

Magic Layers Documentation

Trond Werner Hansen

Copyright © CopyrightÂ©1995 Trond Werner Hansen

COLLABORATORS

	TITLE : Magic Layers Documentation		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Trond Werner Hansen	December 2, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Magic Layers Documentation	1
1.1	Magic Layers	1
1.2	Introduction	1
1.3	Legal stuff	2
1.4	How to install Magic Layers	2
1.5	Using Magic Layers	3
1.6	History & Bugs	4
1.7	Credits	5

Chapter 1

Magic Layers Documentation

1.1 Magic Layers

Magic Layers 1.1 Copyright © 1995 Trond Werner Hansen

- Introduction
- Legal stuff
- How to install Magic Layers
- Using Magic Layers
- History & Bugs
- Credits

1.2 Introduction

A Look Back

With a 68000 and some chip ram, the earlier Amigas were not really suited for realtime movement of layers of graphics, and could not deliver the power needed for speedy refreshment. Sadly, some of the ways layers.library and intuition were made to work then, and still do today, make such realtime window movement slow, even on a Amiga 4000 040. This fact is easily proven with the program OpaqueMove.

Some people may remember a program called Turbo Layers, which appeared in june 1991, a time when having Kickstart 2.0 was a great thing. Magic Layers is not an update, nor based on this program. However, the two programs share some common ideas on how to get things moving. Turbo Layers had some major flaws, and due to its naughty behaviour, said thank you and good night when Kickstart 3.0 showed up. Magic Layers is not very nice either, and may fuck up in various ways.

Magic Layers

Magic Layers introduces realtime movement of windows on any screen with any depth, although screens with depth greater than 4 bitplanes make things move slowly. Keep in mind that it does only work with "native chipset screens", not with gfxcard buffered screens (See below).

In addition, Magic Layers gives you realtime sizing of windows, but its "contents" are not refreshed in realtime. The frame, and any backfill pattern are refreshed realtime. The rest is refreshed as normal afterwards.

Magic Layers, unlike OpaqueMove, does not really move or size the window in realtime. It's fake, and works as a visual representation of what you're about to do. In fact, it's much like intuition's own frame-representation. This fake visual representation has its costs; Magic Layers allocates a chipram-buffer twice the size of the screen which the window in question is residing on.

No graphics cards!

Magic Layers takes an entirely different approach than OpaqueMove. While OpaqueMove is meant to work independantly of graphical devices, Magic Layers fools intuition and graphics.library in believing lots of weird stuff. This weird stuff is at the moment not compatible with other graphical devices.

Trond Werner Hansen.

1.3 Legal stuff

Disclaimer

The program is provided "as is". No representations or warranties are made with respect to the accuracy, reliability, performance, or operation of this software, and all use is at your own risk.

Copyright

This program is freeware and can be freely distributed together with this archive, which must be left unaltered.

Magic Layers is copyright © 1995 Trond Werner Hansen. All rights reserved.

1.4 How to install Magic Layers

Installing Magic Layers

Installing Magic Layers is easy, just drag and copy it to your WBStartup drawer. If you want, you can start the program at once, without rebooting.

Note that Magic Layers requires V39 or newer, and only Magic Workbench icons are included.

1.5 Using Magic Layers

Using Magic Layers

Magic Layers allocates a chipmem buffer twice the size of the screen when you start to drag or size a window on a screen for the first time. This buffer is not freed before Magic Layers is made inactive, removed, or a window on another screen is affected. Then the buffer is freed and a new buffer is allocated for the new screen. The buffer is kept "active" in this way for speed reasons.

Refreshment of simpler refresh windows like workbench's own drawer-windows will not always be completed before they are exposed. The refreshment is in fact done in another buffer and that's why nothing more will appear before you drag the window over it or the movement is ended. Using the delay tooltype will give Magic Layers more time for initial refreshment.

Magic Layers is a commodity, which can be made (in)active with an exchange-like program or by using a defined hotkey. Only one Magic Layers program will run at the same time, and executing a new one quits the one already running and returns.

Magic Layers has the following options as tooltypes.

CX_TOGGLE

Defines the in(active) toggle hotkey.

DEFPUBSCREENONLY

Only the default public screen will be affected.

NOSIZING

Realtime sizing is disabled.

MAXDEPTH

Screens that are deeper will not be affected.

MAXWINWIDTH

Windows with greater width will not be affected.

MAXWINHEIGHT

Windows with greater height will not be affected.

DELAY

Wait (0,1,2.....) for refresh before proceeding with movement/sizing.

INPUTPRIORITY

The handler's priority. Do not change this unless you know what you're doing. Values below 50 will ofcourse don't do any good. (Default=51)

Tips

Blitting always look nicer with interleaved bitmaps, and promoting a screen without this feature can be done with various utilities. ↩

If you use a program like SmartWB, all (wb)icons will be refreshed before proceeding with movement/sizing. ↩

1.6 History & Bugs

Magic Layers 1.0, (25.04.95)

- First public release.

Magic Layers 1.1, (06.06.95)

- Fixed menu-conflict (locked layers) bug.
- Changed WBOonly tooltype to DefPubScreenOnly.
- Added MaxWinWidth/Height tooltypes.
- Now stops redrawing when beyond limits.
- Window with no-op backfillhook, is forced to use default backfillhook.
- When patches cannot be removed, a requester is now displayed.
- Now falls back to system sizing with BORDERLESS windows, which have own "custom" sizing-gadgets. With no border to size, it looked awful.
- Fixed bug when moving windows containing requesters (layers). Was unable to find a similiar solution when SIZING a window containing requesters. Now falls back to system sizing with such windows.
- SIZING a window's layer when ANOTHER window contains requesters is for some reason slower. This "bug" is perhaps related to the one I was unable to fix (over). An intuition requester inside a window is simply a layer and since the sizing in question is done on layers-level, I cannot see why such requesters should suddenly affect the sizing of another layer more than a normal window. Few programs use intuition requesters and the problem shouldn't bother

many...

1.7 Credits

Magic Layers was developed by:

Trond Werner Hansen, Norway

E-Mail: shoeless@mix.hsv.no

I have not been able to reply to all mail, but this update will probably answer most of the questions.

Thanks to Bernardo Innocenti & Stefan Zeiger for their suggestions.