NAME
    dwm – dynamic window manager

SYNOPSIS
    dwm [**−v**]

DESCRIPTION
    dwm is a dynamic window manager for X. It manages windows in tiled, monocle and floating layouts. Either layout can be applied dynamically, optimising the environment for the application in use and the task performed.

    In tiled layouts windows are managed in a master and stacking area. The master area on the left contains one window by default, and the stacking area on the right contains all other windows. The number of master area windows can be adjusted from zero to an arbitrary number. In monocle layout all windows are maximised to the screen size. In floating layout windows can be resized and moved freely. Dialog windows are always managed floating, regardless of the layout applied.

    Windows are grouped by tags. Each window can be tagged with one or multiple tags. Selecting certain tags displays all windows with these tags.

    Each screen contains a small status bar which displays all available tags, the layout, the title of the focused window, and the text read from the root window name property, if the screen is focused. A floating window is indicated with an empty square and a maximised floating window is indicated with a filled square before the windows title. The selected tags are indicated with a different color. The tags of the focused window are indicated with a filled square in the top left corner. The tags which are applied to one or more windows are indicated with an empty square in the top left corner.

    dwm draws a small border around windows to indicate the focus state.

OPTIONS
    −v   prints version information to standard output, then exits.

USAGE
  Status bar
    X root window name
            is read and displayed in the status text area. It can be set with the **xsetroot**(1) command.

    Button1
            click on a tag label to display all windows with that tag, click on the layout label toggles between tiled and floating layout.

    Button3
            click on a tag label adds/removes all windows with that tag to/from the view.

    Mod1−Button1
            click on a tag label applies that tag to the focused window.

    Mod1−Button3
            click on a tag label adds/removes that tag to/from the focused window.

  Keyboard commands
    Mod1−Shift−Return
            Start **st(1).**

    Mod1−p
            Spawn **dmenu(1)** for launching other programs.

    Mod1−,
            Focus previous screen, if any.

    Mod1−.
            Focus next screen, if any.

Mod1−Shift−,
>   Send focused window to previous screen, if any.

Mod1−Shift−.
>   Send focused window to next screen, if any.

Mod1−b
>   Toggles bar on and off.

Mod1−t
>   Sets tiled layout.

Mod1−f
>   Sets floating layout.

Mod1−m
>   Sets monocle layout.

Mod1−space
>   Toggles between current and previous layout.

Mod1−j
>   Focus next window.

Mod1−k
>   Focus previous window.

Mod1−i
>   Increase number of windows in master area.

Mod1−d
>   Decrease number of windows in master area.

Mod1−l
>   Increase master area size.

Mod1−h
>   Decrease master area size.

Mod1−Return
>   Zooms/cycles focused window to/from master area (tiled layouts only).

Mod1−Shift−c
>   Close focused window.

Mod1−Shift−space
>   Toggle focused window between tiled and floating state.

Mod1−Tab
>   Toggles to the previously selected tags.

Mod1−Shift−[1..n]
>   Apply nth tag to focused window.

Mod1−Shift−0
>   Apply all tags to focused window.

Mod1−Control−Shift−[1..n]
>   Add/remove nth tag to/from focused window.

Mod1−[1..n]
>   View all windows with nth tag.

Mod1−0
>   View all windows with any tag.

Mod1−Control−[1..n]
> Add/remove all windows with nth tag to/from the view.

Mod1−Shift−q
> Quit dwm.

Mouse commands

Mod1−Button1
> Move focused window while dragging. Tiled windows will be toggled to the floating state.

Mod1−Button2
> Toggles focused window between floating and tiled state.

Mod1−Button3
> Resize focused window while dragging. Tiled windows will be toggled to the floating state.

CUSTOMIZATION
dwm is customized by creating a custom config.h and (re)compiling the source code. This keeps it fast, secure and simple.

SEE ALSO
dmenu(1), **st**(1)

ISSUES
Java applications which use the XToolkit/XAWT backend may draw grey windows only. The XToolkit/XAWT backend breaks ICCCM-compliance in recent JDK 1.5 and early JDK 1.6 versions, because it assumes a reparenting window manager. Possible workarounds are using JDK 1.4 (which doesn't contain the XToolkit/XAWT backend) or setting the environment variable **AWT_TOOLKIT=MToolkit** (to use the older Motif backend instead) or running **xprop -root -f _NET_WM_NAME 32a -set _NET_WM_NAME LG3D** or **wmname LG3D** (to pretend that a non-reparenting window manager is running that the XToolkit/XAWT backend can recognize) or when using OpenJDK setting the environment variable **_JAVA_AWT_WM_NONREPARENTING=1**.

BUGS
Send all bug reports with a patch to hackers@suckless.org.