

Install and configure Picom

Introduction

Picom was previously known as Compton and it is a window decorator for Linux systems. I was looking into how to setup Picom to rice my i3 desktop, but noticed there wasn't a lot of information on how to do this.

Configuration

So I ended up with the following configuration

```
#####  
#  
# Backend  
#  
#####  
  
# Backend to use: "xrender" or "glx".  
# GLX backend is typically much faster but depends on a sane driver.  
backend = "glx";  
  
#####  
#  
# GLX backend  
#  
#####  
  
glx-no-stencil = true;  
  
# GLX backend: Copy unmodified regions from front buffer instead of redrawing them all.  
# My tests with nvidia-drivers show a 10% decrease in performance when the whole screen is  
modified,  
# but a 20% increase when only 1/4 is.  
# My tests on nouveau show terrible slowdown.  
# Useful with --glx-swap-method, as well.  
#glx-copy-from-front = false;  
  
# GLX backend: Use MESA_copy_sub_buffer to do partial screen update.
```

```
# My tests on nouveau shows a 200% performance boost when only 1/4 of the screen is updated.
# May break VSync and is not available on some drivers.
# Overrides --glx-copy-from-front.
# glx-use-copysubbuffermesa = true;

# GLX backend: Avoid rebinding pixmap on window damage.
# Probably could improve performance on rapid window content changes, but is known to break
things on some drivers (LLVMpipe).
# Recommended if it works.
# glx-no-rebind-pixmap = true;

# GLX backend: GLX buffer swap method we assume.
# Could be undefined (0), copy (1), exchange (2), 3-6, or buffer-age (-1).
# undefined is the slowest and the safest, and the default value.
# copy is fastest, but may fail on some drivers,
# 2-6 are gradually slower but safer (6 is still faster than 0).
# Usually, double buffer means 2, triple buffer means 3.
# buffer-age means auto-detect using GLX_EXT_buffer_age, supported by some drivers.
# Useless with --glx-use-copysubbuffermesa.
# Partially breaks --resize-damage.
# Defaults to undefined.
# (Deprecated) glx-swap-method = "undefined";
use-damage = false;

#####
#
# Shadows
#
#####

# Enabled client-side shadows on windows.
shadow = true;
# Don't draw shadows on DND windows.
# (Deprecated) no-dnd-shadow = true;
# Avoid drawing shadows on dock/panel windows.
# (Deprecated) no-dock-shadow = true;
# option no-dock-shadow -> wintype option shadow of dock
# option no-dnd-shadow -> wintype option shadow of dnd
# menu-opacity -> wintype option opacity of popup_menu and dropdown_menu
# vsync option -> boolean
```

```
# opengl-swc -> interpreted as true, but will be Deprecated
# glx-swap-method -> deprecated -> use-damage = true
# The blur radius for shadows. (default 12)
shadow-radius = 10;
# The left offset for shadows. (default -15)
shadow-offset-x = 0;
# The top offset for shadows. (default -15)
shadow-offset-y = 0;
# The translucency for shadows. (default .75)
shadow-opacity = 0.25;

# Set if you want different colour shadows
# shadow-red = 0.0;
# shadow-green = 0.0;
# shadow-blue = 0.0;

# The shadow exclude options are helpful if you have shadows enabled. Due to the way compton
draws its shadows, certain applications will have visual glitches
# (most applications are fine, only apps that do weird things with xshapes or argb are
affected).
# This list includes all the affected apps I found in my testing. The "! name~='" part
excludes shadows on any "Unknown" windows, this prevents a visual glitch with the XFWM alt tab
switcher.
#shadow-exclude = [
[]#! name~='"',
[]#"name = 'Notification'",
[]#"name = 'Plank'",
[]#"name = 'Docky'",
[]#"name = 'Kupfer'",
[]#"name = 'xfce4-notifyd'",
[]#"name *= 'VLC'",
[]#"name *= 'compton'",
[]#"name *= 'Chromium'",
[]#"name *= 'Chrome'",
[]#"class_g = 'Conky'",
[]#"class_g = 'Kupfer'",
[]#"class_g = 'Synapse'",
[]#"class_g ?= 'Notify-osd'",
[]#"class_g ?= 'Cairo-dock'",
[]#"class_g ?= 'Xfce4-notifyd'",
[]#"class_g ?= 'Xfce4-power-manager'",
```

```

[]#"_GTK_FRAME_EXTENTS@:c"
#];
shadow-exclude = [
[]"class_g = 'i3-frame'",
[]"class_g = 'Polybar'",
[]"_NET_WM_STATE@:32a *= '_NET_WM_STATE_HIDDEN'",
[]"class_g ?= 'rofi'",
[]"_GTK_FRAME_EXTENTS@:c"
[]
];

# Avoid drawing shadow on all shaped windows (see also: --detect-rounded-corners)
shadow-ignore-shaped = false;

#####
#
# Opacity
#
#####

# (Deprecated) menu-opacity = 1;
inactive-opacity = 0.75;
active-opacity = 0.85;
frame-opacity = 0.90;
inactive-opacity-override = false;
# (Deprecated) alpha-step = 0.06;

# Fix tabbed windows
opacity-rule = [
[]"100:class_g = 'mpv'",
[]"0:_NET_WM_STATE@:32a *= '_NET_WM_STATE_HIDDEN'",
[]"90:class_g ?= 'Rofi'"
];

# Dim inactive windows. (0.0 - 1.0)
# inactive-dim = 0.2;
# Do not let dimness adjust based on window opacity.
# inactive-dim-fixed = true;

# Blur background of transparent windows. Bad performance with X Render backend. GLX backend
is preferred.

```

```

blur-background = true;
# Blur background of opaque windows with transparent frames as well.
blur-background-frame = true;
# Do not let blur radius adjust based on window opacity.
blur-background-fixed = false;
#blur-background-exclude = [
#[]"window_type = 'dock'",
#[]"window_type = 'desktop'"
#];

#####
#
# Fading
#
#####

# Fade windows during opacity changes.
fading = true;
# The time between steps in a fade in milliseconds. (default 10).
fade-delta = 10;
# Opacity change between steps while fading in. (default 0.028).
fade-in-step = 0.06;
# Opacity change between steps while fading out. (default 0.03).
fade-out-step = 0.06;
# Fade windows in/out when opening/closing
#no-fading-openclose = true;

# Specify a list of conditions of windows that should not be faded.
fade-exclude = [ ];

#####
#
# Other
#
#####

# Try to detect WM windows and mark them as active.
mark-wmwin-focused = true;
# Mark all non-WM but override-redirect windows active (e.g. menus).
mark-ovredir-focused = true;
# Use EWMH _NET_WM_ACTIVE_WINDOW to determine which window is focused instead of using

```

```
FocusIn/Out events.
# Usually more reliable but depends on a EWMH-compliant WM.
use-ewmh-active-win = true;
# Detect rounded corners and treat them as rectangular when --shadow-ignore-shaped is on.
detect-rounded-corners = true;

# Detect _NET_WM_OPACITY on client windows, useful for window managers not passing
_NET_WM_OPACITY of client windows to frame windows.
# This prevents opacity being ignored for some apps.
# For example without this enabled my xfce4-notifyd is 100% opacity no matter what.
detect-client-opacity = true;

# Specify refresh rate of the screen.
# If not specified or 0, compton will try detecting this with X RandR extension.
refresh-rate = 0;

# Set VSync method. VSync methods currently available:
# none: No VSync
# drm: VSync with DRM_IOCTL_WAIT_VBLANK. May only work on some drivers.
# opengl: Try to VSync with SGI_video_sync OpenGL extension. Only work on some drivers.
# opengl-oml: Try to VSync with OML_sync_control OpenGL extension. Only work on some drivers.
# opengl-swc: Try to VSync with SGI_swap_control OpenGL extension. Only work on some drivers.
Works only with GLX backend. Known to be most effective on many drivers. Does not actually
control paint timing, only buffer swap is affected, so it doesn't have the effect of --sw-opti
unlike other methods. Experimental.
# opengl-mswc: Try to VSync with MESA_swap_control OpenGL extension. Basically the same as
opengl-swc above, except the extension we use.
# (Note some VSync methods may not be enabled at compile time.)
vsync = true;

# Enable DBE painting mode, intended to use with VSync to (hopefully) eliminate tearing.
# Reported to have no effect, though.
dbe = false;
# (Removed) Painting on X Composite overlay window. Recommended.
# paint-on-overlay = true;

# Limit compton to repaint at most once every 1 / refresh_rate second to boost performance.
# This should not be used with --vsync drm/opengl/opengl-oml as they essentially does --sw-
opti's job already,
# unless you wish to specify a lower refresh rate than the actual value.
sw-opti = true;
```

```

# Unredirect all windows if a full-screen opaque window is detected, to maximize performance
for full-screen windows, like games.
# Known to cause flickering when redirecting/unredirecting windows.
# paint-on-overlay may make the flickering less obvious.
unredir-if-possible = false;

# Specify a list of conditions of windows that should always be considered focused.
focus-exclude = [ ];

# Use WM_TRANSIENT_FOR to group windows, and consider windows in the same group focused at the
same time.
detect-transient = true;
# Use WM_CLIENT_LEADER to group windows, and consider windows in the same group focused at the
same time.
# WM_TRANSIENT_FOR has higher priority if --detect-transient is enabled, too.
detect-client-leader = true;

#####
#
# Window type settings
#
#####

wintypes:
{
[]tooltip =
[]{
[]# fade: Fade the particular type of windows.
[]fade = true;
[]# shadow: Give those windows shadow
[]shadow = true;
[]# opacity: Default opacity for the type of windows.
[]opacity = 0.85;
[]# focus: Whether to always consider windows of this type focused.
[]focus = false;
[]};
[]menu =
[]{
[]# fade: Fade the particular type of windows.
[]fade = true;

```

```
    # shadow: Give those windows shadow
    shadow = true;
    # opacity: Default opacity for the type of windows.
    # focus: Whether to always consider windows of this type focused.
    focus = true;
};
dialog =
{
    # fade: Fade the particular type of windows.
    fade = true;
    # shadow: Give those windows shadow
    shadow = true;
    # opacity: Default opacity for the type of windows.
    opacity = 1;
    # focus: Whether to always consider windows of this type focused.
    focus = true;
    # Blur background
    blur-background = true;
};
dropdown_menu =
{
    # fade: Fade the particular type of windows.
    fade = true;
    # shadow: Give those windows shadow
    shadow = true;
    # opacity: Default opacity for the type of windows.
    # focus: Whether to always consider windows of this type focused.
    focus = true;
};
popup_menu =
{
    # fade: Fade the particular type of windows.
    fade = true;
    # shadow: Give those windows shadow
    shadow = true;
    # opacity: Default opacity for the type of windows.
    opacity = 0.85;
    # focus: Whether to always consider windows of this type focused.
    focus = true;
};
Notification =
```

```
{
  # fade: Fade the particular type of windows.
  fade = true;
  # shadow: Give those windows shadow
  shadow = true;
  # opacity: Default opacity for the type of windows.
  opacity = 0.85;
  # focus: Whether to always consider windows of this type focused.
  focus = false;
};

normal =
{
  # fade: Fade the particular type of windows.
  fade = true;
  # shadow: Give those windows shadow
  shadow = false;
  # opacity: Default opacity for the type of windows.
  opacity = 1;
  # focus: Whether to always consider windows of this type focused.
  focus = false;
};
};
```

Revision #2

Created 2021-01-13 19:20:11 UTC by Pascal

Updated 2021-03-14 14:35:05 UTC by Pascal