

Openbox guide

Contents

- [1. Introduction](#)
- [2. Sources](#)
- [3. Installing Openbox](#)
- [4. Configuring Openbox](#)
 - [4.1 Menus](#)
 - [4.2 Obconf](#)
 - [4.3 Keyboard bindings, mouse bindings, and application settings](#)
- [5. Themes](#)
 - [5.1 Openbox themes and fonts](#)
 - [5.2 Gtk themes, fonts and icons in Openbox](#)
 - [5.3 Qt themes, fonts and icons in Openbox](#)
 - [5.4 More on fonts](#)
 - [5.5 Mouse cursor themes](#)
- [6. Autostarting applications](#)
- [7. Managing the Desktop — Wallpapers and Icons](#)
 - [7.1 Wallpapers](#)
 - [7.1.1 Feh](#)
 - [7.1.2 Nitrogen](#)
 - [7.1.3 Other applications](#)
 - [7.2 Icons on the desktop](#)
- [8. Panels and the like](#)
 - [8.1 Panels](#)
 - [8.2 Task lists](#)
 - [8.3 System trays](#)
 - [8.4 Launchers and docks](#)
 - [8.5 Pagers](#)
 - [8.6 Clocks](#)
- [9. Shadows, Fade-in/out menus, and more fancy stuff](#)
 - [9.1 Shadows and Fading menus](#)
 - [9.2 Transparency](#)
 - [9.3 Desktop cubes](#)
- [10. File Managers](#)
- [11. OpenOffice in Openbox](#)
- [12. Tile windows in Openbox](#)
- [13. Shut down and reboot, hibernate, suspend and locking the screen](#)
 - [13.1 Shutdown, reboot and suspend](#)
 - [13.2 Lock the screen](#)
- [14. More useful applications](#)
 - [14.1 System monitors](#)
 - [14.2 Other applications](#)
- [15. Conclusion](#)

1. Introduction

Welcome. This guide is meant to help you set up and configure the window manager [Openbox](#) on [Ubuntu](#) (though much of the information below will also be useful for users of other Linux distributions).

This guide is *not* meant to help you set up a minimal install of Ubuntu with Openbox, though it might be useful for anyone attempting to do that. I only cover how to set up, configure and use Openbox, and assume you already have Ubuntu up and running, either in a command line install, a normal Ubuntu install, or something in between. (If you need help with a minimal install of Ubuntu, have a look [here](#) and [here](#).)

You can use Openbox on its own, or as a window manager in Gnome, KDE or Xfce. This guide is meant for a stand-alone Openbox set-up. For help how to use Openbox in one of these desktop environments, have a look at the [Openbox website](#). You can use both Gtk (Gnome) and Qt (KDE) applications in Openbox and I have tried to give both equal attention, though there are generally more solutions using Gtk apps than Qt apps.

Openbox is very minimalist and does little more than manage the windows (but does that extremely well!). This often intimidates new users, who may find Openbox too spartan or simplistic. Rather than being a flaw, Openbox' minimalism is actually its strength. It doesn't come with panels or desktop configuration tools, but, as you will see, there are plenty of applications available that accomplish such tasks very well. You can pick and choose whatever you like best. By doing less, Openbox thus gives you more control, and the option to turn your desktop environment into something that is exactly what you like it to be (or at least close).

I mention a lot of applications in the following pages, and often mention some configuration options. I recommend reading the manual pages of whatever application you are interested in (open a terminal and type *'man nameoftheapplication'*; press 'q' to exit) as you'll find plenty more options to try out. The websites of the applications often give helpful guides as well. If an application has a website, I have linked to it. Many applications are much more configurable than they initially appear to be! Explore and settle on the configuration that suits you best.

I have tried to give guidelines for doing things both from the command line and a graphical interface (using file managers, text editors, etc.). Pick whatever method you prefer. In the end, they both accomplish the same thing.

For those who may find it useful, I have uploaded my own Openbox configuration files. You can access the rc.xml file [here](#), and the menu.xml file [here](#). There is nothing very special about them (other than that they work for me), but I have learned a lot and gathered many ideas from looking at the configuration files of other users; perhaps you will too.

If you've never used Openbox before and are wondering what it can look like, behold some of my screenshots [here](#).

2. Sources

Most of the information presented here is drawn from other sources. I am particularly indebted to

- [The Openbox documentation](#)
- [K.Mandia's blog](#)
- [The "random openbox chatter" thread on the Ubuntuforums](#)
- [Raublekick's How-to on the Ubuntuforums \(now outdated\)](#)
- [The Ubuntuforums](#)
- [The Gentoo Wiki](#)
- [The Debian forums](#)
- [The Arch Wiki](#)

Special thanks also to those that have helped me discover Openbox and offered assistance when problems arose: fuscia, K.Mandia, moore.bryan, bonzodog, ynnhoj, and raublekick (all from the Ubuntuforums), and the folks on the [IRC #openbox channel](#). A lot of their instructions, guidelines and ideas are incorporated into this guide.

3. Installing Openbox

The latest version of Openbox (3.4.7) is not in the Ubuntu repositories. Though it is much easier to install the Openbox version that is in the repositories (3.4.6 in Hardy), the latest version has plenty of [features](#) that make it worth it to do it the (slightly) harder way.

Compiling from source is not as hard as it looks, though. First, you'll need to make sure you have installed the proper dependencies that are needed to compile and run Openbox. To do so, open a terminal and enter the following command:

```
sudo aptitude install build-essential pkg-config checkinstall
sudo apt-get build-dep openbox
```

The first command will install some basic packages that you need to compile and install applications from source. I've added checkinstall, as it is a useful tool to install packages compiled from source in Ubuntu. Checkinstall creates a .deb package and installs that. This allows you to remove the application very easily using apt or Synaptic. If you prefer not to use it, leave it out and replace *'sudo checkinstall'* with *'sudo make install'* everywhere below.

The second command tells apt, the package manager of Ubuntu, to install all the dependencies that are needed to build Openbox.

Next, download the Openbox source code from the [Openbox website](#), extract the archive and open a terminal in the newly extracted directory. To do all this from the command line, use the following commands:

```
wget http://icculus.org/openbox/releases/openbox-3.4.7.2.tar.gz
tar xzvf openbox-3.4.7.2.tar.gz
cd openbox-3.4.7.2/
```

Now you can build and install the package:

```
./configure --prefix=/usr --sysconfdir=/etc
make
sudo checkinstall
```

All those commands will produce a lot of output in the terminal. You can ignore all of it, unless it reports an error in which case only the first error reported is important. What do these commands actually do? The first command checks whether you have all the required dependencies installed and builds a 'make' file suited to your system. The 'make' command builds the source code with the 'make' file (which may take a while, so be patient), and checkinstall installs that source code using your computer's package management software (apt/dpkg in the case of Ubuntu).

On some computers that I have tried to install Openbox in this way, checkinstall would abort (*'chmod: changing permissions of '/usr/lib/libobparser.a': No such file or directory'* was the reason given). If you run into this, you should be able to install it with *sudo make install* (instead of *sudo checkinstall*). If you'd still like a .deb package (to be able to remove Openbox through apt or Synaptic), you should be able to reinstall it with checkinstall after you've installed it with 'make install' without any difficulties.

If all goes well, Openbox is now installed. If you use GDM or KDM, you should be able to select Openbox in the sessions menu (F10 > Sessions). If you have KDE or Gnome installed, you should also see options to use Openbox as a window manager in either desktop environment ('KDE/Openbox' and 'Gnome/Openbox'). Select Openbox, and login. Now you can start the configuration.

4. Configuring Openbox

4.1 Menus

The first time you log into Openbox, you'll be greeted almost instantly with a blank screen. This is Openbox. Right click on the desktop and you will find your *root-menu*. Middle click on it and you will have the *client-list-combined-menu*, which shows all running applications on all desktops. The 'Exit' entry on the root-menu is to log out of Openbox (be aware that it doesn't ask for a confirmation and will automatically kill all running applications); 'Reconfigure' reconfigures Openbox, which you will need to do to apply changes you made to your configuration files.

The default menu in Openbox is very bare and empty, but editing the menu is relatively easy, so don't despair!

First of all, copy the default menu.xml file, found in */etc/xdg/openbox/*, to the Openbox configuration folder in your home directory, */home/USERNAME/.config/openbox/*. This last directory is where all your configuration files for Openbox will be stored, so remember the location! You can copy the file using the file manager of your choice, or using the terminal by entering the following command:

```
cp /etc/xdg/openbox/menu.xml ~/.config/openbox/menu.xml
```

Now you can open the menu file (*texteditorofyourchoice ~/.config/openbox/menu.xml* in the terminal) and start editing it *à volonté*. Editing the root-menu is very simple and straightforward, and is done by editing this single file.

To make things even easier you can use Obmenu, a graphical application to edit the Openbox menu. Obmenu is in the Hardy and Gutsy-backports repositories. Use the following command to install it:

```
sudo aptitude install obmenu
```

Unfortunately, Obmenu is not any other Ubuntu repository. If you are using a version older than Gutsy, you'll have to build it from the source code. First install its dependencies:

```
sudo aptitude install python2.5 python-glade2 python-gtk2
```

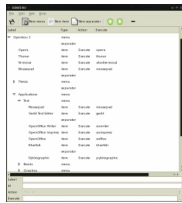
Now download the source code from the [Obmenu website](#). Extract the package and move into newly extracted Obmenu directory with the terminal.

```
tar xzvf ~/obmenu-1.0.tar.gz
cd obmenu-1.0
```

Install the application with the following command:

```
sudo python setup.py install
```

Once it is installed, you can start Obmenu with the command 'obmenu'. Once it is loaded, you can edit your menu by clicking a few buttons and typing the entries of your choice. It couldn't be easier! Here is what Obmenu looks like:



If you want to have all the applications installed on your computer to appear in your menu, you could use [MenuMaker](#). Download the source code, unpack the archive and move into the extracted directory with the terminal

```
tar xzvf menuMaker-0.99.7.tar.gz
cd menuMaker*
```

You can install MenuMaker with the usual commands (`./configure`, `make` and `sudo checkinstall`), but you can also run it without installing with the following command:

```
./mmaker OpenBox3
```

(If you installed menuMaker, run it with the command `mmaker OpenBox3`). You should now have a full menu available. Note that MenuMaker will not overwrite existing menu.xml files. If you have already changed your menu.xml file, you'll need to back it up, remove it, and then add the changes to the new menu.xml file MenuMaker created, or force MenuMaker to overwrite any existing file with the `-f` flag (`mmaker -f OpenBox3`).

If you would like to use the Debian menu in Openbox, follow the instructions found [here](#).

Be creative with the menus, and make your life easier! You can enter basically any command in the command part. If the application needs administrative privileges (such as Synaptic), add 'gksudo' before the application command and you will be asked to give your password before you can run the program. You can create submenus for wallpapers, using Feh (see [below](#)); add entries to change configuration files (`TEXTEDITOR /path/to/text/file`) or to open certain folders (`FILEMANAGER /directory/of/choice`); you can add entries to back up documents (`cp -a /original/directories /destination/directory`); etc. If you like entries to turn off applications, such as panels or conky, you can create a new entry with the following command: `killall nameoftheapplication`.

You can also use 'pipe menus', menus that are automatically generated every time you select them. There are pipe menus available for Gnome's 'Places', Gmail, rss feeds, mpd players, changing of themes, setting wallpapers, and much more. You can find several pipe menus and installation instructions [here](#).

When you edit your menu.xml file manually make sure to 'Reconfigure' Openbox (in the root-menu) so the changes become visible. Obmenu should do so automatically when you 'Save' the menu you are working on. If your menu file contains broken xml code, Obmenu will not start. When you launch it in the terminal, you'll get an error message that ends with a line like this: `xml.parsers.expat.ExpatError: not well-formed (invalid token)` followed by the line of the menu.xml file that contains the error. Luckily Openbox 3.4.7 warns you whenever it starts or reconfigures when your menu and configuration files contain errors.

You can also create custom menus, that can be launched separately from the root menu. For more info on this, see the [Openbox documentation](#).

Note that, unlike Fluxbox or Icewm, Openbox does not support icons in the menus.

4.2 Obconf

Once Openbox is installed and your menus are set up to your liking, you can configure Openbox further. All of the Openbox's configuration is done in the rc.xml file in `/home/USERNAME/.config/openbox`. You can edit this file manually, in your chosen text editor, but you can also use [Obconf](#) to change many of the settings in that file.

The latest version of Obconf is in the Hardy (and Gutsy-backports) repositories, but you won't be able to use it, as it has been compiled for the older versions of Openbox. You will, in other words, have to build it yourself from source. Note that if you already have Obconf installed you will also need to recompile it to be able to use it with the latest version of Openbox.

Download the [source code](#), extract the files and move the terminal into extracted directory. To do so from the command line, type the following:

```
wget http://iccu.us.org/openbox/obconf/obconf-2.0.3.tar.gz
tar xzvf obconf-2.0.3.tar.gz
cd obconf-2.0.3
```

Compile Obconf as follows:

```
PKG_CONFIG_PATH=/usr/local/lib/pkgconfig ./configure --prefix=/usr
make
sudo checkinstall
```

Once Obconf is installed, run the following commands in the terminal to associate the .obt file type (used by Openbox theme files) with Obconf:

```
update-mime-database /usr/share/mime
update-desktop-database /usr/share/applications
```

You can then run Obconf with the command 'obconf'. Here is what it looks like:



The many options in Obconf should be fairly straightforward. The dock of the rightmost tab is the place where you can load 'dock apps', such as system trays, launchers, pagers, clocks, etc. (more info about these [below](#)).

4.3 Keyboard bindings, mouse bindings, and application settings

Not all settings in the rc.xml file can (at present) be changed with Obconf. You'll have to edit the rc.xml file for those.

Keyboard bindings

Openbox makes it very easy to assign particular key combinations (keyboard bindings) to particular commands (such as launching an application, showing a menu, or switching to a particular workspace). These bindings can be added or edited in the "keyboard" section of the rc.xml file. Here are some example settings:

1. For an application or command, using Thunar as an example:

```
<keybind key="A-F4">
<action name="Execute">
<execute>thunar</execute>
</action>
</keybind>
```

2. For a menu, using the root-menu as an example:

```
<keybind key="A-F1">
<action name="ShowMenu">
<menu>root-menu</menu>
</action>
</keybind>
```

3. For switching to a particular workspace:

```
<keybind key="A-1">
<action name="Desktop">
<desktop>1</desktop>
</action>
</keybind>
```

A= Alt, C= Control, W= Windows key. Escape= escape key. Delete= delete key. If you want to use a particular key, but are unsure of its name, you can use [xev](#) to find out what it is called: type xev into a terminal and then press the key you want to use. A lot of information will appear; you need what comes after 'keycode'.

To assign two (or more) commands to a single keybinding, you'll have to use the following syntax:

```
<keybind key="A-F4">
<action name="Execute">
<command>thunar</command>
</action>
<action name="Execute">
<command>osdctl -s 'Thunar'</command>
</action>
</keybind>
```

This will launch the file manager Thunar and prompt osdsh to display the text "Thunar" on my screen.

For more information on keybindings, have a look at the [Openbox documentation](#). If you're interested in my keybindings, have a look at [this post](#).

Mouse bindings

The next section in the rc.xml file concerns the mouse. It governs the things that happen when you click, drag or press a particular mouse button in a particular area. For more information, have a look at the [Openbox documentation](#). For some examples of what you can do with both keybindings and mousbindings, have a look [here](#).

Applications

At the end of the file, you will find settings that affect the way applications are run. For example, if you want your terminal only to appear on workspace 2, or would like xclock to always hide its window decorations, or you want orange to skip the taskbar, or you would like to combine several of these actions for one application, you can set these settings here. Here is an example:

```
<application name="xfce4-terminal">
<desktop>3</desktop>
<layer>below</layer>
<decor>none</decor>
<maximized>yes</maximized>
</application>
```

Everytime you now run xfce4-terminal, it will open on workspace 3, maximized, without window decorations, but below all other applications.

If you want remove all window decorations, you can use the following settings:

```
<application class="*">
<decor>none</decor>
</application>
```

For more info, read [this](#)

When you change your rc.xml file make sure to 'Reconfigure' Openbox (in the root-menu) so the changes take effect. The changes made through Obconf should appear immediately.

5. Themes

5.1 Openbox themes and fonts

Changing the theme for Openbox is very simple. Open Obconf, and select the theme of your choice. To install new themes, you can either use the option in Obconf to install an .obt archived theme, or just extract the theme archive to `/home/USERNAME/themes` and then select the theme in Obconf (you'll have to restart Obconf if it was running before you copied the theme files there). Alternatively, you can edit the rc.xml file manually to change the theme (search for "theme").

Good places to find more Openbox themes are [Box-look.org](#) and [Freshmeat.org](#). There is also a package called [openbox-themes](#) in the Ubuntu repositories that contains a large number of Openbox themes.

You can easily create or modify your own Openbox themes. For more info on this, have a look at the [theme options](#).

To change the font used by Openbox (in the window decorations, menus, and on-screen-display, you can use Obconf (under the second tab 'Appearance')

5.2 Gtk themes, fonts and icons in Openbox

When you start Openbox, you will notice that your Gtk applications use the default ('ugly') grey theme and icons and the default font. There are several ways in which you can change the Gtk themes in Openbox:

1. Use LXappearance, Gtk-theme-switch or Gtk-chtheme

[LXappearance](#), [Gtk-theme-switch](#) and [Gtk-chtheme](#) are three small applications that enable you to change the Gtk themes and fonts. Of these three, LXappearance has the most options: it can set the Gtk themes, fonts and icons. Gtk-chtheme is a little more polished than Gtk-theme-switch, though Gtk-theme-switch allows you to change both Gtk-1.2 themes (using 'switch' as command) and Gtk-2.0 themes (using the command 'switch2'). Both allow you to change the Gtk theme and font, but neither allows you to change the icon theme, for which you will have to use the second method (below). All three offer a preview window. They save the theme information in the `~/.gtkrc-2.0` file (see below).

Gtk-chtheme and Gtk-theme-switch are in the Gutsy and Hardy repositories. Earlier versions of Ubuntu only have Gtk-theme-switch in the repositories. LXappearance can be downloaded from [Sourceforge.net](#).

2. Use the gtkrc-2.0 and gtkrc.mine files

You can force all Gtk applications to use a particular theme, icon and font by specifying those in the `/home/USERNAME/.gtkrc-2.0` and `/home/USERNAME/.gtkrc.mine` files.

To specify a Gtk theme, add the following to your `~/.gtkrc-2.0` file (create it if it doesn't exist):

```
# -- THEME AUTO-WRITTEN DO NOT EDIT
include "/path/to/your/themes/gtkrc/file"
include "/home/USERNAME/.gtkrc.mine"
# -- THEME AUTO-WRITTEN DO NOT EDIT
```

For themes you installed in your home directory, the path to your gtkrc file is generally `/home/USERNAME/themes/THEMENAME/gtk-2.0/gtkrc`. For themes installed system-wide (as with themes installed through Synaptic or apt-get), this is `/usr/share/themes/THEMENAME/gtk-2.0/gtkrc`.

In this file, you can also specify what font or icon theme Gtk applications should use. If you use Gtk-chtheme or Gtk-theme-switch (as in the first method) it is best to specify your icon and font details in the `~/.gtkrc.mine` file (in `/home/USERNAME/`), though, as changing the theme with these applications overrides the settings in the `~/.gtkrc-2.0` file, but leaves the `~/.gtkrc.mine` untouched. Make sure you specify the path to your `~/.gtkrc.mine` in your `~/.gtkrc-2.0` (as shown above) and add the following to that file (create it if it doesn't exist yet):

```
style "Sans"
{
font_name = "Sans 10"
}
widget_class "*" style "Sans"

gtk-font-name = "Sans 10"
gtk-icon-theme-name = "nameoficontheme"
gtk-toolbar-style = GTK_TOOLBAR_ICONS
```

Change 'Sans' and '10' with the font name and size of your choice. Make sure the name of your icon theme matches the name of the folder where they are stored, either in `/home/USERNAME/.icons` or `/usr/share/icons/`, or it will not display.

The last line allows you to change the toolbar style of Gtk applications. You can choose from the following options (you can remove the comments):

```
gtk-toolbar-style = GTK_TOOLBAR_ICONS #Only icons
gtk-toolbar-style = GTK_TOOLBAR_TEXT #Only text
gtk-toolbar-style = GTK_TOOLBAR_BOTH #Both icons and text; default, used if nothing is specified.
gtk-toolbar-style = GTK_TOOLBAR_BOTH_HORIZ #Icons and text next to the icons
```

If you want smaller icons in the toolbar, add the following line:

```
gtk-toolbar-icon-size = GTK_ICON_SIZE_SMALL_TOOLBAR
```

If unspecified, the icons in the toolbar will be large. For more Gtk options, see the [Gtk+ Reference Manual](#)

If you want to easily change or add keyboard shortcuts (accelerators) to entries in the menus of Gtk applications by pressing the new keybinding while you select the menu entry, add the following to that file:

```
gtk-can-change-accels = 1
```

Note that the settings in the `~/.gtkrc-2.0` file will affect all Gtk applications run with your username, even when logged in another window manager or desktop environment.

3. Use the Xfce settings

If you add '[xfce-mcs-manager](#) &' to your autostart file (see [below](#)), all Gtk applications will use the Gtk theme, icons and fonts specified in the Xfce settings dialog. To change these, just start 'xfce-setting-show' and change everything you want. (To make things easy, add an entry in your Openbox menu for this)

Note that these changing these settings will also change your Gtk settings in Xfce.

4. Use the Gnome settings.

If you add '[gnome-settings-daemon](#) &' to your autostart file (see [below](#)), all Gtk applications will use the Gtk theme, icons and fonts specified in the Gnome settings dialog. To change these, just start 'gnome-control-panel' (To make things easy, add an entry in your Openbox menu for this).

Gnome-settings-daemon will also set your wallpaper. If you prefer gnome-settings-daemon not to handle this, you can disable this with gconf-editor, under `/apps/gnome_settings_daemon/plugins/background/active` (untick the box), or with the following command:

```
gconftool-2 --set /apps/gnome_settings_daemon/plugins/background/active --type bool False
```

Gnome-settings-daemon controls more than just the Gtk theme and the wallpaper: it also governs keyboard settings, mouse settings, etc. Gnome-settings-daemon will therefore override any changes you try to make to these settings with commands like `xset` in your [autostart.sh file](#), though you can change those settings through gconf-editor. Note also that these changing the settings of gnome-settings-daemon will also change your Gtk settings in Gnome.

5.3 Qt themes and icons in Openbox

To control and change the theme settings of Qt or KDE applications, you can use [Qt3-qtconfig](#) (for Qt3) or [Qt4-qtconfig](#) (for Qt4). It allows you to change the Qt theme and fonts. The theme configuration options it offers are rather limited, though, and you can also not use it to change the icon theme. For more control, you'll have to rely on the robust [Kcontrol](#), which governs every KDE setting (but note that this carries some heavy dependencies!).

Qt themes are more complex than Gtk themes, and are therefore harder to change through text files than Gtk themes are. But if you want to edit the Qt theme settings manually, you can do so in `~/.qt/qtrc`. In that folder, you can also specify or edit the Qt theme engine settings. You can change the icons in `~/.kde/share/config/kdeglobals`.

5.4 More on Fonts

Earlier I mentioned how you can set the Openbox, Gtk and Qt fonts, but if you don't use gnome-settings-daemon or xfce-mcs-manager, these fonts could look rather ugly. How can you have sharper, antialiased fonts without the use of these daemons?

The font configuration for X is done through a single xml file, either `/etc/fonts/fonts.conf` (default, system-wide) or in `~/.fonts.conf` (you can create this file if it doesn't exist yet). The Arch Linux Wiki has [an excellent page](#) detailing what to do with this file; at the [bottom](#) of that page, you'll also find a sample fonts.conf file you can experiment with.

If you want to use the artwiz fonts in your Gtk, Qt or Openbox themes, [this post](#) will help you install these fonts.

5.5 Mouse cursor themes

What do you do when you want to change the mouse cursor theme? You can do this in two ways: system-wide, or for a single user.

If you install an X cursor theme, such as [dmz-cursor-theme](#), system-wide (in `/usr/share/icons`) you can change the default X cursor theme with the command: `sudo update-alternatives --config x-cursor-theme`, which will show you a list of available cursor themes. Select the one you like, and when you restart X the new cursor theme should be used. The same is also accomplished if you edit `/usr/share/icons/default/index.theme` and add or adjust the following lines (using the cursor theme DMZ-White as an example):

```
[Icon Theme]
Inherits=DMZ-White
```

You can also specify the mouse cursor theme in `/home/USERNAME/Xdefaults` to change the theme for a single user only. To change the mouse cursor theme in this way, add the following to that file:

```
Xcursor.theme=NAMEOFTHEHERE
Xcursor.size: SIZE #optional
```

Change the mouse Xcursor:theme to whatever your preferred theme is named. Some cursor themes support can display more than one size; if you use such a theme, you can specify the size with the second line (normal sizes are 32, 48 or 64). If the cursor theme only has a single size, this line is meaningless.

You can install cursor themes in `/home/USERNAME/.icons` or `/usr/share/icons/`. Make sure the name you specify in `~/.Xdefaults` matches the name of the folder the theme is stored in (as always case sensitive). When you restart X, the new cursor theme should be applied.

If you prefer graphical tools, you could try out [gcursor](#), a Gtk application to change cursor themes (with previews).

6. Autostarting applications

To start applications automatically when Openbox starts is very simple. You can achieve this by editing the autostart file. If you haven't done so already, create a file called `autostart.sh` in `/home/USERNAME/.config/openbox`.

In this file you can add any applications that you want to be started whenever openbox-session is started. If you do not have such a file in `>/config/openbox/` Openbox will launch the default system-wide `autostart.sh` script, located at `/etc/xdg/openbox/`. The default autostart file loads quite a lot. If sets a plain colour background, loads either `gnome-settings-daemon` or `xfce-mcs-manager` if you have them installed to govern your wallpaper and Gtk settings, and launches `kdeinit` if you have KDE installed so that your KDE applications start faster. If you desire, you can start the default autostart file with the following command: `.$GLOBALAUTOSTART &`.

I generally build up my own `autostart.sh` file and load only those things that I need. Here is what a typical `autostart.sh` file on my computer looks like:

```
nametack &
.HOME/.fehbg &
docker &
lat &
bbpager &
gtodo &
```

Make sure that you have a `&` after each entry, or any program after it will not run!

If you want your disks and removable media to be mounted automatically, you can use [Thunar's volume manager](#) (`thunar-volman`), [gnome-volume-manager](#) (as in the above example), or [ivman](#) using the command `"ivman &"`. If you use Thunar as your file manager, `thunar-volman` is the lightest as it doesn't require an extra daemon running but is activated automatically whenever you run Thunar. If you want Thunar to load faster, you can launch it as a daemon with this file using the command `thunar -daemon &`.

If an application mentioned in your `autostart.sh` file does not load when you start Openbox, you might want to try moving it up or down in the list. Sometimes an application loads best before or after another application is loaded. You can also use the `'sleep'` option, and force the application to wait some seconds before loading. This is especially useful for panels, which can easily pull Openbox down if they do not load properly. Here is an example:

```
(sleep 3 && pypanel) &
```

Pypanel will now load with a delay of three seconds, i.e. after Openbox has fully loaded.

If you use the Xfce or Gnome settings for your Gtk themes, you'll want to add either of the following two lines as well:

```
gnome-settings-daemon &
xfce-mcs-manager &
```

The above example illustrates how to set the wallpaper using Feh (see [infra](#)). If you prefer to use Nitrogen, add this command:

```
nitrogen --restore &
```

If you run a lot of KDE applications in Openbox, it might be worth it to add the following to your autostart list:

```
kdeinit &
```

If you'd like your dockapps to start in a particular order, read [this](#).

Note that when you use Openbox as a window manager in Gnome, KDE or Xfce, Openbox's `autostart.sh` file is not used. You'll have to use the autostart options of the desktop environment you are using.

7. Managing the Desktop — Wallpapers and Icons

7.1 Wallpapers

Openbox itself does not manage the desktop, but using other applications, you can set your wallpaper in many different ways. I will illustrate two methods here at length - Feh and Nitrogen - as these are the most popular. More are mentioned at the end of this section.

7.1.1 Wallpapers with Feh

Feh is a very fast and light image viewer that is also capable of setting the wallpaper. This is a simple method, but my favourite. First we need to install Feh:

```
sudo aptitude install feh
```

Once you have Feh installed, you can set a wallpaper with Feh using one of the following commands:

-

```
To scale the image:
feh --bg-scale "/path/to/image"
```

-

```
To tile the image:
feh --bg-tile "/path/to/image"
```

-

```
To centre the image:
feh --bg-center "/path/to/image"
```

If you use Feh also as an image viewer, open the image, right click on it and go to File > Background and select your preferred option. If you use Thunar, you can also create a custom action (Edit > Configure custom action >). Create a new action called 'Set as Background' and use the following command: `feh --bg-scale %f` (or `tile/center/etc.`) and select 'Image files' in the 'Appearance condition' tab. Now you'll just have to right click on an image file in Thunar and select the 'Set as Background'.

To make sure Feh loads the wallpaper again when you start Openbox, add this to your autostart file (see [above](#)):

```
.HOME/.fehbg &
```

If you would like to use a graphical user interface in combination with Feh, you can use [CWallpaper](#) or the simpler [Oboinus](#). For more information on CWallpaper, have a look [here](#) (little over half way down).

7.1.2 Wallpapers with Nitrogen

[Nitrogen](#) is graphical application that allows you set a desktop wallpaper, and it provides previews of the wallpapers. Nitrogen is only in the Ubuntu repositories from Intrepid onwards. If you use an earlier version of Ubuntu you can easily build it from source. First you'll have to install the dependencies:

```
sudo aptitude install libgtkmm-2.4-dev libgtk2.0-dev libglb2.0-dev
```

Download the latest version of Nitrogen [here](#). Extract the archive, configure, make and install:

```
tar -xvzf nitrogen-1.4.tar.gz
cd nitrogen-1.4
./configure --prefix=/usr
make
sudo checkinstall
```

To run the application use the command `'nitrogen /path/to/your/wallpapers/directory'`, and select the wallpaper you like. To restore the last used wallpaper when you log into Openbox, add this to your autostart file:

```
nitrogen --restore &
```

7.1.3 Other tools

There are other applications as well you can use to set a desktop wallpaper, such as [hsetroot](#), [Habak](#), or [xli](#). See [this post](#) for more information.

7.2 Icons on the desktop

To have icons on the desktop, you can use [iDesk](#), though it only supports launchers. There is a graphical configuration tool for iDesk available, called [iDesktools](#) that should make using iDesk even simpler, but I've never managed to get it to work properly. Others, though, have had more success.

If you want the files and directories that are in your `/home/USERNAME/Desktop/` folder to show on the desktop, you can use a file manager that draws the desktop, such as [PCMan-FM](#), [Rox-filer](#), or Gnome's [Nautilus](#). Note that when some file managers, like Nautilus, draw the desktop you will also lose your right and middle click Openbox menus on the desktop.

8. Panels and the like

Unlike Fluxbox, Blackbox or Icewm, Openbox does not come with a panel. If you do want to use a panel, you'll have to add one yourself, but, since you have a wide selection of panels to choose from, this allows you to have everything just the way you want it.

8.1 Panels

The following are some panels that are popular with Openbox users:

- **Pypanel:** A very minimalistic panel, that is light and quite configurable. Supports launchers, a system tray, a clock with time and/or date, and a workspace indicator. I give some tips on what to do with Pypanel in [this post](#).
- **Fbpanel:** A light panel with many plug-ins (paggers, task bar, menu, clock, etc.) that follows your Gtk theme and supports transparency. You can run [two panels](#) at the same time (top & bottom), Gnome-style.
- **Lxpanel:** A light panel with all the things that normally come with a panel. Lxpanel is similar to fbpanel, but easier to configure. It is part of the the [LXDE](#) project.
- **Perlpanel:** As the name suggests, perlpanel is written in perl. It is heavier on resources than the above mentioned panels, but has a lot of extensions that can be loaded.

- **BMpanel:** A lightweight panel with a taskbar, system tray, desktop switcher and a clock. The panel is themeable. It is based on [fspanel](#), which is no longer developed.
- **Barpanel:** A themeable panel with lots of plugins and nice menus. It is a continuation of [Foopanel](#). I've had some issues with this panel (it only showed up on one workspace), but others seemed to have had more luck with it.
- **Xfce4-panel:** The panel that is used in Xfce. Everything will work in Openbox, including all the [applets](#) (clocks, pagers, launchers, etc.), except for the power options (shut down, restart, etc.).
- **Gnome-panel:** The panel that is used in Gnome. Everything will work in Openbox, including all the applets (clocks, pagers, launchers, etc.), except for the power options (shut down, restart, etc.).
- **Kicker:** The panel that is used in KDE. Everything will work in Openbox, including all the applets (clocks, pagers, launchers, etc.), except for the power options (shut down, restart, etc.).

8.2 Task lists

There are also some applications that offer only a task list:

- **Tint2:** Tint2 is a continuation of [Tint task manager](#); a text-only task list that is very configurable in looks. Tint2 added more configuration options, as well as a clock to it. It seems launcher icons are also planned for future releases.
- **visibility:** Visibility is a simple icon-box task list. A [python/gtk version](#) is also available, which will follow your Gtk2 theme.

Instead or in addition of a task list, you can also use [Skippy](#), a full-screen task-switcher that shows you the (non-iconified) windows that are running in your current workspace.

8.3 System trays

For an system system tray/notification area, you can either use those that come with one of the above panels, or a stand-alone system tray (some of which can appear in Openbox' dock). There are several of these available:

- **docker:** A very simple and very light system tray that runs in the dock. It is my favourite system tray: does what it needs to do well, without any complications.
- **peksysstray:** A system tray that appears in the dock, and is a bit more configurable than docker.
- **stalonetray:** A stand alone tray, that is somewhat configurable, and can be placed in the dock (if you run it in the withdrawn mode: *stalone -w*)
- **traver:** A lightweight GTK2-based system tray, extracted from [fbpanel](#)'s code. It supports transparency.

If you use [GKrellm](#), you can use the [gkrellm travicons plugin](#) to add a system tray to it.

8.4 Launchers and docks

If you want icons to launch the programs of your choice, here is what you can use (besides panels with launchers):

- **Whar:** Whar is a quick launch bar that is fairly configurable. It doesn't autohide (yet), so don't spend hours looking for that option.
- **WMdrawer:** A drawer with launchers that goes in the dock. Like most dockapps, it is quite ugly with its default settings, but its looks can easily be changed.
- **Apwal:** A simple icon-box application launcher. It is quite neat and fast and gives you quick (graphical) access to whatever commands you configure.
- **Tablaunch:** A quick launch bar that is configurable and remains hidden when not used. It is based on [yeahlaunch](#).
- **Simdock:** A light and customizable dock, that doesn't require compositing to work. It seems still in early stages of development, but is functional.
- **Tabble:** Tabble is "a small alternative X shell which presents your programs in tabs for easy launch" (have a look at the picture on the website; that will explain better what Tabble actually is). You can use it as a light-weight graphical menu, if you add it to your panel.
- **BBDock:** BBDock is an application launcher that loads into the dock. You can configure it so that clicking on already launched icons will raise the corresponding application instead of invoking a new instance. The default size of the icons (no matter what size your image file is) is 64x64. If you'd like it smaller, read [this post](#).

Docks that require compositing, such as [Avant-window-navigator](#), [Cairo dock](#), or [Kooldock](#), can be used in Openbox with [xcompmgr](#) (see [below](#)). (Since a lot of new users (including me!) are terribly confused by this, let me clarify: the Openbox 'dock' (which is configured through Obconf) is nothing like these docks. It is a place where [dockapps](#) are loaded, and that is its only use. Have a look [here](#) for more information.)

8.5 Pagers

For pagers, you can either use a panel that has a pager, or use a standalone pager. Here are a few standalone pagers:

- **Obpager:** Obpager is a pager specifically designed for Openbox. I've never liked it, as I find the pager squares too big, but tastes vary. It is in the repositories.
- **Netwmpager:** Netwmpager is a simple pager application that can be run on the desktop or on top of all other windows. It is my favourite pager. Its size, colours and basic behaviour can be changed in a simple text file. Netwmpager supports transparency if xcompmgr is running. You can download it [here](#).
- **Ipager:** A transparent pager with some nice touches. It has an (optional) mouse-over effect, and its colour and size can be configured. It is not in the Ubuntu repositories, and despite what the website says, you don't need Fluxbox installed to compile this.
- **BBpager:** This is the Blackbox pager. BBpager can run in the dock, and allows some basic configuration: it can use whatever colours and size you want it to use, and the number of rows and columns can be changed. It is in the repositories.
- **Minipager:** A very minimilastic pager that represents the desktops as very small coloured squares.

8.6 Clocks

If you want to know what time it is in Openbox, you again have plenty of choice.

1. Use a panel that has a clock to display the time
2. Use [conky](#) or [gkrellm](#) to display the time (If you want conky to remain always visible, you could adjust Openbox's margin settings in Obconf)
3. Use a dock app. There are several clocks available that you can load into Openbox' dock. The best available, in my opinion, is [Lal](#), a great simple dock clock that is light, and somewhat configurable. You can also use [hbtime](#) and/or [hbdate](#), two very simple Blackbox apps that display the time and date respectively; if you run them in a withdrawn mode ('hbtime -w' or 'hbdate -w') they will appear in the dock. There are also several WindowMaker dockapps available, though most are ugly. Have a look [here](#) and [here](#).
4. Use xclock. To make it more attractive, you can configure its colours and looks (have a look at 'man xclock'), and set it to display without window decorations (change the settings in Openbox' rc.xml file).
5. Use a desklet. Both [gdesklets](#) and [adesklets](#) have clock desklets.
6. Display the time in the root menu of Openbox: use the pipe menu mentioned in [this thread](#) on the Ubuntuforums
7. Use [Osdsh](#) to display the time. Use the command `osdctl -c 1` and `osdctl -c 0` to turn the clock on and off. Run `osdshconfig` to configure colours, fonts and position on the screen.
8. Place a clock next to your computer.



9. Shadows, Fade-in/out menus, and more fancy stuff

9.1 Shadows and Fading menus

If you thought that drop shadows, fading menus and true transparency were only possible in heavyweight desktop environments, here is a surprise. All of this is possible in Openbox (provided your hardware supports it - check [this thread](#) first) when you use `xcompmpgr`. Here is how you do it:

Install `xcompmpgr` through Synaptic or apt:

```
sudo aptitude install xcompmpgr
```

Enable compositing in your Xorg file. Open a terminal and open the Xorg file:

```
sudo your_text_editor /etc/X11/xorg.conf
```

and add the following to the end of the file:

```
Section "Extensions"
    Option "Composite" "Enable"
EndSection
```

Save and close the file. You'll now have to restart X (the graphical environment). Make sure you have everything saved that is still open, and press Ctrl+Alt+Backspace. You will be back at your login screen. Once you are back in Openbox, you can start using `xcompmpgr`. You have plenty of options to experiment with. Have a look at [jdong's thread](#) about `xcompmpgr` on the Ubuntu forums; you'll find it very useful in finding the right `xcompmpgr` settings for you (drop shadows, acceleration, fading menus, etc.). I prefer the following settings, which only displays drop shadows:

```
xcompmpgr -cC -t-3 -l-5 -r5
```

`Xcompmpgr` is quite light. The above settings consume about 1-2 MB on my laptop. If you'd like a pipe-menu to control `xcompmpgr` from your Openbox menu, [this](#) will make you happy.

`Xcompmpgr` may interfere with `conky`. You'll need the following settings in your `~/conkyrc` file to run `conky` and `xcompmpgr` together without any issues:

```
own_window yes
own_window_transparent yes
own_window_type desktop
own_window_hints undecorated,below,sticky,skip_taskbar,skipPager
doubleBuffer yes
```

9.2 Transparency

But what about true transparency? For that you'll need an additional application: [transset](#). Install the application:

```
sudo aptitude install transset
```

Make sure `xcompmpgr` is running, and then run `transset` by typing:

```
transset
```

Your cursor will change, and the next window you click on will become transparent. You can change the level of transparency by adding a value after the command (0 is fully transparent, 1 is fully opaque):

```
transset .4
```

If you want to bind `transset` to the scroll button on your mouse, you'll need [transset-df](#). It is not in the repositories, so you'll have to compile it from source. Download the source code [here](#), extract it and install it:

```
tar xzvf transset-df-6.tar.gz
cd transset-df-6
make
sudo checkinstall
```

You can bind transset to the mouse wheel (thanks [K.Mandla](#)) if you add the following to your rc.xml in the mousebind section (make sure you haven't assigned the mouse binding to something else already)

```
<mousebind button="C-Up" action="Click">
<action name="Execute">
<execute>transset-df -p --inc 0.2 </execute>
</action> </mousebind>
<mousebind button="C-Down" action="Click">
<action name="Execute">
<execute>transset-df -p --min 0.2 --dec 0.2</execute>
</action></mousebind>
```

The first of these makes Ctrl+mousewheel-up on the titlebar of a window increase the opacity of the window, the second makes the window grow more transparent when you do Ctrl+mousewheel-down on the titlebar of the window.

9.3 Desktop cubes

If you like desktop-cubes (when you change the workspace), you can play around with [3ddesktop](#). Install it:

```
sudo aptitude install 3ddesktop
```

Add the preferred keybindings to the rc.xml file (Ctrl+Alt+Left and Right Arrows, or whatever else you like to use to change workspaces), and use the command 3ddesk. Note that it doesn't run too well if you are on older hardware or have limited RAM, especially when used with xcompmgr and transset.

10. File Managers

You can use any file manager you like in Openbox. [Thunar](#), [PCMan-FM](#), [Rox-filer](#), [Xfce](#), or [Nautilus](#) seem to be popular graphical file managers with Openbox users. If you are a KDE fan, you can also use your favourite KDE file manager: [Konqueror](#), [Dolphin](#) and [Krusader](#) all run fine in Openbox. If you want to use Nautilus, make sure it doesn't draw the desktop, as you will lose your Openbox desktop menu. To prevent this, run Nautilus with the following command:

```
nautilus --browser --no-desktop
```

Alternatively, you can prevent Nautilus from drawing the desktop in gconf-editor, under apps/nautilus/preferences/show_desktop (untick the box), or with the following command:

```
gconftool-2 --set /apps/nautilus/preferences/show_desktop --type bool False
```

11. OpenOffice in Openbox

When you load OpenOffice in Openbox, it will look very ugly (grey with huge ugly blueish icons and big fonts). To change this, and make OpenOffice follow your Gtk theme, do the following:

Install [openoffice.org-gtk](#) (sudo aptitude install openoffice.org-gtk) and add this to the very beginning of your autostart file (right after #!/bin/sh)

```
export OOO_FORCE_DESKTOP=gnome
```

When you login next time, OpenOffice should follow your general Gtk theme.

If you prefer KDE or Qt styles, you can force OpenOffice to use the Qt widgets. To do so, install [openoffice.org-kde](#) (sudo aptitude install openoffice.org-kde), and add the following line to your autostart file:

```
export OOO_FORCE_DESKTOP=kde
```

12. Tile windows in Openbox

If you want to be able to tile windows in Openbox, here is what you have to do:

You can use a small application, called 'tile', to tile windows in any window manager. Download the appropriate .deb package from the [tile website](#) (that website seems to be down frequently, so I've uploaded the source code [here](#) and the deb package for i386 [here](#); rename the files to remove the pdf extension and turn them into .tar.gz and .deb files respectively).

Double click on the package to install, or use the terminal and install it with the following command (make sure you are in the right directory!):

```
sudo dpkg -i tile_0.7.4_i386.deb
```

Once installed, all you need to tile windows is use the command 'tile'. Add a menu entry or a keyboard binding with that command, and your windows are tiled instantly.

Alternatively, you can use [whaw](#) to tile a selected number of windows either horizontally or vertically, using either the entire screen, or a selected area of it.

Note that cascading of windows (un-tiling) is not possible, as far as I know. Neither tile nor whaw support it.

13. Shut down and reboot, hibernate, suspend and locking the screen

13.1 Shut down, reboot, and suspend

Gnome, KDE and Xfce are all able to shut down, restart, hibernate or suspend your computer without administrative privileges. Window managers, such as Openbox or Fluxbox, are unable to do this on their own. Even if you use gnome-panel or xfce4-panel, you won't be able to use the power buttons that come with it (they depend on gnome-sessions and xfce4-sessions to work if I've understood correctly). Advanced shutdown applications like [gshutdown](#) or [kshutdown](#) won't work either.

Shutdown, reboot and suspend using gdm-control

In the latest Openbox release (3.4.7), you can shutdown, reboot or suspend from within Openbox with *gdm-control*, by telling [GDM](#) (The Gnome Display Manager) to perform these actions.

To shutdown the computer with gdm-control, use the following command:

```
gdm-control --shutdown && openbox --exit
```

To restart, use this command:

```
gdm-control --reboot && openbox --exit
```

And to suspend, use the following line:

```
gdm-control --suspend
```

If you'd like a little dialog for these options, have a look [here](#) (using PyGtk), or modify [this](#) appropriately (using Gmessage).

Other options

If you are not using GDM, are not using the latest version of Openbox, or prefer not to use this method, there are two other ways in which you can perform these tasks: one that requires administrative privileges, one that doesn't.

1. Shut down and reboot with administrative privileges

Add the following lines to your menu.xml file, wherever you want them to be in your menu. If you want to use Obmenu, add what is specified in the "Label" part in the "Label" box and what comes between the "execute" tags in the "Execute" box.

```
<item label="Reboot">
<action name="Execute"> <execute>gksudo 'shutdown -r now'</execute> </action> </item>
<item label="Shut down">
<action name="Execute"> <execute>gksudo 'shutdown -h now'</execute> </action> </item>
<item label="Suspend">
<action name="Execute"> <execute>gksudo pm-suspend</execute> </action> </item>
<item label="Hibernate">
<action name="Execute"> <execute>gksudo pm-hibernate</execute> </action> </item>
```

The good thing about this method is that you get some sort of confirmation before you shut down or reboot, as you'll be asked to give your sudo password first. This might be useful to prevent your computer from shutting down unexpectedly when you accidentally press the Shut down entry in your menu. Note that you need [gksu](#) installed for this to work.

2. Shut down and reboot without administrative privileges

If you don't want to have to give your sudo password every time you shut down or restart, you can edit the sudoers file (in /etc/sudoers) to allow anyone to shut down or reboot the computer without administrative privileges. The sudoers file specifies who has administrative privileges on your computer and what user can do what. You'll edit the file so that any user can use the 'shutdown' command without the need to give your sudo password. To do so, open a terminal and type

```
sudo visudo
```

(It is best to edit the sudoers file with visudo, as visudo will check for possible errors before saving, thus preventing you from messing up your system)

Add the following at the bottom of the page:

```
ALL ALL=NOPASSWD:/sbin/shutdown
```

Save and exit, and you won't be needing your password to reboot or shut down. You can then add the following to your menu.xml file, wherever you want them to be in your menu. If you want to use Obmenu, add what is specified in the "Label" part in the "Label" box and what comes after "execute" in the "Execute" box.

```
<item label="Reboot">
<action name="Execute"> <execute>sudo 'shutdown -r now'</execute> </action></item>
<item label="Shut down">
<action name="Execute"> <execute>sudo 'shutdown -h now'</execute> </action></item>
```

You will no longer be asked to give your password to shut down or reboot now, or be a user with administrative privileges. (Note, though, that you'll still need to preface the command with sudo!) The downside is that you also will no longer receive any warning before your computer shuts down. If you would like a confirmation window that comes up whenever you want to shut down, have a look at [this post](#).

13.2 Lock screen

You can easily add an entry in your Openbox menu or assign a keyboard binding in the rc.xml file to lock the screen. You can use [xlockmore](#) for this. Read the [man page](#) to see the many ways in which you can use this. I prefer a simple black screen (no animation) that asks for your password when a key is pressed or the mouse is moved, and use the following command for it:

```
xlock -mousetion +description -mode blank -bg black -fg grey30 -font '-adobe-helvetica-bold-r-normal-*-*110-*-*-*-*-*' -planfont '-adobe-helvetica-bold-r-normal-*-*110-*-*-*-*-*' -timeout 6 -info " " -username " " -password " " -icongeometry 30x30
```

If you use xscreensaver, you can also use that to lock the screen with the following command:

```
xscreensaver-command -lock
```

To use this, you need xscreensaver running, so make sure you add a line for it to your autostart file (cf. [supra](#)):

```
xscreensaver -no-splash &
```

14. More useful applications

There are several other useful applications that can be used with Openbox. Here are a few of them:

14.1 System monitors

In Openbox you can use the system monitor from [Gnome](#) or [KDE](#), but if you'd like something a lot more lighter have a look at [Gnemosage](#), [gPS](#), [gTaskmanager](#), or [htop](#).

There are also quite a few applications available that enable you to display system information, such as cpu and memory usage, hard disk and network status, etc., on the desktop, in Openbox' dock, or in a panel-like form. Here are a few of these:

- [Conky](#)

Conky is a light-weight system monitor that can display system information (such as uptime, kernel version, cpu and memory usage, etc) on the desktop. It is highly configurable. For examples of conky setups, have a look at [this thread](#) on the Ubuntuforums. It is in the repositories.

- [Dzen2](#)

Dzen is a "general purpose messaging, notification, and menuing application." You can use it to display all sorts of information. Have a look at the [wiki](#) to comprehend its potential. Dzen2 is in the Hardy repositories.

- [Gkrellm](#)

Like conky, Gkrellm is a light-weight system monitor. It looks a little less nice than conky can look, but you can use [themes](#) to change its looks (use the 'Invisible' theme if you would like to see only the stats and no borders). There are a lot of plugins available. Gkrellm is in the repositories.

- [Light-monitor](#)

Light-monitor is a light transparent panel-like system monitor that can display and/or manage the volume, cpu and memory usage, network traffic, battery and power status, as well as the date and time. A component of light-monitor is light-calendar, which you can use to display a transparent calendar on your desktop (which allows you to also cycle through the next and previous months).

- [Noteo](#)

Noteo is a little app that runs in the system tray and displays notifications. It can notify you of the status of your laptop battery, emails in your Gmail account, what mpd is playing, etc.

14.2 Other applications

- [Backstep](#)

Backstep draws icons of your iconified applications on your desktop. If enabled, it can also draw miniature versions of the iconified windows on the desktop. It is a bit rough, and since long no longer developed, but it works. It is in the repositories.

- [Dmenu](#)

Dmenu is the application menu developed for dwm and used by several other tiling window managers. It can easily be used in Openbox as well; read [this post](#) for more info on using Dmenu in Openbox. Dmenu can be used for a wide variety of things: see, for example, [here](#) for how to use dmenu with mpd, or [this post](#) on how to easily access your configuration files with it.

- [Gmrun](#)

Gmrun is a small run dialog, in which you can enter commands, application names, or files to run or open. You can easily bind it to Alt+F2 in your rc.xml file, to have it work like the run dialog in Gnome, KDE or Xfce. It is in the repositories.

- [Gtk-tray-utils](#)

Gtk-tray-utils is set of little Gtk utilities that load in the system tray. It consists of gbatt (a battery status indicator), gtim (a clock), gdat (a date indicator) and gtray (a volume control). David Barr (mulberry) made a [modification of gtray](#) you might be interested in.

- [OSDsh](#)

OSDsh (and its component osdctl) is a little application that enables you to display all battery status, a clock, volume levels or any text on your screen. It is themeable, allowing you to change the font, colours, and position on the screen. Unfortunately, the volume control applet causes your CPU to be fully consumed on recent versions of Ubuntu.

- [Parcellite](#)

Parcellite is a light and simple Gtk based clipboard manager. There is a deb package available at [GetDeb.net](#).

- [Unclutter](#)

Unclutter is an application that runs in the background, and hides the mouse cursor when it has not moved and no buttons are pressed and the mouse is not on the desktop. This is a useful tool if you like things uncluttered. It is in the repositories.

- [VolWheel](#)

VolWheel is a little application which lets you control the volume through a tray icon you can scroll on. It is written in Perl/Gtk2. You can download it [here](#).

- [Wmctrl](#)

Wmctrl is a command-line application that can interact with EWMH/NetWM compatible X window managers. It doesn't add any functionality to your window manager, but allows you to control it from the command line. For an indication of its potential, see [this post](#). Wmctrl is in the repositories.

- [Xautolock](#)

Xautolock "monitors input devices under the X Window System, and launches a program of your choice if there is no activity after a user-configurable period of time. You can use this to automatically start up a screen locker if you have left your computer unattended for some period of time. The program launched need not be a screen locker such as xlock." See [this post](#) for how to use it with Skippy. It is in the repositories.

15. Conclusion

That's it. Now you are fully ready to use Openbox! Configure it more to your liking and needs, and don't feel afraid to try things out. Openbox is small and light, but is highly configurable. Settle on what you like best.

Where do you go from here? You can visit [Planet Openbox](#), a feed aggregator for blogs with (some) Openbox content, or you can hang out on the Openbox IRC channel and join the active Openbox mailing list (see [here](#) for more info).

If you run into problems, don't be afraid to ask for help. There are a large number of knowledgeable Openbox users on the [Ubuntuforums](#). If you can't find a solution for your problems elsewhere, try the [mailing list](#) or [IRC channel](#).

I'd like to receive feedback on this guide. If you find errors or inconsistencies, if you think sections could be improved on, or you have material that could be added to it, please let me know. Leave a comment, send me a pm on the Ubuntuforums (user [urukrama](#)) or send me an email at *ukdasa AT yahoo DOT com*. Thank you.