

Openbox

Openbox (http://openbox.org/wiki/Main_Page) is a lightweight, powerful, and highly configurable *stacking* **window manager** with extensive standards support. It may be built upon and run independently as the basis of a unique **desktop environment**, or within other integrated desktop environments such as **KDE** and **Xfce**, as an alternative to the window managers they provide. The **LXDE** desktop environment is itself built around Openbox.

Related articles

[Desktop environment](#)

[Display manager](#)

[File manager functionality](#)

[Xdg-menu](#)

[Oblogout](#)

Contents

[Installation](#)

[Starting](#)

[Standalone](#)

[Other desktop environments](#)

[Configuration](#)

[rc.xml](#)

[menu.xml](#)

[Autostart](#)

[environment](#)

[Themes](#)

[Edit or create](#)

[GUI configuration](#)

[Openbox reconfiguration](#)

[Keybinds](#)

[Modifiers](#)

[Multimedia keys](#)

[Volume control](#)

[Navigation keys](#)

[Menus](#)

[Static](#)

[menumaker](#)

[obmenu](#)

[xdg-menu](#)

[logout menu options](#)

[Pipes](#)

Examples

Generators

[obmenu-generator](#)

[openbox-menu](#)

Menu icons

[Desktop menu as a panel menu](#)

[XDG compliant menu](#)

Examples

Tips and tricks

[Cursor and icon themes](#)

[Desktop icons and wallpapers](#)

[Compositing effects](#)

[oblogout](#)

[Openbox for multihead users](#)

[Launch a complex command with hotkey](#)

[Switch desktops using the mouse](#)

[Set default applications / file associations](#)

[Ad-hoc window transparency](#)

[Using obxprop for faster configuration](#)

[Xprop values for applications](#)

[Switching between keyboard layouts](#)

[Set grid layout for virtual desktops](#)

[Enable Hot Corners](#)

[Window snapping](#)

[Smooth display manager transition](#)

[Window Decorations](#)

Troubleshooting

[Firefox](#)

[Missing themes](#)

[Stop continuous desktop switching](#)

[Windows load behind the active window](#)

[See also](#)

Installation

Install the [openbox](https://www.archlinux.org/packages/?name=openbox) (<https://www.archlinux.org/packages/?name=openbox>) package.

Starting

Standalone

Run `openbox` or `openbox-session` with [xinit](#). Note that only `openbox-session` provides [#Autostart](#).

Note: After executing `openbox-session`, there is only a blank grey screen. Try to move your mouse and **right click** to get an openbox menu to make sure that it is actually working.

Other desktop environments

Note:

- When replacing the native window manager of a [desktop environment](#) with Openbox, keep in mind that Openbox does not provide any compositing effects (such as transparency). See [#Compositing effects](#).
- Openbox does work with GNOME applications (but see [GTK+ #Client-side decorations](#)). [\[1\] \(http://comments.gmane.org/gmane.comp.window-managers.openbox/6595\)](#)

See [Desktop environment#Use a different window manager](#).

Configuration

Note: Local configuration files will always override global equivalents.

Four key files form the basis of the [openbox configuration \(http://openbox.org/wiki/Configuration\)](http://openbox.org/wiki/Configuration), each serving a unique role. They are: `rc.xml`, `menu.xml`, `autostart`, and `environment`. Although these files are discussed in more detail below, to start configuring Openbox, it will first be necessary to create a **local** Openbox profile (i.e for your specific user account) based on them. This can be done by copying them from the **global** `/etc/xdg/openbox` profile (applicable to any and all users) as a template:

```
$ cp -R /etc/xdg/openbox ~/.config/
```

rc.xml

Tip: Custom keyboard shortcuts (keybindings) must be added to the `<keyboard>` section of this file, and underneath the `<!-- Keybindings for running applications -->` heading.

`~/.config/openbox/rc.xml` is the main configuration file, responsible for determining the behaviour and settings of the overall session, including:

- Keyboard shortcuts (e.g. starting applications; controlling the volume)
- Theming

- Desktop and Virtual desktop settings, and
- Application Window settings

This file is also pre-configured, meaning that it will only be necessary to amend existing content in order to customise behaviour to suit personal preference.

Note: Per-application settings pertaining to fixed placement of applications per monitor will only work if the x & y position have also been defined.

menu.xml

`~/.config/openbox/menu.xml` defines the type and behaviour of the desktop menu, accessible by right-clicking the background. Although the default provided is a **static menu** (meaning that it will not automatically update when new applications are installed), it is possible to employ the use of **dynamic menus** that will automatically update as well.

The available options are discussed extensively below in the [#Menus](#) section.

Autostart

`openbox-session` provides two autostart mechanisms: [XDG Autostart](#) (which only works if [python2-xdg](#) (<https://www.archlinux.org/packages/?name=python2-xdg>) is installed) and [Openbox's own autostart mechanism](#) (<http://openbox.org/wiki/Help:Autostart>).

Openbox's own autostart mechanism:

- sources `/etc/xdg/openbox/environment`
- sources `~/.config/openbox/environment`
- runs `/etc/xdg/openbox/autostart`
- runs `~/.config/openbox/autostart`

Issues regarding commands in `~/.config/openbox/autostart` being executed out of order (or skipped altogether) are often resolved by the addition of small delays. For instance:

```
xset -b
(sleep 3s && nm-applet) &
(sleep 3s && conky) &
```

environment

`~/.config/openbox/environment` can be used to export and set relevant environmental variables such as to:

- Define new pathways (e.g. execute commands that would otherwise require the entire pathway to be listed with them)

- Change language settings, and
- Define other variables to be used (e.g. the fix for GTK theming could be listed here)

Themes

Install **obconf** (<https://www.archlinux.org/packages/?name=obconf>) and/or **lxappearance-obconf** (<https://www.archlinux.org/packages/?name=lxappearance-obconf>) for a GUI to configure visual settings and theming.

A good selection of themes are available in the **openbox-themes** (<https://aur.archlinux.org/packages/openbox-themes/>)^{AUR} package or the **AUR**. Some **GTK+ #Themes** come with an Openbox theme as well. Both Openbox-specific and Openbox-compatible themes will be installed to the `/usr/share/themes` directory and will also be immediately available for selection.

box-look.org (<https://www.box-look.org/browse/ord/latest/>) is an excellent and well-established source of themes. **deviantART.com** (<http://www.deviantart.com/>) is another excellent resource. Many more can be found online.

Edit or create

Tip: It's better to copy a theme to your home directory than to edit those found in `/usr/share/themes/`. This will retain the original should anything go wrong and ensure that your changes are not overwritten on update.

The process of creating new or modifying existing themes is covered extensively at the official **openbox.org** (<http://openbox.org/wiki/Help:Themes>) website. **obtheme** (<https://aur.archlinux.org/packages/obtheme/>)^{AUR} is a user-friendly GUI for doing so.

GUI configuration

Several GUI applications are available to quickly and easily configure your Openbox desktop.

- **ObConf** — A GTK2 based configuration tool for the Openbox window manager.

<http://openbox.org/wiki/ObConf:About> || **obconf** (<https://www.archlinux.org/packages/?name=obconf>)

- **LXAppearance ObConf** — Plugin for LXAppearance to configure Openbox. Note that not all options to configure Openbox are available in this plugin, so you might want to install obconf anyway.

<http://lxde.org> || **lxappearance-obconf** (<https://www.archlinux.org/packages/?name=lxappearance-obconf>)

- **LXInput** — LXDE keyboard and mouse configuration

<http://lxde.org> || [lxinput \(https://www.archlinux.org/packages/?name=lxinput\)](https://www.archlinux.org/packages/?name=lxinput)

- **LXRandR** — LXDE monitor configuration.

<http://wiki.lxde.org/en/LXRandR> || [lxrandr \(https://www.archlinux.org/packages/?name=lxrandr\)](https://www.archlinux.org/packages/?name=lxrandr)

- **obkey** — Configure Openbox keyboard shortcuts

<https://code.google.com/p/obkey/> || [obkey \(https://aur.archlinux.org/packages/obkey/\)](https://aur.archlinux.org/packages/obkey/)^{AUR}

- **ob-autostart** — A simple autostart application for Openbox.

<http://pastebin.com/012YgXTk> || [ob-autostart \(https://aur.archlinux.org/packages/ob-autostart/\)](https://aur.archlinux.org/packages/ob-autostart/)^{AUR}

- **obapps** — Graphical tool for configuring application settings in Openbox.

<https://sourceforge.net/projects/obapps/> || [obapps \(https://aur.archlinux.org/packages/obapps/\)](https://aur.archlinux.org/packages/obapps/)^{AUR}

Programs and applications relating to the configuration of Openbox's desktop menu are discussed in the [Menus](#) section.

Openbox reconfiguration

Tip: where not already present, it would be worthwhile adding this command to a menu and/or as a keybind for convenience.

Openbox will not always automatically reflect any changes made to its configuration files within a session. As a consequence, it will be necessary to manually reload those files after they have been edited. To do so, enter the following command:

```
$ openbox --reconfigure
```

Where intending to add this command as a keybind to `~/.config/openbox/rc.xml`, it will only be necessary to list the command as `reconfigure`. An example has been provided below, using the `Super + F11` keybind:

```
<keybind key="W-F11">
  <action name="Reconfigure"/>
</keybind>
```

Keybinds

All keybinds must be added to the `~/.config/openbox/rc.xml` file, and below the `<!-- Keybindings for running applications -->` heading. Although a brief overview has been provided here, a more in-depth explanation of keybindings can be found at [openbox.org \(http://openbox.org/wiki/Help:Bindings\)](http://openbox.org/wiki/Help:Bindings).

Keybinds can be added to the configuration file using the following syntax:

```
<keybind key="my-key-combination">
  <action name="my-action">
    ...
  </action>
</keybind>
```

The action name for running an external command is *Execute*. Use the following syntax to define an external command to execute:

```
<action name="Execute">
  <command>my-command</command>
</action>
```

See [the Openbox wiki \(http://openbox.org/wiki/Help:Actions\)](http://openbox.org/wiki/Help:Actions) for a list of all available actions.

Tip: The [obkey \(https://aur.archlinux.org/packages/obkey/\)](https://aur.archlinux.org/packages/obkey/)^{AUR} utility provides a graphical interface for configuring key bindings. Before using *obkey*, you should use *obconf* to create `~/.config/openbox/rc.xml`.

While the use of standard alpha-numeric keys for keybindings is self-explanatory, special names are assigned to other types of keys, such as `modifiers`, `multimedia` and `navigation`.

Modifiers

`Modifier` keys play an important role in keybindings (e.g. holding down the `shift` or `CTRL / control` key in combination with another key to undertake an action). Using modifiers helps to prevent conflicting keybinds, whereby two or more actions are linked to the same key or combination of keys. The syntax to use a modifier with another key is:

```
"<modifier>--<key>"
```

The modifier codes are as follows:

- `S` : Shift
- `C` : Control / CTRL
- `A` : Alt

- W : Super / Windows
- M : Meta
- H : Hyper (If it is bound to something)

Multimedia keys

Where available, it is possible to set the appropriate `multimedia` keys to perform their intended functions, such as to control the volume and/or the screen brightness. These will usually be integrated into the `function` keys, and are identified by their appropriate symbols. See [Extra keyboard keys](#) for details.

The volume and brightness multimedia codes are as follows (note that commands will still have to be assigned to them to actually function):

- `XF86AudioRaiseVolume` : Increase volume
- `XF86AudioLowerVolume` : Decrease volume
- `XF86AudioMute` : Mute / unmute volume
- `XF86MonBrightnessUp` : Increase screen brightness
- `XF86MonBrightnessDown` : Decrease screen brightness

For a full list of XF86 multimedia keys, see the [LinuxQuestions wiki \(http://wiki.linuxquestions.org/wiki/XF86_keyboard_symbols\)](http://wiki.linuxquestions.org/wiki/XF86_keyboard_symbols).

Volume control

What commands should be used for controlling the volume will depend on whether [ALSA](#), [PulseAudio](#), or [OSS](#) is used for sound.

- ALSA: see [Advanced Linux Sound Architecture#Keyboard volume control](#).
- PulseAudio: see [PulseAudio#Keyboard volume control](#)
- OSS: see [OSS#Using multimedia keys with OSS](#).

Navigation keys

These are the directional / arrow keys, usually used to move the cursor up, down, left, or right. The (self-explanatory) navigation codes are as follows:

- Up : Up
- Down : Down
- Left : Left
- Right : Right

Menus

It is possible to employ three types of menu in Openbox: `static` , `pipes` (dynamic), and `generators`

(static or dynamic). They may also be used alone or in any combination.

Static

As the name would suggest, this default type of menu does not change in any way, and may be manually edited and/or (re)generated automatically through the use on an appropriate software package.

Fast and efficient, while this type of menu can be used to select applications, it can also be useful to access specific functions and/or perform specific tasks (e.g. desktop configuration), leaving the access of applications to another process (e.g. the **synapse** (<https://www.archlinux.org/packages/?name=synapse>) or **xfce4-appfinder** (<https://www.archlinux.org/packages/?name=xfce4-appfinder>) applications).

The `~/.config/openbox/menu.xml` file will be the sole source of static desktop menu content.

menumaker

menumaker (<https://www.archlinux.org/packages/?name=menumaker>) automatically generates `xml` menus for several window managers, including Openbox, **Fluxbox**, **IceWM** and **Xfce**. It will search for all installed executable programs and consequently create a menu file for them. It is also possible to configure MenuMaker to exclude certain application types (e.g. relating to **GNOME** or **KDE**), if desired.

Once installed and executed, it will automatically generate a new `~/.config/openbox/menu.xml` file. To avoid overwriting an existing file, enter:

```
$ mmaker -v OpenBox3
```

Otherwise, to overwrite an existing file, add the `force` argument (`f`):

```
$ mmaker -vf OpenBox3
```

Once a new `~/.config/openbox/menu.xml` file has been generated it may then be manually edited, or configured using a GUI menu editor, such as **obmenu** (<https://aur.archlinux.org/packages/obmenu/>)^{AUR}.

obmenu

Warning: `obm-xdg` - a pipe menu to generate a list of **GTK+** and **GNOME** applications - is also provided with **obmenu**. However, it has long-running bugs whereby it may produce an invalid output, or even not function at all. Consequently it has been omitted from discussion.

obmenu (<https://aur.archlinux.org/packages/obmenu/>)^{AUR} is a "user-friendly" GUI application to edit `~/.config/openbox/menu.xml`, without the need to code in `xml`.

xdg-menu

archlinux-xdg-menu (<https://www.archlinux.org/packages/?name=archlinux-xdg-menu>)

will automatically generate a menu based on `xdg` files contained within the `/etc/xdg/` directory for numerous Window Managers, including Openbox. Review the [Xdg-menu#OpenBox](#) article for further information.

logout menu options

Tip: The commands provided can also be attached to [keybinds](#).

The `~/.config/openbox/menu.xml` file can be edited in order to provide a sub-menu with the same options as provided by [oblogout](#). The sample script below will provide all of these options, with the exception of the ability to lock the screen:

```
<menu id="exit-menu" label="Exit">
  <item label="Log Out">
    <action name="Execute">
      <command>openbox --exit</command>
    </action>
  </item>
  <item label="Shutdown">
    <action name="Execute">
      <command>systemctl poweroff</command>
    </action>
  </item>
  <item label="Restart">
    <action name="Execute">
      <command>systemctl reboot</command>
    </action>
  </item>
  <item label="Suspend">
    <action name="Execute">
      <command>systemctl suspend</command>
    </action>
  </item>
  <item label="Hibernate">
    <action name="Execute">
      <command>systemctl hibernate</command>
    </action>
  </item>
</menu>
```

Once the entries have been composed, add the following line to present the sub-menu where desired within the main desktop menu (usually as the last entry):

```
<menu id="exit-menu"/>
```

Pipes

Tip: It is entirely feasible for a static menu to contain one or more pipe sub-menus. The

functionality of some pipe menus may also rely on the installation of relevant software packages.

This type of menu is in essence a script that provides dynamic, refreshed lists on-the-fly as and when run. These lists may be used for multiple purposes, including to list applications, to provide information, and to provide control functions. Pre-configured pipe menus can be installed, although not from the [official repositories](#). More experienced users can also modify and/or create their own custom scripts. Again, `~/.config/openbox/menu.xml` may and commonly will contain several pipe menus.

Examples

- [openbox-xdgmnu](https://aur.archlinux.org/packages/openbox-xdgmnu/) (<https://aur.archlinux.org/packages/openbox-xdgmnu/>)^{AUR}: fast xdg-menu converter to xml-pipe-menu
- [obfilebrowser](https://aur.archlinux.org/packages/obfilebrowser/) (<https://aur.archlinux.org/packages/obfilebrowser/>)^{AUR}: Application and file browser
- [obdevicemenu](https://aur.archlinux.org/packages/obdevicemenu/) (<https://aur.archlinux.org/packages/obdevicemenu/>)^{AUR}: Management of removable media with [Udisks](#)
- [wifi pipe menu](https://bbs.archlinux.org/viewtopic.php?pid=1345031) (<https://bbs.archlinux.org/viewtopic.php?pid=1345031>): Wireless networking using [Netctl](#)

[Openbox.org](http://openbox.org/wiki/Openbox:Pipemenus) (<http://openbox.org/wiki/Openbox:Pipemenus>) also provides a further list of pipe menus.

Generators

This type of menu is akin to those provided by the taskbars of desktop environments such as [Xfce](#) or [LXDE](#). Automatically updating on-the-fly, this type of menu can be powerful and very convenient. It may also be possible to add custom categories and menu entries; read the documentation for your intended dynamic menu to determine if and how this can be done.

A menu generator will have to be executed from the `~/.config/openbox/menu.xml` file.

obmenu-generator

Tip: icons can still be disabled in [obmenu-generator](https://aur.archlinux.org/packages/obmenu-generator/) (<https://aur.archlinux.org/packages/obmenu-generator/>)^{AUR}, even where enabled in `~/.config/openbox/rc.xml`.

[obmenu-generator](https://aur.archlinux.org/packages/obmenu-generator/) (<https://aur.archlinux.org/packages/obmenu-generator/>)^{AUR} is highly recommended despite being an unofficial package. With the ability to be used as a static or dynamic menu, it is highly configurable, powerful, and versatile. Menu categories and individual entries may also be easily hidden, customised, and/or added with ease. The [official homepage](http://trizenx.blogspot.co.uk/2012/02/obmenu-generator.html) (<http://trizenx.blogspot.co.uk/2012/02/obmenu-generator.html>) provides further information and screenshots.

Below is an example of how obmenu-generator would be dynamically executed without icons in

~/.config/openbox/menu.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<openbox_menu>
  <menu id="root-menu" label="OpenBox 3" execute="/usr/bin/obmenu-generator">
  </menu>
</openbox_menu>
```

To automatically iconify entries, the `-i` option would be added:

```
<menu id="root-menu" label="OpenBox 3" execute="/usr/bin/obmenu-generator -i">
```

openbox-menu

Tip: If this menu produces an error, it may be solved by enabling icons in

~/.config/openbox/rc.xml .

openbox-menu (<https://aur.archlinux.org/packages/openbox-menu/>)^{AUR} uses the **LXDE menu-cache** (<http://sourceforge.net/projects/lxde/files/menu-cache/>) to create dynamic menus. The **official homepage** (http://fabrice.thiroux.free.fr/openbox-menu_en.html) provides further information and screenshots.

Menu icons

To show icons next to menu entries, it will be necessary to ensure they are enabled in the `<menu>` section of the `~/.config/openbox/rc.xml` file:

```
<applicationIcons>yes</applicationIcons>
```

Where using a static menu, it will then be necessary to edit the `~/.config/openbox/menu.xml` file to provide both the `icon =` command, along with the full path and icon name for each entry. An example of the syntax used to provide an icon for a category is:

```
<menu id="apps-menu" label="[label name]" icon="[pathway to icon]/[icon name]">
```

Desktop menu as a panel menu

Tip: XDoTool can simulate any keybind for any action, and as such, it may therefore be used for many other purposes...

xdotool is a package that can issue commands to simulate key presses / keybinds, meaning that it is possible to use it to invoke keybind-related actions without having to actually press their assigned keys. As this includes the ability to invoke an assigned keybind for the Openbox desktop menu, it is therefore possible to use

XDoTool to turn the Openbox desktop menu into a panel menu. Especially where the desktop menu is heavily customised and feature-rich, this may prove very useful to:

- Replace an existing panel menu
- Implement a panel menu where otherwise not provided or possible (e.g. for [Tint2](#))
- Compensate where losing access to the desktop menu due to the use of an application like [xfdesktop](https://www.archlinux.org/packages/?name=xfdesktop) (<https://www.archlinux.org/packages/?name=xfdesktop>) to manage the [desktop](#).

Once XDoTool has been installed - if not already present - it will be necessary to create a keybind to access the root menu in `~/.config/openbox/rc.xml`, and again below the `<!-- Keybindings for running applications -->` heading. For example, the following code will bring up the menu by pressing `CTRL + m`:

```
<keybind key="C-m">
  <action name="ShowMenu">
    <menu>root-menu</menu>
  </action>
</keybind>
```

Openbox must then be [reconfigured](#). In this instance, XDoTool will be used to simulate the `CTRL + m` keypress to access the desktop menu with the following command (note the use of `+` in place of `-`):

```
xdotool key control+m
```

How this command may be used as a panel launcher / icon is largely dependent on the features of panel used. While some panels will allow the above command to be executed directly in the process of creating a new launcher, others may require the use of an executable script. As an example, a custom executable script called `obpanelmenu.sh` will be created in the `~/.config` folder:

```
$ text editor ~/.config/obpanelmenu.sh
```

Once the empty file has been opened, the appropriate XDoTool command must be added to the empty file (i.e. to simulate the `CTRL + m` keypress for this example):

```
xdotool key control+m
```

After the file has been saved and closed, it may then be made into an executable script with the following command:

```
$ chmod +x ~/.config/obpanelmenu.sh
```

Executing it will bring up the Openbox desktop menu. Consequently, where using a panel that supports drag-and-drop functionality to add new launchers, simply drag the executable script onto it before changing the icon

to suit personal taste.

XDG compliant menu

A xdg compliant menu is based on the freedesktop.org standard. The menu is defined in menu-files which reside in `/etc/xdg/menus`. New applications will occur automatically in the menu.

Examples

- **californium** (<https://github.com/mlde/californium>): xdg menu based on the LXQt main menu and easily themable

Tips and tricks

Cursor and icon themes

See [Cursor themes](#) and [Icons](#) for details.

Desktop icons and wallpapers

Openbox does not natively support the use of desktop icons or wallpapers.

See [PCManFM](#), [SpaceFM](#) and [Idesk](#).

Note: You may have to edit `~/.conkyrc` and set `own_window_type` to `normal`.

See [List of applications#Wallpaper setters](#).

Compositing effects

Openbox does not provide native support for [compositing](#), and thus requires an external compositor for this purpose.

Although compositing is not a necessary component, it may specifically avoid issues such as screen distortion with [oblogout](#), and visual glitches with terminal window transparency. See [Xorg#Composite](#) for common choices.

oblogout

See the [Oblogout](#) article for an overview on how to use this useful, graphical logout script.

Openbox for multihead users

While Openbox provides better than average multihead support on its own, [openbox-multihead-git](https://aur.archlinux.org/packages/openbox-multihead-git/) (<https://aur.archlinux.org/packages/openbox-multihead-git/>)^{AUR} provides a development branch called **Openbox Multihead** that gives multihead users per-monitor desktops. This model is not commonly found in floating window managers, but exists mainly in [tiling window managers](#). It is explained well on the [Xmonad web site \(http://xmonad.org/tour.html#workspace\)](http://xmonad.org/tour.html#workspace). Also, please see [README.MULTIHEAD](https://github.com/BurntSushi/openbox-multihead/blob/multihead/README.MULTIHEAD) (<https://github.com/BurntSushi/openbox-multihead/blob/multihead/README.MULTIHEAD>) for a more comprehensive description of the new features and configuration options found in Openbox Multihead.

Openbox Multihead will function like normal Openbox when only a single head is available.

A downside to using Openbox Multihead is that it breaks the EWMH assumption that one and only one desktop is visible at any time. Thus, existing pagers will not work well with it. To remedy this, you can install [pager-multihead-git](https://aur.archlinux.org/packages/pager-multihead-git/) (<https://aur.archlinux.org/packages/pager-multihead-git/>)^{AUR}[\[broken link\]](#): archived in [aur-mirror \(https://github.com/felixonmars/aur3-mirror/tree/master/pager-multihead-git\)](https://github.com/felixonmars/aur3-mirror/tree/master/pager-multihead-git) alongside Openbox Multihead. It will work without Openbox Multihead if only one monitor is active.

Launch a complex command with hotkey

If you need to execute a complex command, use shell functionality.

Special character replacement are as follows:

- & : &
- < : <
- > : >

This example will turn off display immediately and lock screen with [slock](https://www.archlinux.org/packages/?name=slock) (<https://www.archlinux.org/packages/?name=slock>). It was taken from [this thread](https://bbs.archlinux.org/viewtopic.php?pid=903057) (<https://bbs.archlinux.org/viewtopic.php?pid=903057>).

```
<keybind key="W-l">
  <action name="Execute">
    <command>sh -c 'slock &amp; (sleep .5 &amp;&amp; xset dpms force off)'

```

Sometimes one need to specify environment variable for application:

```
<keybind key="A-F7">
  <action name="Execute">
    <command>sh -c "LC_ALL=C obconf"</command>
  </action>
</keybind>
```

Another example will launch application preserving all stdout and stderr output to file:

```
<keybind key="A-f">
  <action name="Execute">
    <command>sh -c sh -c "exec gimp &gt;/tmp/gimp.out 2&gt;&amp;1"</command>
  </action>
</keybind>
```

Switch desktops using the mouse

It is possible to switch desktop by moving the mouse cursor to the edges of the screen. First install [xdotool](https://www.archlinux.org/packages/?name=xdotool) (<https://www.archlinux.org/packages/?name=xdotool>) and add the following two lines to your `~/.xinitrc`:

```
xdotool behave_screen_edge --delay 500 left set_desktop --relative -- -1 &
xdotool behave_screen_edge --delay 500 right set_desktop --relative -- +1 &
```

Set default applications / file associations

See the [Default applications](#) article.

Ad-hoc window transparency

Warning: This may not work where other actions are defined within the action group.

The program [transset-df](https://www.archlinux.org/packages/?name=transset-df) (<https://www.archlinux.org/packages/?name=transset-df>) can enable window transparency on-the-fly.

For example, using the following code in the `<mouse>` section of the `~/.config/openbox/rc.xml` file will enable control of application window transparency by hovering the mouse-pointer over the title bar and scrolling with the middle button:

```
<context name="Titlebar">
  ...
  <mousebind button="Up" action="Click">
    <action name="Execute" >
      <execute>transset-df -p .2 --inc </execute>
    </action>
  </mousebind>
  <mousebind button="Down" action="Click">
    <action name="Execute" >
      <execute>transset-df -p .2 --dec </execute>
    </action>
  </mousebind>
  ...
</context>
```


Using obxprop for faster configuration

The [openbox](https://www.archlinux.org/packages/?name=openbox) (<https://www.archlinux.org/packages/?name=openbox>) package provides a `obxprop` binary that can parse relevant values for applications settings in `rc.xml`. Officially `obxprop | grep "^_OB_APP"` is recommended for this task. Start the process by running the command shown, then click a window to see its properties in the terminal.

Xprop values for applications

[xorg-xprop](https://www.archlinux.org/packages/?name=xorg-xprop) (<https://www.archlinux.org/packages/?name=xorg-xprop>) can be used to relay property values for selected applications. Where frequently using per-application settings, the following [Bash Alias](#) may be useful:

```
alias xp='xprop | grep "WM_WINDOW_ROLE\|WM_CLASS" && echo "WM_CLASS(STRING) = \"NAME\", \"CLASS\""'
```

To use Xorg-XProp, run using the alias given `xp`, and click on the active program desired to define with per-application settings. The results displayed will only be the information that Openbox itself requires, namely the `WM_WINDOW_ROLE` and `WM_CLASS` (name and class) values:

```
WM_WINDOW_ROLE(STRING) = "roster"  
WM_CLASS(STRING) = "gajim.py", "Gajim.py"  
WM_CLASS(STRING) = "NAME", "CLASS"
```

Switching between keyboard layouts

See the article section [switching between keyboard layouts](#) for instructions.

Set grid layout for virtual desktops

Install [obsetLayout](https://aur.archlinux.org/packages/obsetLayout/) (<https://aur.archlinux.org/packages/obsetLayout/>)^{AUR}. To set a 2x2 grid for example:

```
obsetLayout 0 2 2 0
```

Run it without arguments to know what the arguments mean.

Enable Hot Corners

[lead](https://github.com/mlde/lead) (<https://github.com/mlde/lead>) provides hot corners for openbox and other lightweight window managers. Start the application with a entry in the autostart-file:

```
mlde.lead &
```

Commands can be edited in the configuration file `~/.config/mlde/lead.conf` :

```
[eDP1]
bottom=
bottomLeft=chromium
bottomRight=thunar
left=
right=
top=
topLeft=mlde.californium toggle
topRight=skippy-xd
```

Window snapping

Many desktop environments and window managers support *window snapping* (e.g. Windows 7 Aero snap), whereby they will automatically snap into place when moved to the edge of the screen. This effect can also be simulated in Openbox through the use of [keybinds](#) on focused windows.

As illustrated in the example below, percentages must be used to determine window sizes (see [openbox.org](http://openbox.org/wiki/Help:Actions) (<http://openbox.org/wiki/Help:Actions>) for further information). In this instance, The `super` key is used in conjunction with the `navigation` keys:

```
<keybind key="W-Left">
  <action name="UnmaximizeFull"/>
  <action name="MaximizeVert"/>
  <action name="MoveResizeTo">
    <width>50%</width>
  </action>
  <action name="MoveToEdge"><direction>west</direction></action>
</keybind>
<keybind key="W-Right">
  <action name="UnmaximizeFull"/>
  <action name="MaximizeVert"/>
  <action name="MoveResizeTo">
    <width>50%</width>
  </action>
  <action name="MoveToEdge"><direction>east</direction></action>
</keybind>
```

However, it should be noted that once a window has been 'snapped' to an edge, it will remain vertically maximised unless subsequently maximised and then restored. The solution is to implement additional keybinds - in this instance using the `down` and `up` keys - to do so. This will also make pulling 'snapped' windows from screen edges faster as well:

```
<keybind key="W-Down">
  <action name="Unmaximize"/>
</keybind>
<keybind key="W-Up">
  <action name="Maximize"/>
</keybind>
```

This [Ubuntu forum thread](http://ubuntuforums.org/showthread.php?t=1796793) (<http://ubuntuforums.org/showthread.php?t=1796793>) provides more

information. Applications such as [opensnap](https://aur.archlinux.org/packages/opensnap/) (<https://aur.archlinux.org/packages/opensnap/>)^{AUR} are also available to automatically simulate window snapping behaviour without the use of keybinds. Another option is to use [bunsen-utilities-git](https://aur.archlinux.org/packages/bunsen-utilities-git/) (<https://aur.archlinux.org/packages/bunsen-utilities-git/>)^{AUR} which provides `bl-aerosnap --left` and `bl-aerosnap --right` commands which will snap active window on left or right edge respectively if it's not snapped and restore it to original size and position otherwise. Just bind these commands to the key combination of your choosing.

Smooth display manager transition

Note: This has been confirmed to work with [LightDM](#).

Users of display managers might experience a flickering during the transition between the display manager and the Openbox desktop. The flickering comes from Openbox setting the root window's color during startup. Therefore there's a brief moment when the display flashes in a grey color, between the display manager's background and the desktop's wallpaper.

Setting the root window's background color can be disabled by editing the Openbox startup script found in `/usr/lib/openbox/openbox-autostart`. Simply comment out (or delete) the block starting with `# Set a background color`.

Note: Users who don't specifically set their wallpaper will "inherit" the display manager's background automatically if they disable the root window color adjustment.

Window Decorations

To remove window decorations for all or particular applications, use the `<decor>` option in the `<applications>` section of `rc.xml` (user: `~/.config/openbox/` or system: `/etc/xdg/openbox/`).

Example for Firefox, including variants like Firefox-Beta and Firefox-Nightly:

```
<application class="Firefox*">
  <decor>no</decor>
</application>
```

One could also disable decorations for all applications (using class `"*"`), then enable them (using **yes**) for individual ones. To apply the changes, restart your desktop session, and thus Openbox. Reference: [Openbox FAQ](http://openbox.org/wiki/Help:FAQ#How_do_I_remove_the_decorations_from_all_my_windows.3F) (http://openbox.org/wiki/Help:FAQ#How_do_I_remove_the_decorations_from_all_my_windows.3F)

Troubleshooting

Firefox

Mozilla based browsers may ignore application rules (e.g. `<desktop>`) unless `class="Firefox"` is used. See [#Xprop values for applications](#).

Missing themes

If for any reason the newly extracted theme cannot be selected, open the theme directory to first ensure that it is compatible with Openbox - there should be an `openbox-3` directory and a `themerc` file within it. An `.obt` (**O**pen**B**ox **T**heme) file may also be present in some instances, which can then be manually loaded in [obconf](https://www.archlinux.org/packages/?name=openbox) (<https://www.archlinux.org/packages/?name=openbox>).

A theme may also be not accessible due to wrong permissions. See [File permissions and attributes](#) for more.

Stop continuous desktop switching

By default Openbox switches from the last desktop back to the first desktop on mouse wheel scroll. Use `<wrap>no</wrap>` in the `mousebind` section to disable this behaviour.

```
<context name="Desktop">
  <mousebind button="Up" action="Click">
    <action name="GoToDesktop">
      <to>previous</to>
      <wrap>no</wrap>
    </action>
  </mousebind>
  <mousebind button="Down" action="Click">
    <action name="GoToDesktop">
      <to>next</to>
      <wrap>no</wrap>
    </action>
  </mousebind>
</context>
```

Windows load behind the active window

Some application windows (such as Firefox windows) may load behind the currently active window, causing you to need to switch to the window you just created to focus it. To fix this behavior add this to your `~/.config/openbox/rc.xml` file, inbetween the `<openbox_config>` and `</openbox_config>` tags:

```
<applications>
  <application class="*">
    <focus>yes</focus>
  </application>
</applications>
```

See also

- **[Openbox Website \(http://openbox.org/\)](http://openbox.org/)** - Official website
 - **[Box-Look.org \(http://www.box-look.org/\)](http://www.box-look.org/)** - A good resource for themes and related artwork
 - **[Openbox Hacks and Configs Thread \(https://bbs.archlinux.org/viewtopic.php?id=93126\)](https://bbs.archlinux.org/viewtopic.php?id=93126)** @ Arch Linux Forums
 - **[Openbox Screenshots Thread \(https://bbs.archlinux.org/viewtopic.php?id=45692\)](https://bbs.archlinux.org/viewtopic.php?id=45692)** @ Arch Linux Forums
 - **[An Openbox guide \(http://urukrama.wordpress.com/openbox-guide/\)](http://urukrama.wordpress.com/openbox-guide/)**
-

Retrieved from "<https://wiki.archlinux.org/index.php?title=Openbox&oldid=576185>"

This page was last edited on 23 June 2019, at 05:27.

Content is available under [GNU Free Documentation License 1.3 or later](#) unless otherwise noted.