

aterm guide - make the aterm terminal look good

v2.0.1, en.

A guide to aterms basic usage and settings

1. [What is aterm?](#)
 - ◆ [1.1. Installing aterm](#)
 2. [Aterms basic command line and configuration options](#)
 - ◆ [2.1. Making aterm transparent](#)
 3. [True Border-less aterm terminal](#)
-

1. What is aterm?

A terminal emulator (also known as console) is the most basic Linux tool. It simply gives you a nice command prompt where you can do your 'work' by issuing cryptic shell commands like `ls`. There are many terminals for [xfree](#) available. Most people know `xterm`, the default terminal installed with [xfree](#). [kde](#) comes with a nice terminal called [konsole](#). But there are other alternatives. Let's have a look at [aterm](#).

Aterm is a vt102 terminal, meaning you can use `alt+left/right/up/down` to move between windows in [irssi](#) - [user guide](#). It is highly customizable and can be set to look, feel and behave according to most desires. More importantly: it is lightweight, requires close to nothing and depends on few libraries, making it the ideal choice for a light and simple desktop.

1.1. Installing aterm

- [Mandrake Linux](#): aterm is included on the [Mandrake Linux](#) install CDs, install as usual using `rpm-drake`.
- [Gentoo Linux](#) users can install aterm with the command `USE="-cjk" emerge x11-terms/aterm`. (If you compile aterm with `+cjk` (support for Multi-byte character languages) Scandinavian chars like `æøå ÆØÅ` are screwed! Asia/Japan/civilized part of the world: Go for `+cjk`).
- Packages for Debian, Red Hat, SPARC and Alpha are available on [aterms](#) homepage.

2. Aterms basic command line and configuration options

The default terminal window you get from `aterm &` is very dull, and many people are fooled by this to believe aterm is just another useless `xterm` clone.

aterm guide - make the aterm terminal look good (Linux Reviews)

aterm can be configured in the files `~/.Xdefaults` and `~/.Xresources` or by running it with command line parameters.

You really should configure aterm if you plan on using it. Transparency is good as long as you are running a workstation, thin client users should stick with a plain background color.

The most important basic settings are:

```
-ls , +ls    Start as a login-shell/sub-shell; resource loginShell.
```

Aterm defaults to starting in a sub-shell where you lose your environment (your nice command prompt, aliases and other variables) and you get a dull `sh-2.05b$`. Use **aterm -ls** to make aterm inherit your systems settings.

Example `~/.Xdefaults` setting:

```
aterm*loginShell:true
```

If your distribution did not provide a lovely prompt, then try adding one of the following to your login-script:

```
export PS1='\[\033[01;32m\]\u@\h \[\033[01;34m\]\W \$ \[\033[00m\]'
```

or

```
export PS1='\[\033[01;32m\]\u@\h \[\033[01;33m\]\W \$ \[\033[00m\]'
```

(Refer to `tlpd:Bash-Prompt-HOWTO` for more information)

To make Aterm (and others) display your current path in the windowtitle, add:

```
export PS1=$PS1"\[\e]0;\H:\w\a\""
```

Please note that this does not work well with Midnight Commander!

```
-sl number
```

Save number lines in the scroll back buffer; resource `saveLines`.

The default setting is rather low, you probably want to be able to scroll back a higher number of lines. After some experimenting I found the maximum setting is **32767** lines. Setting 32768 gave jumped back to aterms default setting (`aterm -sl 32767`).

Example `~/.Xdefaults` setting:

```
aterm*saveLines:32767
```

aterm guide - make the aterm terminal look good (Linux Reviews)

`-bg colour` Window background colour; resource background.

`-fg colour` Window foreground colour; resource foreground.

If you do not like the default black on white setting, use **-fg** and **-bg** to change the colors. Use the plain color names (like White, Black, Orange, Green, etc). Add Light or Pale in front of the color names for variations (PaleGreen, LightSlateBlue, LightSkyBlue, etc).

```
aterm -fg white -bg black
```

Example `~/.Xdefaults` setting:

```
aterm*background:Black aterm*foreground:White
```

Setting up nice default fonts

The syntax for font configuration (from the aterm manpage):

`-fn fontname` Main normal text font; resource font.

`-fb fontname` Main bold text font; resource boldFont.

`-fm fontname` Main multicharacter font; resource mfont.

Use the utility `fontsel` (included in [openmotif](#)) to find a nice font. You can also use `xlsfonts` (included in [xfree](#)) to view all fonts and sizes available on your system:

```
xlsfonts | less
```

Example settings in `~/.Xdefaults`:

```
aterm*font:*-*-fixed-medium-r-normal--*-140-*-*-*-iso8859-1
aterm*boldFont:*-*-fixed-bold-r-normal--*-140-*-*-*-iso8859-1
```

2.1. Making aterm transparent

`-tr , +tr` Turn on/off pseudo transparency mode. In this mode aterm will use part of root background as it's own background, emulating see-through effect; resource transparent

`-trsb , +trsb` Turn on/off pseudo transparent scrollbar. Works the same as `-tr`; resource transpscrollbar.

`-sh amount` This option allows for darkening/lightening of the background image amount is the %value of the desired brightness, where 100 is the original. if amount is less then 100 - image will be darkened. if amount is less then 0 or more then 100 - image will be lightened. Lightening can cause some strange looking effects if applied on bright image. This option causes aterm to create shaded/lightened copy of the background - it's more mem-ory cosuming as the result. It does not make much difference, though, if applied on non transparent background image. resource shading.

These settings go hand in hand. **-tr** makes aterm transparent, **-trsb** makes the scrollbar transparent aswell. **-sh** lets you shade your root background image in order to keep the actual text in the image visible no matter how light or dark your background image is.

aterm guide - make the aterm terminal look good (Linux Reviews)

```
aterm -tr -trsb -sh 60
```

Example `~/.Xdefaults` settings:

```
aterm*transparent:true aterm*transpscrollbar:true aterm*shading:60
```

You could use these settings in your `~/.Xdefaults`: [Xdefaults.aterm.txt](#)

```
aterm*loginShell:true
aterm*transparent:true
aterm*shading:60
aterm*background:Black
aterm*foreground:White
aterm*scrollBar:true
aterm*scrollBar_right:true
aterm*transpscrollbar:true
aterm*saveLines:32767
aterm*font:*-*-fixed-medium-r-normal--*-140-*-*-*-iso8859-1
aterm*boldFont:*-*-fixed-bold-r-normal--*-140-*-*-*-iso8859-1
```



- Elisha Cuthbert:
 - ◆ [elisha-cuthbert-1.jpg](#)
 - ◆ [elisha-cuthbert-3.jpg](#)
 - ◆ [wallpapers](#) (elisha-cuthbert.com)

graphix wants you to know: You can fade your aterms

`-fade x%` makes not atvice aterm windows fade. graphix really likes to start aterm like this:

```
aterm -ls -tr -trsb -fg gray -fade 60 (hmm why doesn't -trsb work wonders graphix)
```

3. True Border-less aterm terminal

The borders surrounding aterm are placed there by your window manager. Check your window manager to find out how you can make windows border-less.

aterm guide - make the aterm terminal look good (Linux Reviews)

aterm running in fluxbox, a theme with only borders on the top and bottom

KDE 3.2+ users can turn off the borders by selecting `Advanced -> No Border` in the window menu. You can activate the window menu with the keyboard shortcut `Alt+F3` or by right-clicking once on the window decorations.

Aterm version 1.00.beta3+ has a `-bl` switch that tries to ask the window manager to create the aterm window borderless. This works with Fluxbox version 0.9.12 and the latest versions of aterm.

Thanks to Gerikes Markholm

Elisha Cuthbert pictures are copyright [Elisha Cuthbert](#) (used with permission)

Text Copyright (c) 2000-2005 [GNU Copyleft](#) xiando Corp and Gerikes Markholm

> [Linux Reviews](#) > [Software guides](#) > [x11 Terminals](#) >
aterm guide - make the aterm terminal look good