

kompngr Information

Thomas Luebking



kompngr Information

Contents

1	Introduction	1
2	Xorg	2
2.1	Setup	2
3	Settings	3
3.1	Translucency	3
3.2	Shadows	4
3.3	Effects	5
4	Problems	6
5	Credits and Licenses	8
5.1	Authors	8

Abstract

kompngr is a composite rendering engine to be used by KWin.

This document is rather a general explanation of what it does, how it works and what the restrictions are than a documentation how to use it (as it works in the background automatically).

Chapter 1

Introduction

kompngr is based upon xcompngr, Keith Packards reference composing implementation for the use of the new X features.

A composite manager catches the output of the X windows and renders them into a single picture that will be displayed on the screen. The advantage of doing so is that you can do basically anything with the windows contents, including blending them together (translucency) or attaching shadows. You could also tint them, add a nasty advertise on any window, squeeze them, fract them, add scanlines, or anything else you can imagine, limited by your hardware.

Chapter 2

Xorg

You necessarily need *X.org 6.8 or later* to have kompngr work. *X.org 6.7 or older* or *XFree86* won't work.

2.1 Setup

You need to explicitly enable the Composite extension. Add a new Section to your X.org configuration file:

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

If supported by your GPU (the complex chip that powers a modern graphics card), activate the Render acceleration. This is best supported by NVIDIA™, and less completely by ATI cards.

```
Section "Device"
....
....
    Option      "RenderAccel" "true"
....
....
EndSection
```

kompngr should be available now. For more information, please see the [FAQ](#).

Chapter 3

Settings

This chapter describes which parameters you can tweak, how they will result visually and their performance impact.

3.1 Translucency

A translucent object is one that allows light to pass through it. In terms of windows on your desktop, that means that the contents of windows can be seen through the one on the top.

Translucency allows you to emphasize special windows, have a 3 dimensional view on your desktop, keep track of covered windows, and just looks cute. The price is, that blending things together costs some system capacity.

You can independently configure translucency for the following items:

Active windows It is suggested you deactivate translucency for the active window. The main reason is for performance, and secondly because to scan the content of a translucent window means your brain has to strip the irritating throughshining information, which is tiring.

Inactive windows If you set inactive windows translucent, active windows will appear emphasized and are easier to focus. However, if you choose a lower value, you may have trouble to find an inactive window. If you choose very low values (< 20%) you may not be able to distinguish windows in their stack order - so you may accidently click the OK button of a dialog when you just wanted to activate a window. Optimal values are between 60% and 80%.

Moving windows Though it's a nice effect to have heavily translucent (opacity < 20%) moving windows, there is a heavy price to pay in performance, especially if you do not deactivate shadows (see below). Just try out and

kompmgr Information

if you feel your system is too slow, keep moving windows opaque. This value also applies to windows in resize state.

Dock windows As docks like kicker are seldom if ever moved and usually of limited size, this is purely visual and won't detrimentally affect your system's performance.

Treat "keep above" windows as active ones If you set a window to keep above others, you usually want to keep focus on it, so it can make sense to give it the same emphasis.

Disable ARGB windows XRender supports windows with an alpha mask, i.e. translucent sections. Currently there are no or only very few applications that would make use of this feature, as it doesn't make any sense without using a composite manager. This may change in the future.

On the other side, most gtk 1.x applications (e.g. xmms) set such an alpha mask, resulting in almost unusable windows (as long as the sublying windows are not black), so you can disable the support for ARGB windows here to make use of gtk applications. There will hopefully be a patch for gtk in the near future to fix this.

3.2 Shadows

Why should anyone want his windows to throw shadows? Well, maybe just because it looks cute, or because it allows a better window separation, or... because it looks cute!

Use shadows As shadows cost some additional cpu/gpu power, they can be deactivated, while keeping a general alpha channel capability.

Active window size, Inactive window size, Dock window size You may select different shadow sizes for different window states/types. The values are not absolute, but will apply to your screen resolution.

In principle, you can set every value you want (if you edit the config file with a text editor, far beyond the offered '32'), but bigger shadows cost more cpu/gpu power.

Vertical offset, Horizontal offset By default, the window would throw a 'all around equal' shadow, implying a frontal light source. Users familiar with Mac® OS may like a vertical offset, Windows® users may prefer a slighter offset to the southeast. Experiment with the values until you are happy.

Shadow color Usually shadows as the absence of light appear in shades of grey (so the maximum shadow color is black, or no light at all). But hey, this is a virtual world, and if you'd like to have even pink shadows, why not?

Remove shadows on move Check this if you need more performance (especially when using translucent moving windows).

Remove shadows on resize It's a good idea to check this item whether using translucent resizing (moving) windows, or not. The windows shadow pixmap has to be permanently recreated on resizes what has a deep impact on the system performance.

3.3 Effects

Fade-in windows, Fade between opacity changes Instead of just popping up a new window, you might want to slowly fade it in. While this looks impressive, there is an equally impressive price to pay in performance speed, and it is difficult to interact meaningfully with windows that are in the process of fading.

Fade-in speed, Fade-out speed For the usability reasons mentioned under Fade-in windows, it is most practical to use a fast fade in speed and (for more eye candy) a slow fade out speed. This will provide you nice effects and a smooth feeling system as well as low latency before information appears.

Chapter 4

Problems

The whole Composite extension is quite new. It may cause several problems and even crash X, so *it is strongly suggested to not even enable the Composite extension in Xorg.conf on mission critical production systems*. If you however can risk a slightly more unstable system, it's certainly nice to have some eye candy around.

In this case you may notice some glitches. Here are some common problems - and workarounds:

1. *I have X.org 6.8.x, but kompMgr fails to start*

You need to explicitly enable the Composite extension, add a new Section to /etc/X11/XorgConfig:

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

2. *Ok, kompMgr works, but it's horribly slow*

If supported by your GPU (mainly NVIDIA™, and somewhat by ATI cards), first make sure you activated the Render acceleration

```
Section "Device"
....
....
Option      "RenderAccel" "true"
....
....
EndSection
```

If it's still slow, you can try to adjust your memory usage. Either decrease your screen depth (e.g. from 24 to 16) or lower your resolution (e.g. from 1280x1024 to 1024x768). Please notice that the currently limiting factor for the Composite extension seems to be the CPU cache size.

kompmgr Information

3. *After enabling the Composite extension, I cannot run any GLX applications anymore. I've got an NVIDIA™ card.*

To prevent some problems, NVIDIA™ deactivated the GLX support when Coposite is active. Reactivating is possible, but may cause problems on some kernel/driver/GPU combinations.

```
Section "Device"
....
....
Option "AllowGLXWithComposite" "true"
....
....
EndSection
```

4. *I wanted to play a game using SDL (but not GL), say scummvm, but when the compmgr is active, all I can see is a shadow!*

This is a problem with the PictType reported by SDL. Workaround:

```
Instead of calling scummvm directly, call
SDL_VIDEO_X11_VISUALID=0x24 scummvm This will tell SDL ↔
to
use a supported format and you can play as ever.
```

5. *Application XXX refuses to start after enabling the Composite extension. Application YYY breaks X after enabling the Composite extension. Application ZZZ looks weird after activating the composite manager.*

The Composite extension is still experimental. Workaround: Instead of calling **appName** directly, call **XLIB_SKIP_ARGB_VISUALS=1 appName** Applications that are known to cause problems:

- All gtk1 applications (e.g. gmpayer, xmms, gaim) - failing startup or look unusable weird or crash X
- KuickShow - displays only a black frame
- Qt™ Designer - crashes X
- Kopete - crashes X
- Kolf - crashes X

6. *I try to watch a video, but only see artefacts in the Video Window.*

You're using 'xv' as video backend. This is the overlay mode, where the video content is written directly into the video card memory, bypassing X. Therefore the window seems to be static (colored background) and is not updated by the damage extension. There will hopefully be a fix for this in the future. Currently I had the best results using Xine but displaying translucent videos isn't fast anyway.

Chapter 5

Credits and Licenses

5.1 Authors

Thomas Luebking baghira-style@gmx.net - Editor

This documentation is licensed under the terms of the [GNU Free Documentation License](#).

This program is licensed under the terms of the [GNU General Public License](#).