

# Jo's alternative Desktop HOCUS POCUS

JO MOSKALEWSKI

**Only you can decide how your Linux desktop looks. With deskTOPia we regularly take you with us on a journey into the land of window managers and desktop environments, presenting the useful and the colourful, viewers and pretty toys. In the last issue, a program disappeared before the very eyes of a window manager. This time we are going to make the background of a terminal transparent – and at the same time save memory.**



Hastily put down by many a newbie as a cumbersome relic of past ages of computing, many are surely still becoming familiar with the advantages of a text-based command input (known as shell or terminal) – at least if the limits of the standard installation from the distributor are not simultaneously meant to mark the edge of one's own horizon. Lots of things can be used more quickly and effectively with this. And for some things, a shell is actually necessary – such as for putting together a new kernel. Even environments like KDE2 are not making any effort to do away (as many might presume) with the shell – on the contrary, a meaningful symbiosis is the objective.

## Mirror, mirror, on the wall...

Terminal programs for the graphical user interface are abundant in the Unix environment. The standard terminal is called xterm – and there is also a special (optically adapted) wterm for the user of Window Maker. The terminal of KDE calls itself

Console. All fulfil the same purpose with varying equipment and appearance. With respect to consumption of resources, clear differences can also be discerned. Only you can say which terminal suits you best. But one thing is for sure, a user of Window Maker no more has to implement a wterm than the KDE faction their Console. All terminals are fully usable on another desktop.

Admittedly, the console of KDE2 is a thing of beauty. But anyone who has only 64MB of working memory (which is just enough for KDE2) will not be happy to pay for these good looks with more than 8MB of memory capacity for the console alone. Users of the GNOME desktop are a bit better served in this instance: Here 3.4MB will go to the GNOME terminal. But even that is clearly too much for a user who wants to use several text-based programs in a graphical environment – or is working with four or more consoles at the same time (and that's something everyone does, sooner or later). On the opposite page is a little table in which the various terminal programs are compared in this respect:

Memory	Terminal	Desktop
8.4MB	console	KDE2
3.4MB	gnome-terminal	GNOME
2.7MB	Eterm	Enlightenment
1.8MB	xterm	-
1.3MB	wterm	Window Maker
1.1MB	aterm	AfterStep
1.0MB	rxvt	-

## Gingerbread house

The aterm terminal has a lot to offer, coupled with very little memory requirement. The latest version can be found on the coverdisc. Future releases can be fetched from <http://aterm.sourceforge.net/>. aterm arose from the code of wterm, which in turn took rxvt as a model. Nevertheless, the prime choice was aterm, which apart from all the usual features, offers the following:

- scaleable display of background graphics
- pseudo-transparency (can also be coloured)
- sensible scrollbars (also transparent if required)
- fading when there is loss of focus

This list is certainly not exhaustive, but hopefully it has been sufficient to whet your appetite. Installing aterm is really simple. It needs what every other graphics program also requires: A compiler (e.g. GCC) including the associated tool Make and support for XPM graphics (xpm4g-dev or libXpm) is required – equally, the header files for X (xdevl) should be installed. And all this comes with an ordinary distribution. After that the aterm packet can be converted for your own distribution and installation:

```
jo@planet ~-> tar -xvzf aterm-0.4.0.tar.gz
[ ... ]
jo@planet ~-> cd aterm-0.4.0
jo@planet ~/aterm-0.4.0> ./configure --enable2
--fading --enable-background-image
jo@planet ~/aterm-0.4.0> make
jo@planet ~/aterm-0.4.0> su -
Password:
root@planet ~-> cd /home/jo/aterm-0.4.0
root@planet /home/jo/aterm-0.4.0> make install
root@planet ~-> logout
```

I gave a few options with the Configure script here, to enable the display of background graphics, for instance. There are a large number of additional options, by means of which the terminal can be made leaner or even more voluminous – it's up to you. There is an overview in the file *README.configure* in the packet.

## Off to the realm of dreams

Obviously, our aterm is started with the command aterm – but first the program will surely gladden only the purists with a look at the amount of free working memory. Until we tell aterm how it is going to look in future, it puts on a face as drab as an ordinary xterm (even if it does have a sensible scrollbar). So let's start to configure our aterm.

## Font and window size

To be able to use this thing immediately it is definitely essential to design the outputs legibly. The font can be selected by calling up with the option *-fn fontname*. However, as with every other terminal, a font has to be selected with a specified character width. The simplest thing to do here is to experiment with *-fn 8x16* or *-fn 10x20*, but the font *lucidatypewriter* is also suitable (e.g.: *aterm -fn \*-lucidatypewriter-medium-r-\*-\*-\*-120-\*-\*-\*-iso8859-\**). The tool *xfontsel* reveals additional font names (as does *gfontsel*). But that's not all: Anyone who prefers to work with 80x30 characters instead of the usual 80x25 characters simply adds *-geometry 80x30* when starting the program off.

## Foreground and background

Fancy a dark blue background with pale grey text? No problem: An *aterm -bg "#000040" -fg lightgrey* and it's done. But what if, instead of dark blue you'd rather have a graphic? Graphics should preferably be in XPM format – if not, the tool

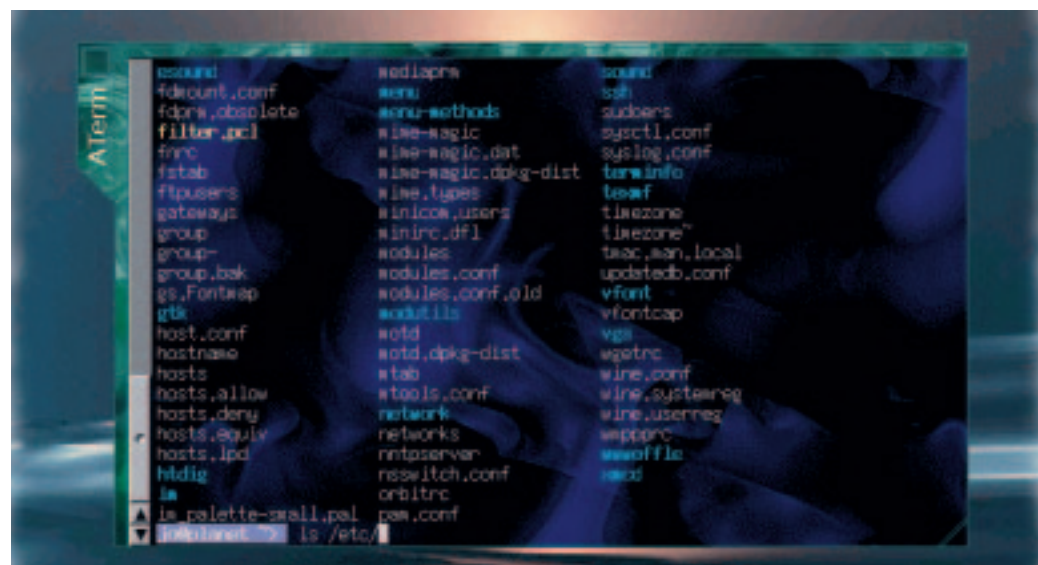


Figure 1: aterm with blue-coloured background pixmap.

*Convert* for example from the ImageMagick packet would recommend itself. After that the reward comes via *aterm -pixmap graphic.xpm*. By default the graphic now becomes tiled. Anyone wanting to adapt this graphic to the window size can do this using the option *-backgroundType scale*. Additional display options for graphics can be found in the instructions (*man aterm*).

Another popular option is to see the desktop background through the terminal. This is simple to achieve with *aterm -tr*. But there is one restriction on this function: Only pseudo-transparency is offered, with just the corresponding section of the desktop background used – any windows lying underneath this are not visible.

The drawback with all graphics here is the poorer legibility of the characters and figures in *aterm*. And here, too, the makers of *aterm* have not been idle. They have come up with the option *tint colourname* to ensure that the background shining through in the terminal can be coloured. Those who prefer their backgrounds to look a bit more primitive can darken the window content by 50% – the option *-sh 50* clears the way for this. And thus, high-contrast graphics as background for text outputs is no longer a knockout criterion.

## Fading on off-focus

What's so dramatic about this is not that simple, when all's said and done: Just as a window frame highlights an active window in colour, an *aterm* can (additionally) signal whether it is currently active – and thus listening to our keyboard. This function is activated with the call parameter *-fading number*. For the number, insert a value between 0 and 100 -. Therefore, *-fading 90* has the effect that 90% of the brightness is retained when it is inactive, while with *-fading 50* the window is darkened by 50%. A combination of foreground colour, transparency, colouring and fading can be seen in Figure 2.

## Scrollbar

If you don't want one at all, say so with *+sb*, and those preferring to keep it by way of tradition can simply place it on the right side (*-sr*). But anyone who has just set everything to be transparent may well take exception to the conspicuous appearance of this scrollbar – although there is also a remedy for this: A *-trsb* makes this transparent, too. Figure 3 shows one possible result of these options.

## Brave little tailor?

Now, our Linux should certainly be good-looking, but above all it must be easy to use. And who wants to have to face a five-line series of options on his terminal every time he enters a call-up? The whole

thing would obviously be much easier if you stored the defaults in the file *~/.Xdefaults*. If this file does not exist, it is easy to make. Next, enter the defaults to be altered together with their new values in this file (not simply all options – this creates more confusion; if you select *transparent*, you should no longer specify a background colour...). These are described in full in the Manpage (*man aterm*) and for the settings described in this article these would be:

```
aterm*font: 8x16
aterm*geometry: 80x30
aterm*background: #000040
aterm*foreground: lightgrey
aterm*backgroundPixmap: graphic.xpm
aterm*backgroundType: scale
aterm*transparent: true
aterm*tinting: blue
aterm*shading: 50
aterm*fading: 90
aterm*scrollBar: false
aterm*scrollBar_right: true
aterm*transpscrollbar: true
```

Our next deskTOPia will also be about a memory-saving program: We introduce Sapphire, a small, young Window Manager with extremely low memory requirements.

[top]  
Figure 2: Aterm times two with coloured transparency – the back one shows fading

[below]  
Figure 3: *aterm -fg lightgrey -tr -sh 60 -sr -trsb*

