

Jo's alternative desktop

ROOT-TAIL

JO MOSKALEWSKI



Only you can decide how your Linux desktop looks. We 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.

We all know the scenario: Something in the system is hanging, and a tedious look into the log files under */var/log/* is necessary to track down the evildoer. Even when there are no acute problems, it is clearly better to be kept informed at all times so as not to have to go troubleshooting in the log files when it's too late. With *Root-tail*, all the information you want from log files you can obtain, without mouse clicks, on the desktop background.

Something special

There are numerous tools which track log file entries and report their (hopefully) good news. But who wants to keep a window open on their desktop all the time, just to note the sign of life from the syslog every 20 minutes? Surely it would be much nicer to have this directly on the desktop: No separate window, which has to be closed, opened or moved around, and nothing to interrupt the beloved background graphics. Such a program does exist: *Root-tail*.

Off we go!

If your own distribution does not have a finished *root-tail* packet among its baggage, it will help if you go to the source code of the program or take a look at <http://www.goof.com/pcg/marc/root-tail.html>. The archive that you can get there is easy to install. The basic requirements are (as ever) the X-Includes, which can be found in SuSE in the *xdevel* packet. If the X-Includes are on the hard drive, the first step in unpacking the archive is done. After that, create a Makefile with the tool *xmkmf*, which the following tool *make* uses to read off exactly what needs to be done. Consequently, you don't need a lot of experience, as your system does everything for you. After that, a *make install* and *make install.man* follow, with which just the completed program and its documentation are copied to the right places and the correct rights are set:

```
jo@planet ~$ tar xvzf root-tail-0.0.10.tar.gz
jo@planet ~$ cd root-tail-0.0.10
jo@planet root-tail-0.0.10$ xmkmf -a
jo@planet root-tail-0.0.10$ make
```

```
jo@planet root-tail-0.0.10$ su -
Password:
root@planet ~# cd /home/jo/root-tail-0.0.10
root@planet root-tail-0.0.10# make install
root@planet root-tail-0.0.10# make install.man
root@planet root-tail-0.0.10# logout
```

Nothing happening?

The syntax to start *Root-tail* is simple: As with almost all programs, you obtain information on this by entering the option *"-help"* as you go along – thus *"root-tail -help"*. Anyone who wants it a bit more precise and extensive should consult the man page, which can be found via *"man root-tail"*. If you want to place the popular log file */var/log/messages* on the desktop, you might use the following command:

```
root-tail /var/log/messages
```

But since, as a prudent Linux user, you do not start an X-session as *"User root"*, a problem arises: *root-tail* answers with a curt *"/var/log/messages: Permission denied"*. Quite right too, because these files are nothing to do with the normal user. So *root-tail* has to be started with *root-rights* to do this.

Super

A good tool allowing a user to start a specific program with *root-rights* is *super*. *super* should come with every distribution. *super* is configured with the file */etc/super.tab*, in which we simply allow the user *jo* also to start *root-tail* as superuser:

```
root-tail /usr/X11R6/bin/root-tail jo
```

The first word states which command is involved (in this connection new names can also be invented), followed by the command to be executed, and lastly the name of the user who is to be authorised to start the command as *root* (additional user names can be added easily, separated by a comma). After this simple configuration, *root-tail* starts with *root* rights via the command



```
super root-tail /var/log/messages
```

But here again, the desired log file still does not appear on the user desktop, and instead *root-tail* responds with "Error opening display (null).":

Accessing the X-Server

Even *root* cannot simply use some other desktop illegally as output medium: The X-server has its own access control. This is governed in the hidden file *~/.Xauthority* and as the user *root* can read the data of every user, it can simply make joint use of that of the user. The simplest variant of this is the setting of the environment variable *XAUTHORITY*:

```
export XAUTHORITY=/home/jo/.Xauthority
```

But that's not all: As *root* has not started an X-server, this permission is insufficient because our program cannot make use of this until it also knows that an external display is to be used. Here again, the setting of a variable helps:

```
export DISPLAY=:0.0
```

Before what is now being described is pulled together, here is a little test, which illustrates the necessary basis and ought to function:

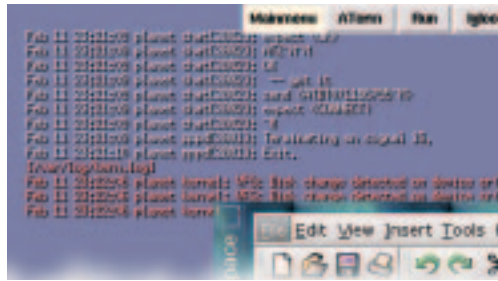
```
jo@planet ~$ su -
Password:
root@planet:~# export XAUTHORITY=/home/jo/.X
authority
root@planet:~# export DISPLAY=:0.0
root@planet:~# root-tail /var/log/messages
```

In this case, *root-tail* can be ended with the key combination *Ctrl+C*.

Puzzle

Since now as *root* two variables still have to be set before starting *root-tail*, this can no longer occur with a simple command. A little script can help here – although it is then no longer *root-tail*, but the entire script which is started via Super (the variable, too, must set *root*, if they are valid for it). It is best to make a file */usr/local/sbin/root-tail.username* with the following content:

```
#!/bin/sh
export XAUTHORITY=/home/username/.Xauthority
export DISPLAY=:0.0
/usr/X11R6/bin/root-tail /var/log/messages
```



[left]
Figure 1:
Root-tail on the desktop

[right]
Figure 2:
root-tail -g 120x13+20+20 -color
mistyrose1 -fn 6x13 -shade
/var/log/messages
/var/log/kern.log,red

Since scripts called up via tools such as Super can easily be misused as the result of modifications, the file should only be read/write for *root* (otherwise anyone who could change the file and start it via *super* could start any programs they liked in it simply by adding!). Therefore type as *root*:

```
chown root.root /usr/local/sbin/root-tail.u
sername
chmod 700 /usr/local/sbin/root-tail.username
```

And then adapt the configuration file *super.tab*:

```
root-tail /usr/local/sbin/root-tail.username jo
```

And from now on */var/log/messages* appears on the user desktop after entering *super root-tail*.

Admittedly, because of the user rights and the access to the X-server, there is a lot of work initially, but once you have understood and applied this, it's easy: *super* releases any programs for one or more users, and a script allows graphic output. This method can be transferred to countless programs – to your favourite file manager for example, which you can then place a second time, but with *root* rights, in the Start menu. This then starts without any additional password challenges whatsoever. And that's worth a bit of effort.

Personal edition

root-tail can obviously be adapted to your own requirements. The most interesting options here could be the following:

-g states where Root-tail should appear on the desktop, and also the number of characters to be shown is defined hereby. A "-g 120x13+20+20" moves Root-tail by 20 pixels away from the edge of the desktop and sets its size at 13 lines, each with 120 characters. Another helpful option here is **-frame**, which can be used for test purposes to display a frame until the optimal geometry specification has been found.

-color sets a standard font colour. Each log file can also be given its own, by specifying a colour name when calling up the log file: "*/var/log/messages,green*" (see Figure 2).

-font: By their nature, fonts with a fixed width are most suitable here, thus 5x7, 5x8, 6x10 etc up to 12x24. Exactly which ones are available depends on the distribution being used. The tool *xfontsel* can help with the selection.

-shade gives the letters a shadow. ■

Note with respect to KDE 2

KDE 2 lays a frameless window over the entire desktop and thus covers everything another program paints onto the desktop. We do not know of any solution for using root-tail in conjunction with KDE 2.