

Desktopia WINDOW PICKER

Is your `.xinitrc` a permanent building site? Jo Moskalewski gets your house in order with `selectwm`



Do you fancy an ABC of window managers and desktop environments? Well, here goes: AfterStep, Blackbox, CTWM, dxwm, evilwm, FVWM, GNOME, HeliWm, IceWm, KDE, lwm, MWM, NovaWm, OLWM, PWM, qwm, Ratpoison, Sapphire, twm, UDE, VTWM, wmx, XFce and YAWM. Admittedly, there's no J or Z just yet, but they'll surely join the crowd at some point. Nevertheless, there's no denying that we have a huge number of interfaces to choose from. (By the way: the alphabet could also be made up from completely different window managers). All that's missing now is a tool via which the user can select one of the installed window managers and desktop environments for the next session.

This gap in functionality has been neatly filled by Luc Dufresne, who has given the fruits of his efforts the name `selectwm`; made it available for free download from his Web site, <http://ordiluc.net/selectwm/>; and placed the program under the GPL.

Rules of the game

The principles of the program are simple and effective: instead of a window manager, `selectwm` will start in future. This program in turn offers a choice of the window managers and desktop environments entered by the user (with the option of starting a default entry automatically after a pre-defined time limit). If required the user can go back to `selectwm` after shutting down the window manager – otherwise the user Xsession is ended immediately.

So as not to complicate matters needlessly, `selectwm` needs no separate configuration tool for all this, but can be configured completely on the fly within the one interface by the use of a mouse.

Even a perfect tool needs a bit of help every now and then: it would be pointless if every programmer had to paint the buttons and texts in their programs themselves. In general the so-called "Toolkits" are used for this, and these are available as libraries for several applications. `selectwm` relies on the GTK+ toolkit in version 1.2.0 or higher, which should be familiar from Gimp and GNOME. Don't worry,

deskTOPia

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.

though, this minimum requirement stems from February 1999, and so any halfway recent distribution should easily meet this.

Preparation

It's still up to you to check that the associated Developer package is also installed on your system. This contains parts of the GTK+ library, which users of pre-compiled distribution packages do not need. Only those who want to compile or install the corresponding source code must retain these. `selectwm` comes as source code, but if you're using the Debian distribution, there's an up-to-date, ready compiled deb package, which can be installed directly with `dpkg`.

Even if you don't use Debian, self-compilation is not all that much harder. Once GTK+ is on the hard drive, the following steps (entered at an input prompt) deal with the rest of the installation:

```
tar xvfz selectwm-0.3.tar.gz
cd selectwm-0.3
./configure
make
su
make install
logout
```

The tool `tar` unpacks the archive, at which point the `configure` script, which comes with it, interrogates your system. In its output, you can ignore a simple "no" – but if clear indications of any errors appear, or if `configure` stops without creating a Makefile, you'll first have to correct the reasons for this. You can find whatever is wrong from the script output.

If no errors occur, you will finally be prompted to proceed with the command *make*. This tool translates code written by the programmer using the previously made Makefile into a code the Linux kernel can execute; the binary is created. You can get to the right place with the command *make install*, and to execute this the user root must briefly enter the source directory.

Casting

If *selectwm* is now in place on your hard drive, the next thing to do is to allow this tool to control your Xwindow system. If a user has an (executable) file in his home directory called *.xinitrc* (in the case of a text-based log-in) or *.xsession* (in the case of a graphical log-in), then this will control the progress of an X session at user-level. This is dealt with in exactly the same way as a shell script: as soon as the last command in it is complete, the whole script comes to an end. If *selectwm* is to come into play, you could make this file as follows:

```
#!/bin/bash

xsetroot -solid black
selectwm
```

First call up everything which is to remain the same in every X session, regardless of which window manager is being used. In this example *xsetroot* starts by colouring the background black. Lastly, *selectwm* takes up the baton.

The game begins

When you first start you will be met by a completely blank window, as in Figure 1. So that you can now actually choose something, you must first create the future selection list. Press the right mouse button within the clear, white window area. A pop-up menu will appear and you can now add the description and start command for a new window manager entry (Figure 2).

This list is stored (together with the rest of the configuration) automatically in the file *.selectwmrc*. If you have already entered your window manager and desktop environments, you can also use the right mouse button to make an entry the default – this will then be selected in future unless you manually choose a different alternative. If you no longer like the sequence of selection options, then simply drag them with the left mouse button to the right spot. A double-click, on the other hand, starts the entered command. The selection list also responds to the arrow keys and space bar.

Time limit?

There must be one or two of you wondering, when you are experimenting, what is the point of the so-far

blank field immediately below the selection list? This question can be answered by pressing the Config button. A further dialogue opens, in which a length of time can be defined to the precise tenth of a second: Once this has elapsed the default window manager starts. The remaining tenths are counted down in this very field (Figure 4). If on the other hand one sets the time limit to "0", *selectwm* awaits a user action.

There's also another button, labelled "Go back to selectwm when the WM exits". If you select this, when your window manager is shut down you will return to *selectwm*. You can also pre-set this in the Config dialog (Figure 3).

As you can see from Figure 4, the visual appearance of *selectwm* can be modified. But there are strict limits imposed on the freedom to do so: Since this is a GTK application, its appearance is in line with the current GTK theme. You can select this, for example, in the GNOME Control Center (*gnomecc*) or else in the palette dialog of the XFce desktop.

Trick 17

Anyone who only starts large interfaces like XFce or Window Maker will be delighted with the options described so far for selecting their window manager. But what if, for example, a window manager comes onto the desktop with its own clock, while the next one graces the desktop without any trimmings at all – and a clock would thus have to be started manually, as well? There is a simple solution – the desktop environments XFce and KDE show you how: their interfaces are not invoked in one go, but are assembled behind the scenes using the scripts *startkde* or *startxfce*. All the necessary tools are additionally started by these. Maybe you want to provide the window manager PWM with a clock at all times (which is not necessary with another interface), in which case you make an executable shell script named *startpwm* in the */usr/local/bin* directory. There you should enter, before actually starting the window manager, your clock command:

```
#!/bin/bash

oclock -geometry -0-0 &
pwm
```

You can expand any simple window manager by an autostart function using this type of wrapper. This is also the way to realise various configurations with one and the same window manager. In future there will also be no need for fiddly modification of your *.xinitrc* before starting an X session.

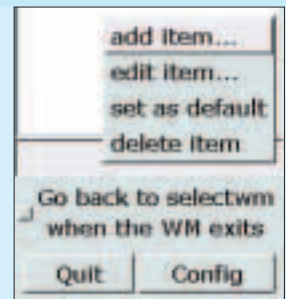


Figure 1: Right click in the still-blank *selectwm* window

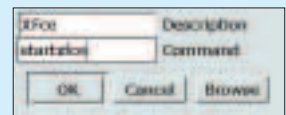


Figure 2: A new interface is added

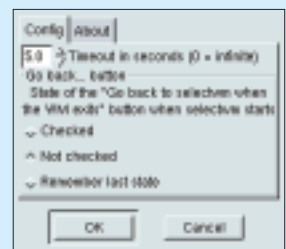


Figure 3: Behind the "Config" button

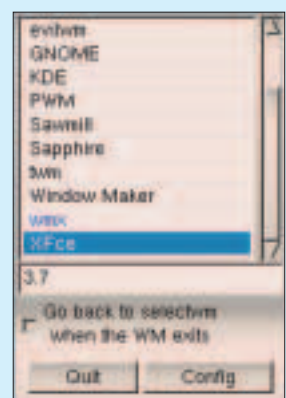


Figure 4: Theme-capable