Comparing screencast applications

DESKTOP RECORDERS

Screenshots provide a static impression of an active program, whereas desktop video sequences show the software hard at work. We'll introduce you to some of the major tools for creating screencasts. **BY OLIVER FROMMEL**

Screenshots are just not good enough. Today's open source projects need to invest in marketing to attract users and fellow programmers, and a screenshot gallery is just a matter of course. Many of today's projects take things a step further and offer screencasts, video clips that show their software at work. Screencasts are also useful as web tutorials.

A number of programs help users create screencasts. Many of the candidates come from the Gnome camp, which is always open to the latest trends. This article investigates some of the more widespread offerings.

Xvidcap

The oldest contender has been around since the days before screencasts. Xvidcap [1] is a simple X tool that launches with a tiny menu and draws a red frame around the recording area (Figure 1). Xvidcap can optionally use the Gtk toolkit, but it will also run on KDE, Gnome, and other window managers.

It can be difficult to build Xvidcap from the source code; the Ffmpeg dependency is particularly tricky, and the program needs Ffmpeg to encode the resulting Mpeg. The *--with-forced-embedded-ffmpeg* option might help if you are using the Ffmpeg that comes with the distribution. Otherwise, you may need to search the Internet for an alternative binary for your distribution. Xvidcap is the only program in the test field that gives you the option of recording a sound track while recording the images. This lets you comment on what you are doing on screen.

During recording, Xvidcap simply stores the individual XWD-formatted images in the current directory. You will probably want to create a directory for this purpose and change to the directory before you launch Xvidcap.

Istanbul

One of the more recent programs, and one that many Gnome projects use, is Istanbul [2]. This rather unusual name is to celebrate Liverpool FC winning the Champions League in Istanbul says developer Zaheer Abbas Merali. It outputs the video clip in Flash format, which is proprietary but fairly well-suited to web presentations.

If you have Gnome 2.12, the installation should be fairly simple. Istanbul requires Gstreamer version 0.8.10 – Gnome 2.14 includes Gstreamer 0.10, which is not downwardly compatible. Apart from this, Istanbul also needs Python-Gtk 2.6 because it was written in Python. On Ubuntu 5.10 we had no trouble installing Istanbul directly from the repository, although we did need to remove the *pygtk.require('2.0')* line from the *istanbul* script.

On Gnome and KDE a red record button appears in the panel when you launch Istanbul. Unfortunately, we couldn't use the Istanbul menu on KDE due to a display error. As a workaround, you could try docking the KDE panel at the top of the screen. The menu lets users set the image size and frame rate. One special feature available here is the ability to opt to stream the video you are recording directly to an Icecast server. If Istanbul is too slow, on an older machine for example, you can use the menu to tell Istanbul to record invidual images

and encode them when you are finished recording.

Byzanz

As the name suggests, Byzanz [3] is related to Istanbul. (The name is based on Istanbul's former name Byzantium.) Byzanz works in a similar way to Istanbul, but it creates an animated Gif rather than a Theora video; and this gives users the freedom to view the results in more or less any web browser without needing special codecs or plugins.

Byzanz is fairly simple to compile, although it does require a compositing manager, which restricts use to systems with XGL or AIGLX. The program failed to convince in our lab. After initial setup difficulties, it did manage to create an animated Gif showing the screencast, but the whole video was colored red. As Byzanz is at an early stage of development, I would tend to regard it as a future option.

VNC2SWF

As the name suggests, VNC2SWF [4] requires the VNC remote desktop system, which it leverages to record screencasts. This should be no problem on a recent distribution where both Gnome and KDE come with remote desktops based on a variant of VNC.

As an alternative to the C version, the developers also offer a Python variant, dubbed PYVNC2SWF. Besides the script for animated screenshots, the package also includes tools for cutting or viewing the resulting video, both of which require the Pygame and Pymedia Python libraries.

The command line program *vnc2swf* is less than intuitive. It not only expects the name of the host on which the VNC server is running, it also starts a virtual X server in which you need to run the programs you will be creating screencasts of. To remove this need, you can specify the *-nowindow* option when launching *vnc2swf*.

The Python version, vnc2swf.py, makes things simpler by launching a GUI in which users only need to specify the name of the resulting Flash file.

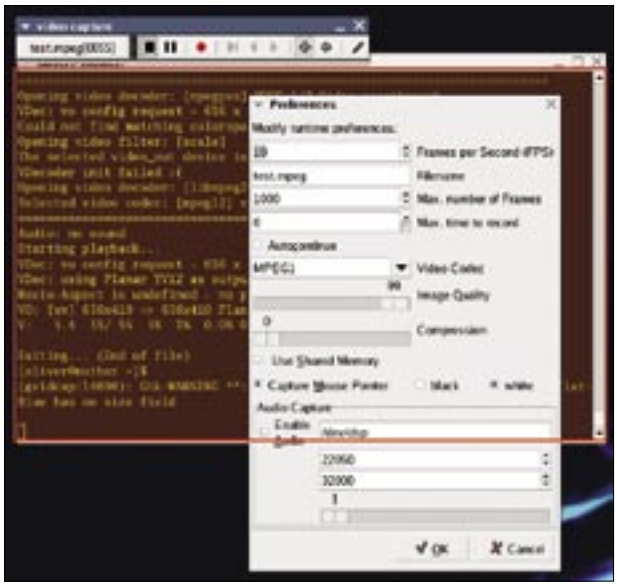

Figure 1: Xvidcap draws a red frame around the recording area.

Wink

Good software is not always free. Although the Wink [5] screencast tool is free as in beer, the source code is not provided. This said, the install is fairly trouble-free: just unpack and run the install script, which doesn't even need root privileges. The program is linked against the current Gnome and X libraries, which need to be installed, but it does not require a specific desktop environment at runtime.

After you launch Wink, an application window appears offering a number of menus. Selecting *File* | *New*, or clicking the corresponding icon, takes you to a dialog where you can set the record frame, which can be a specific window, the whole screen, or an area defined by dragging the mouse. After you confirm, another dialog appears with a short note on using the program (Figure 2). At this point, you can hide all the Wink windows and focus on using the hotkeys to control the program. *[Break]* takes a single screenshot; *[Shift]* + *[Break]* starts and stops continuous recording. The stop command only interrupts the recording, giving users the ability to compile movies from several scenes recorded in succession.

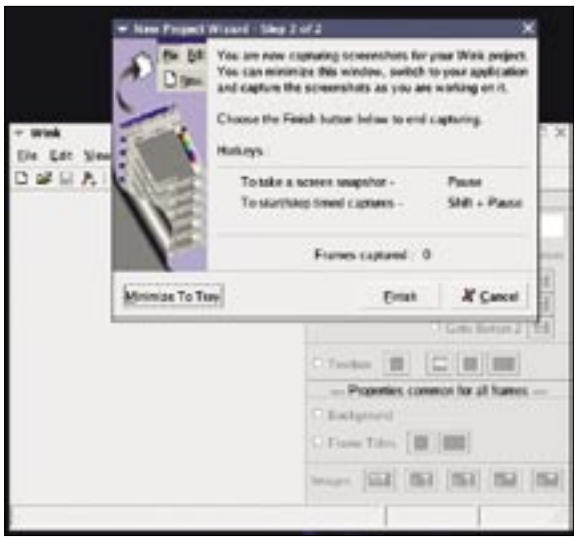

Figure 2: Wink provides hotkey controls for recording.

To quit recording, users can press the *Finish* button. Wink will then show you the individual images it has recorded for editing. When you are done, select *Project* | *Render* to tell the program to create the final Flash movie.

Conclusions

The clear winner with respect to featurescope, usability, and ease of installation is Wink. Wink even gives users the ability to post-process individual images before rendering a Flash video. But Wink is not an option for those who insist on free software, as the freeware tool does not include source code.

Xvidcap is a tried-and-trusted program without much in the line of bells and whistles; but at least you can work with it. If your Linux desktop supports remote desktop functionality, you might like to try VNC2SWF. In the meantime, you can look forward to its successor, a Python application that will likely include some useful improvements. ■

INFO

[1] Xvidcap: *http://sf.net/projects/xvidcap*

[2] Istanbul: *http://live.gnome.org/Istanbul*

[3] Byzanz: *http://www.advogato.org/person/company/diary.html?start=18*

[4] VNC2SWF: *http://www.unixuser.org/~euske/vnc2swf*

[5] Wink: *http://www.debugmode.com/wink*