

NAME

evilwm—minimalist window manager for X11

SYNOPSIS

evilwm [*OPTION*]...

DESCRIPTION

evilwm is a minimalist window manager for the X Window System. It features plenty of reconfigurable mouse and keyboard controls while providing a clean display, uncluttered by less useful window furniture (like title bars).

OPTIONS

- display** *display*
specifies the X display to run on. Usually this can be inferred from the **DISPLAY** environment variable.
- term** *termprog*
specifies an alternative program to run when spawning a new terminal (defaults to xterm, or x-terminal-emulator in Debian). Separate arguments with whitespace, and escape needed whitespace with a backslash. Remember that special characters will also need to be protected from the shell.
- fn** *fontname*
specify a font to use when resizing or displaying window titles.
- fg** *colour*
- fc** *colour*
- bg** *colour*
frame colour of currently active, fixed active, and inactive windows respectively. Either specify an X11 colour name like **goldenrod**, or a hex triplet like **#DAA520**.
- bw** *borderwidth*
width of window borders in pixels.
- snap** *distance*
enable snap-to-border support. *distance* is the proximity in pixels to snap to.
- wholescreen**
ignore monitor geometry and use the whole screen dimensions. This is the old behaviour from before multi-monitor support was implemented, and may still be useful, eg when one large monitor is driven from multiple outputs.
- numvdesks** *value*
number of virtual desktops to provide. Defaults to 8. Any extras will only be accessible by pagers or using Control+Alt+(Left/Right).
- nosoliddrag**
draw a window outline while moving or resizing.
- mask1** *modifier*[+*modifier*]...
- mask2** *modifier*[+*modifier*]...
- altmask** *modifier*[+*modifier*]...
override the default keyboard modifiers used to grab keys for window manager functionality.
mask1 is used for most keyboard controls (default: control+alt), and **mask2** is used for mouse button controls and cycling windows (default: alt). **altmask** is used to modify the behaviour of certain controls (default: shift). Modifiers may be separated with + signs. Valid modifiers are 'shift', 'control', 'alt', 'mod1'...'mod5'.
- bind** *key*[+*modifier*]...=[*function*[,*flag*]...]
bind a key pressed with specified modifiers to a window manager function. *key* is an X11 keysym name, *modifiers* are as above, but may also include 'mask1', 'mask2' and 'altmask' to refer to the globally-configured combinations. See FUNCTIONS for a list of available functions and the flags they recognise. If *function* is empty, a bind is removed.
- bind** *button*=[*function*[,*flag*]...]
bind a mouse button to a window manager function. While modifiers can be specified, they will be ignored; the button on its own will trigger if pressed within a window's frame, or with 'mask2' held anywhere within a window. Function and flags is as with key binds above. Valid buttons are 'button1'...'button5'.

- app** *name/class*
match an application by instance name and class (for help in finding these, use the *xprop* tool to extract the *WM_CLASS* property).
Subsequent **--geometry**, **--dock**, **--vdesk** and **--fixed** options will apply to this match.
- g, --geometry** *geometry*
apply a geometry (using a standard X geometry string) to applications matching the last **--app**.
- dock**
specify that application should be considered to be a dock, even if it lacks the appropriate property.
- v, --vdesk** *vdesk*
specify a default virtual desktop for applications matching the last **--app**. Note that virtual desktops are numbered from zero.
- f, --fixed**
specify that application is to start with a fixed client window.
- h, --help**
show help
- v, --version**
show program version

evilwm will also read options, one per line, from a file called *.evilwmrc* in the user's home directory. Options listed in a configuration file should omit the leading dash(es). Options specified on the command line override those found in the configuration file.

USAGE

In **evilwm**, the focus follows the mouse pointer, and focus is not lost if you stray onto the root window. The current window border is shaded gold (unless it is fixed, in which case blue), with other windows left as a dark grey.

You can use the mouse to manipulate windows either by click/dragging the single-pixel border (easier when they align with a screen edge), or by holding down Alt and doing so anywhere in the window. The controls are:

- Button 1
Move window.
- Button 2
Resize window.
- Button 3
Lower window.

Most keyboard controls are used by holding down Control and Alt, then pressing a key. Available functions are:

- Return Spawn new terminal.
- Escape Delete current window. Hold Shift as well to force kill a client.
- Insert Lower current window.
- H, J, K, L
Move window left, down, up or right (16 pixels). Holding Shift resizes the window instead.
- Y, U, B, N
Move window to the top-left, top-right, bottom-left or bottom-right of the current monitor.
- I Show information about current window.
- Equals Maximise current window vertically on current monitor (toggle). Holding Shift toggles horizontal maximization.

- X Maximise current window to current monitor (toggle).
- D Toggle visible state of docks, eg pagers and launch bars.

If compiled with virtual desktop support, these functions are also available:

- F Fix or unfix current window. Fixed windows remain visible when you switch virtual desktop.
- 1—8 Switch to specific virtual desktop (internally, desktops are numbered from zero, so this actually switches to desktops 0—7; this only becomes important if you use application matching).
- Left Switch to next lower numbered virtual desktop.
- Right Switch to next higher numbered virtual desktop.
- A Switch to the previously selected virtual desktop.

In addition to the above, Alt+Tab can be used to cycle through windows.

To make **evilwm** reread its config, send a HUP signal to the process. To make it quit, kill it, ie send a TERM signal.

FUNCTIONS

The keyboard and mouse button controls can be configured with the **--bind** option to a number of built-in functions. Typically, these functions respond to an additional set of flags that modify their behaviour.

delete

Delete a window. This is the co-operative way of closing applications, as it sends the client a signal indicating that they should shut down.

dock When called with the **toggle** flag, toggles visibility of any window claiming to be a dock.

fix With the **toggle** flag, toggle whether a window is fixed (visible on all virtual desktops) or not.

info Shows extra information about the current window for as long as the key is held.

kill Kill a window. A more forceful way of closing an application if it is not responding to delete requests.

lower Lower the current window.

move When bound to a button, moves a window with the mouse.

When bound to a key, if the **relative** flag is specified, moves a window in the direction indicated by other flags: **up**, **down**, **left** or **right**. Without the **relative** flag, moves a window in the direction specified by other flag to the edge of the monitor.

next Cycle to the next window.

raise

Raises the current window.

resize

When bound to a button, resizes a window with the mouse.

When bound to a key, if the **relative** flag is specified, modifies the width or height of the window as indicated by other flags: **up** (reduce height), **down** (increase height), **left** (reduce width) or **right** (increase width). If instead the **toggle** flag is specified, maximises along axes specified by other flags: **horizontal**, **vertical** or both.

spawn Start a terminal.

vdesk With the **toggle** flag specified, switch to the previously visible vdesk. With the **relative** flag set, either increase vdesk number (with **up** flag) or decrease it (with **down** flag).

If neither flag is specified, a numerical argument indicates which vdesk to switch to.

DEFAULT BINDS

These are the default lists of modifiers, button and keyboard binds. The built-in binds use the globally-configurable modifier combinations 'mask1', 'mask2' and 'altmask', making a sweeping change to a different

modifier combination easy.

Note that 'mod1' typically refers to the Alt key.

Modifiers

```
mask1 control+mod1
mask2 mod1
altmask shift
```

Button binds

```
bind button1=move
bind button2=resize
bind button3=lower
```

Keyboard binds

```
bind mask1+Return=spawn
bind mask1+Escape=delete
bind mask1+altmask+Escape=kill
bind mask1+Insert=lower
bind mask1+KP_Insert=lower
bind mask1+i=info
bind mask2+Tab=next
bind mask1+h=move,relative+left
bind mask1+j=move,relative+down
bind mask1+k=move,relative+up
bind mask1+l=move,relative+right
bind mask1+y=move,top+left
bind mask1+u=move,top+right
bind mask1+b=move,bottom+left
bind mask1+n=move,bottom+right
bind mask1+altmask+h=resize,relative+left
bind mask1+altmask+j=resize,relative+down
bind mask1+altmask+k=resize,relative+up
bind mask1+altmask+l=resize,relative+right
bind mask1+equal=resize,toggle+v
bind mask1+altmask+equal=resize,toggle+h
bind mask1+x=resize,toggle+v+h
bind mask1+d=dock,toggle
bind mask1+f=fix,toggle
bind mask1+1=vdesk,0
bind mask1+2=vdesk,1
bind mask1+3=vdesk,2
bind mask1+4=vdesk,3
bind mask1+5=vdesk,4
bind mask1+6=vdesk,5
bind mask1+7=vdesk,6
bind mask1+8=vdesk,7
bind mask1+Left=vdesk,relative+down
bind mask1+Right=vdesk,relative+up
bind mask1+a=vdesk,toggle
```

FILES

\$HOME/.evilwmrc

LICENCE

Copyright (C) 1999-2022 Ciaran Anscorb <evilwm@6809.org.uk>

This is free software. You can do what you want to it, but if it breaks something, you get to pay for the counselling. The code was originally based on aewm, so this is distributed under the same terms, which

follow.

AEWM LICENCE

Copyright (c) 1998-2000 Decklin Foster.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES OF ANY KIND. IN NO EVENT SHALL THE AUTHOR BE HELD LIABLE FOR ANY DAMAGES CONNECTED WITH THE USE OF THIS PROGRAM.

You are granted permission to copy, publish, distribute, and/or sell copies of this program and any modified versions or derived works, provided that this copyright and notice are not removed or altered.

Portions of the code were based on 9wm, which contains this license:

9wm is free software, and is Copyright (c) 1994 by David Hogan. Permission is granted to all sentient beings to use this software, to make copies of it, and to distribute those copies, provided that:

- (1) the copyright and licence notices are left intact
 - (2) the recipients are aware that it is free software
 - (3) any unapproved changes in functionality are either
 - (i) only distributed as patches
 - or (ii) distributed as a new program which is not called 9wm and whose documentation gives credit where it is due
 - (4) the author is not held responsible for any defects or shortcomings in the software, or damages caused by it.
- There is no warranty for this software. Have a nice day.

SEE ALSO

xterm (1), **xprop** (1)