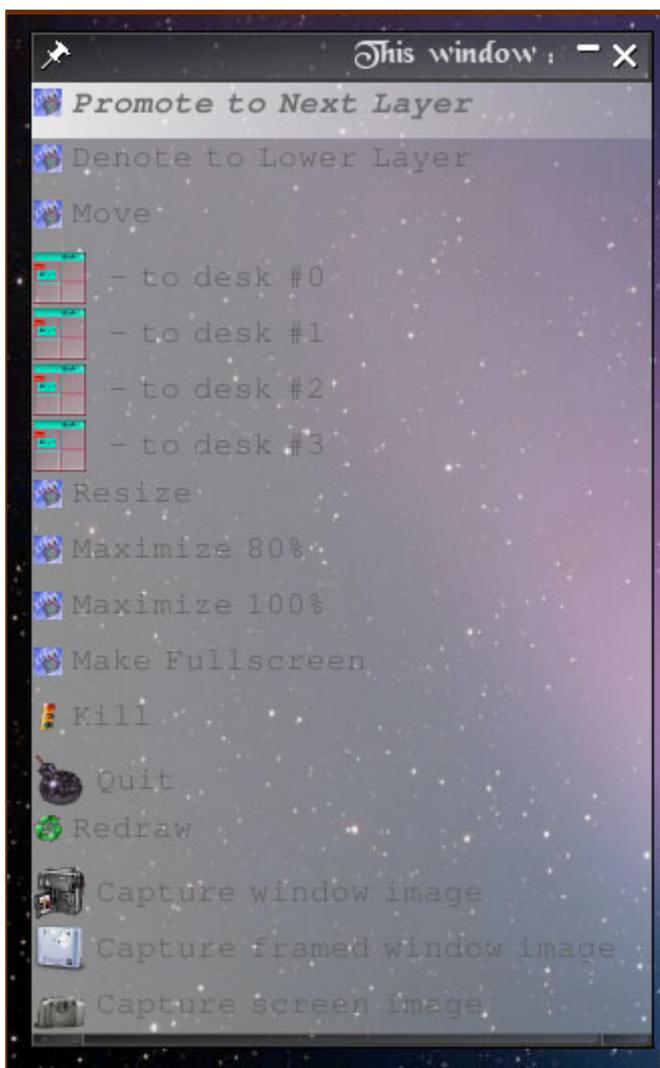
Figure G



A neat feature: press on the push-pin icon to keep the menu open while working.

The middle-click menu gives you a number of options to enact upon windows. This menu allows you to select an action (Kill, for example) and then select the window you want to act upon to run the command. So if you're having problems with Firefox, middle-click the desktop. You'll then see a menu similar to the one in **Figure H**. Select Kill and then click on the Firefox titlebar to kill the application.

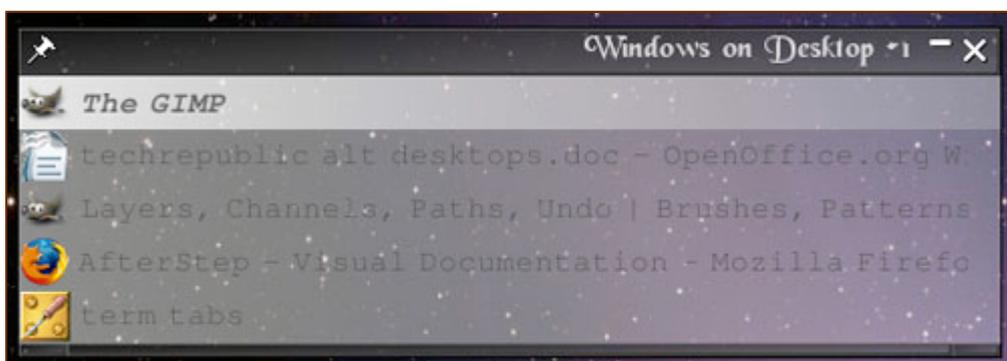
Figure H



Need to kill a window? This figure shows you how.

The right mouse menu acts as the winlist -- a list of running applications -- as shown in **Figure I**.

Figure I

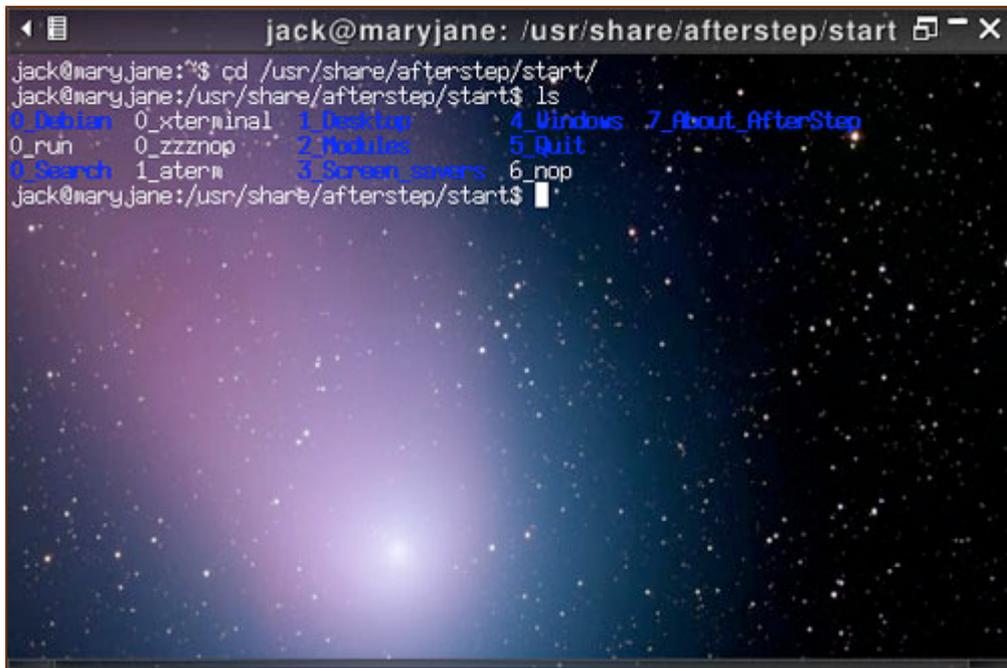


Even if an application is minimized, it will leap up to the front if selected from the winlist menu.

In order to alter the AfterStep application menu, you're going to have to

hand-edit text files. Unlike Fluxbox, however, AfterStep requires the manipulation of multiple files. Take a look at the `/usr/share/afterstep/start` directory; you should see something similar to what's shown in **Figure J**.

Figure J



The only thing you will not see in your default `start` directory is the `1_aterm` entry (that was my doing).

As you can see, there are both files and directories. The files act as single menu entries consist of one or two-line menu entries that look like:

```
Exec "aterm" exec aterm -tr -fg white -bg blue +sb &
MiniPixmap "mini-app.xpm"
```

The entry is broken down in nearly the same way as the Fluxbox menu entry:

- **Exec:** The name of the entry follows this.
- **"aterm":** The name of the entry.
- **exec:** The actual command follows this.
- **aterm -tr -fg white -bg blue +sb:** The actual command.
- **&:** Tells the system to run the command, but give the prompt back.
- **MiniPixmap:** The mini icon used for the menu entry follows this.
- **"mini-app.xpm":** The actual icon for the entry.

In the same *start* directory, you'll find subdirectories. Each subdirectory serves as a submenu within the main menu. Inside of each subdirectory are files for each application needed for the subdirectory. If you look into the *o_Debian* menu (the main application menu) you will see nothing, until you look for the hidden file called *.include*. In this file is a pointer to the */var/lib/afterstep/menu/Debian* directory. Within that directory, you'll find all of the application's menus and submenus.

Final thoughts

I've always been a big fan of alternative desktops. It's been a long time since I've used anything but Enlightenment, Fluxbox, or AfterStep. There are many reasons for my choice: aesthetics, speed, reliability, and because I am an old-school Linux rebel and it just feels good to use something *not* Windows. Everyone else will likely find appeal in the former categories, and in the ability to bring older machines to life.

Give one of the environments a try; you'll find yourself going back to them more often than not.

Copyright © 2007 CNET Networks, Inc. All Rights Reserved. [Privacy Policy](#) | [Terms of Use](#)