

## TV and Videotext on Linux

# For Your Viewing Pleasure



Whether you want to watch TV, read videotext, or even record a TV program, Xawtv provides the functionality you need. Also, the software package can display images from webcams and provides convincing stability and performance.

BY GERD KNORR

**X**awtv [1] started life as a front-end for a single TV card driver, but gradually developed into a comprehensive program suite. The Xawtv package provides full viewing pleasure on Linux. The former Bttv driver made its way into the kernel as Video4linux, and Xawtv now supports any hardware that supports the Video4linux **API**. The suite includes the TV viewer programs *xawtv*, *motv*, *fbtv*, and *ttv*. *mtt* and *alevtd* provide teletext support and *webcam* can put your webcam images on the wire.

Thanks to the Openmotif library, the *motv* UI appears more modern than *xawtv*, and it provides a few additional features. *fbtv* runs on the Linux framebuffer console, providing viewing facilities without X.

*ttv* is a fairly unusual program. It allows TV viewing on the console and uses ASCII characters to display images [4]. This allows you to use SSH to check

a webcam on a webserver without needing to launch an X server.

## Installation

Xawtv is part of most Linux distributions standard equipment. The individual packages are typically distributed between multiple packages. Thus, you cannot simply install the *xawtv* package to access the complete range of functionality we will be discussing in this article.

The current SuSE version uses the following call to YaST2 to install the required packages:

```
yast2 -i xawtv motv alevtd ↵
v4l-tools
```

Debian provides even more granular distribution of the packages, which simply means that Apt will need a few more package names:

```
apt-get install xawtv motv fbtv ↵
```

```
ttv radio xawtv-tools webcam ↵
alevtd tv-fonts
```

*motv* and Openmotif are located in the *non-free contrib* category, and should be listed in */etc/apt/sources.list*. The current stable version of Debian unfortunately does not include the *tv-fonts* package, and this may lead to problems when displaying teletext pages.

If you prefer to compile the latest tarball yourself, you can use the typical *make* && *make install* steps to install Xawtv. To additionally compile *motv*, *mtt*, *scantv*, and *alevtd*, you will need Zvbi [2] and Openmotif [3]. RPM users can create an RPM package from the tarball, by passing the tarball directly to *rpm*:

```
rpm -ta xawtv_3.88.tar.gz
```

You can then use the standard *rpm -i* syntax to install the finished package.



Figure 1: *motv* offers the user a more modern UI and more features than *Xawtv*

## TV Viewing

Simple viewing is the task typically required of *Xawtv*, even though it supports a lot more hardware than just a TV card. The program can display USB webcam images on screen, for example.

You might prefer to launch *xawtv* or *motv* while familiarizing yourself with the suite. *fbtv* and *ttv* have fairly basic UIs and are not really that useful without a configuration file.

Type *xawtv -hwscan* or *motv -hwscan* to display an overview of the installed hardware. This is particularly useful if multiple devices are attached to your

module in this case. The GATOS project [5] is developing special drivers that will provide access to ATI video card TV hardware via an Xvideo port; this is good news for users with ATI graphics adapters.

X11 applications can use Xvideo, port 54, to scale images from the graphics adapter. When playing videos, media players such as Xine or Mplayer can thus offload part of their task to the graphics adapter, thus reducing the load on the CPU. *Xawtv* can also use this port to display frames in grabdisplay mode (see boxout “How an image ...”).

machine, a USB webcam and a TV card for example, as shown in Listing 1. Depending on the device, *xawtv/motv* will use the X server to access the device’s ports or access the device file directly.

The port shown in the list, port 53, is an Xvideo port. The port handler that interacts between this port and the Bttv driver is the X server’s V4l

The first Video4linux device – a USB webcam in this case – is accessed as */dev/video0*. Applications will then use */dev/video1* to access the second device; this is a Bttv card by Pinnacle in Listing 1. Programs can also access this adapter via Xvideo port 53, however, *Xawtv* uses the Bttv driver directly without taking the detour via the X server in this case.

*xawtv* will access the first entry in the hardware list by default. To use another device, you have to specify the *-device* option when launching the program. You can type *xawtv -hwlist* to display the correct parameters. The following call tells *xawtv* to use the USB camera in Listing 1:

```
xawtv -device /dev/video0
```

When you launch *xawtv* or *motv* for the first time, you will simply see static on the TV screen. Right click to open a second window with a range of preferences. There are keyboard shortcuts for many functions. For example, you can use the arrow keys to control the tuner and search for TV channels. Keyboard shortcuts are supplied along with the menu entries, and the manpage additionally provides an overview.

As channel searching with the arrow keys is not much fun, *motv* also provides an automatic channel search, which you can access via the *Options / Channel scan* menu. *Options / Save configuration* tells *motv* to store the channels you have discovered in a configuration file, which is also used by *xawtv*.

If you prefer to perform this step manually, you can launch *scantv* instead. This program automatically searches for channels and stores the results in *\$/HOME/.xawtv*. You can simply edit the configuration file to modify the channel list. [Page Up] and [Page Dn] or the space key will zap from one channel to

## Listing 1: Xawtv can generate a list of video devices in a system

```
$ xawtv -hwscan
This is xawtv-3.88, running on Linux/i686 (2.4.20-0611-suse)
looking for available devices
port 53-53                                [ -xvport 53 ]
    type : Xvideo, video overlay
    name : video4linux

port 54-54
    type : Xvideo, image scaler
    name : Matrox G-Series Backend Scaler

/dev/video0: OK                            [ -device /dev/video0 ]
    type : v4l
    name : Logitech USB Camera
    flags: capture

/dev/video1: OK                            [ -device /dev/video1 ]
    type : v4l
    name : BT878(Pinnacle PCTV Studio/Ra)
    flags: overlay capture tuner
```

## GLOSSARY

**API:** “Application Programming Interface”. In this case an interface between the programs and the Linux kernel. A program uses the API functions provided by the Linux kernel to access the hardware driver.

**Xvideo:** A video extension for the X server that provides X11 programs with two additional features: video hardware control and color scheme conversion from YUV to RGB, which allows scaling of video frames.

the next in *xawtv/motv*. The scroll wheel on your mouse, if available, will do the same job.

## Grabbing Frames to Disk

*xawtv*, *motv* and *fbtv* all provide a point and click function for grabbing individual frames to disk. You can hit the [G] key to store a frame in PPM format and [J] for jpeg. Files are stored in the current directory using the following convention “snap-channel-timestamp.jpeg”. Additionally, a program called *v4lctl* is available for grabbing frames from video devices in the command line:

```
v4lctl snap jpeg
```

In contrast to *xawtv*, *motv* is one of the few viewer programs besides Kview from KDE 3.1 that supports cut-&-paste. You can launch *motv*, select a channel and press [Ctrl-C] to grab a frame. You can then press [Ctrl-V] in Kview to insert the image.

## Recording Movies...

Both *xawtv* and *motv* provide facilities for recording video and sound sequences. Press the [R] key to display a dialog box where you can specify para-

meters for, and launch the recording. Do specify a file format before you start, as the format will determine the available video and audio features. AVI or Quicktime movie formats are typical, but *xawtv* also supports uncompressed raw data, which can be used to create mpeg 2 files, *mpeg2enc* [6] for example.

If you intend to use the command line, you should take a look at *streamer*. this program basically provides the same functionality as *xawtv* or *motv*, but without needing a GUI. *streamer -h* outputs a list of the available options and default settings, additionally providing one or two examples.

Pay attention to the mixer settings for sound recordings. What you hear (through your speakers) is not necessarily what you get (in the recording). There are two main issues to recording: As many drivers automatically assume the microphone as the source, the source for the recording is not configured correctly as the TV card will

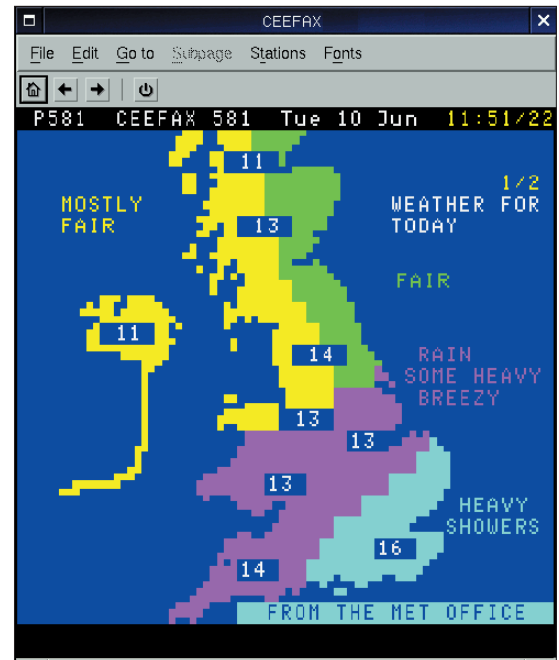


Figure 2: Thanks to Unicode, teletext pages can be viewed directly in the mtt browser and copied to other programs

typically be attached to the line-in jack. And occasionally, the *Igain* controller is turned down on the card.

Of course, you can use any mixer program, such as *aumix*, to specify the mixer settings. *motv* also provides an integrated recording level monitoring tool (*Tools / Record Level Monitor*); you can use the tool to find the right mixer setting and adjust recording levels.

## ...and Play

The Xawtv package has its own simple movie player, which is capable of playing any AVI and Quicktime movies you create using *xawtv*, *motv* or *Streamer*. *pia* expects a file as an argument and simply plays the file. You can hit [Q] to exit *pia*, but there are no other interactive features. You can also launch the program from within *xawtv* or *motv* by clicking the *playback* button.

Mplayer and Xine should have no trouble playing movies created by *xawtv*. Windows causes the user far more problems, as the OS does not provide a Quicktime player by default; some older versions do not even include an Mpeg codec.

## Teletext

TV channels not only provide images and sound, but information via teletext. Some channels also transmit a sender ID

## How an image appears on screen

There are two ways of displaying images from TV cards on screen. The card may use PCI-PCI TRANSFER mode to transmit images directly to video memory via the PCI bus, thus offloading the CPU. In this case the X server or Xawtv will control the TV card. Alternatively, the program may grab individual FRAMES from the TV card and play them as if it were using the video adapter to display a movie on screen. These options provide three different operating modes:

*Overlay with Xvideo extension.* In this case, the X server controls the TV card. The X server thus “knows” that it is handling a video image. If the visible part of the window changes (due to moving, scaling, partially covering the window), the X server interacts between the TV card and the video card to prevent the image from ending up in the wrong window. If the driver and the video adapter work well together, the X server may be able to scale the TV image to full screen mode. But not all video cards will allow this. Additionally, Xawtv cannot store frames or movies in this mode at present.

In *Overlay without Xvideo extension* mode,

Xawtv controls the TV card directly. The card uses PCI-PCI DMA to transmit the image to the video adapter without the X server even noticing. The TV image occasionally ends up in an area of the screen that it should not really occupy, but Xawtv avoids this problem by forcing a refresh, which will be noticeable as the whole screen flickers.

In *Grabdisplay* mode, Xawtv grabs the images from the TV card and outputs them; this tends to avoid hardware problems (some motherboards have difficulty with PCI-PCI DMA transfers in overlay mode). You can also apply software filters (such as deinterlacing) to the image in this mode. However, Grabdisplay does need more resources. If you want to watch TV in full screen mode while compiling a kernel or creating MP3 files, you will definitely need a fairly quick machine.

*xawtv* provides a menu function for toggling between Overlay without Xvideo and Grabdisplay mode. Incidentally, the USB webcam in Listing 1 (-device /dev/video0) does not support Overlay mode, so this leaves only the Grabdisplay option.



Figure 3: Konqueror displaying a teletext page provided by alevtd

and VPS information, using the vertical blank interval, VBI, to do so. Linux can use the `/dev/vbi` file to access this data.

If you intend to use the teletext function, you might like to install the `tv-fonts` package, which unfortunately is not supplied with Debian Woody (3.0). Debian users who intend to use teletext will require either the `tv-fonts` package from the testing distribution, or compile `xawtv` from source code. `tv-fonts` provides the fonts required to display teletext pages (via the `mtt` browser). Without these fonts, the program will be unable to display block graphics (such as channel logos).

You can enter the following command to view these characters:

```
xfd -fn " *-teletext-*-r*-*  
20-*-*-*c-*-iso10646-1"
```

The Unicode character set includes teletext block graphic characters, and this vastly simplifies teletext page handling. The user can simply select Unicode encoding such as UTF-8 and copy the pages directly to another program.

Xawtv provides two teletext viewing programs: `mtt` and `alevtd`. Figure 2 shows `mtt` in action. Users can click on the page number to access a page, or type the number on their keyboards. `mtt` automatically launches in terminal

mode, if there is no X server running; the `-tty` command-line option also selects terminal mode. A teletext font and UTF-8 make a really pretty picture. `alevtd` is a web-server for videotext: it runs as a background daemon and renders HTML pages in clear-text with UTF-8. The browser does need advanced CSS and Unicode support; you will not get far with your old faithful Netscape 4 browser. Additionally, it makes sense to install the `tv-fonts` package on the client. Figure 3 shows a teletext page in Konqueror.

The name “Alevtd” has a historical background. Older versions used the VBI decoder provided by the teletext program, Alevt [7]. Today, the `Zvbi` library [2] is

used instead.

## Bunches of Tools

Besides the TV and teletext programs we already discussed, the Xawtv tarball comprises a collection of useful command line tools.

`v4lctl` provides command line based control for Video4linux devices. It is used to select channels or switch on the TV sound, if you just want to listen to, rather than watch the news for example:

```
v4lctl -c /dev/video1 >  
setstation cnn  
v4lctl -c /dev/video1 >  
volume mute off
```

## GLOSSARY

**PCI-PCI Transfer:** *PCI devices can assume control over the PCI bus (bus mastering) and initiate data transfer autonomously. Generally data is transferred from or to the main memory (DMA – Direct Memory Access), and this is what any modern hard disk controller does. PCI-PCI transfer provides direct data transfer from one PCI device to another, without taking a detour via the main memory.*

**Frame:** *A single image in a video. The European PAL norm specifies that a TV transmitter must send 25 frames per second. Thus in TV reception mode, xawtv must read 25 individual frames per second from the TV card and pass them to the video card to be displayed on screen.*

The manpage lists a whole range of possibilities.

`xawtv-remote` provides similar parameters to `v4lctl`, but does not access Video4linux devices directly, instead passing the command to the current `xawtv` (or `motv`) instance. Thus `xawtv-remote` provides for script based channel switching.

The `record` program records sound only and saves the results in a wav-file. The program includes a recording level meter and provides both an interactive and a command line based interface. `record -h` shows an overview of the available options.

`webcam` grabs images from a camera, optionally adding text and a timestamp and uses FTP or SSH to transmit them to another computer.

Many TV cards have an integrated radio tuner; the `radio` program access the tuner, thus providing radio listening pleasure on Linux.

## Future Plans

So far, users have always had to restart `xawtv` to switch between Video4linux devices and operating modes (with/without Xvideo)

This is scheduled to change in Version 4.0 The new version will also support DVB (digital TV). Xawtv will then be capable of controller DVB tuners and decoding Mpeg-2, thus supporting even cheaper hardware without a hardware decoder.

As these changes require some rearranging of internal functions, and a lot of code is affected by them, it may be a while until the new version appears. ■

## INFO

- [1] Xawtv-Homepage: <http://bytesex.org/xawtv/index.html>
- [2] Zvbi library: <http://zapping.sourceforge.net/doc/libzvbi/>
- [3] Openmotif library: <http://www.opengroup.org/openmotif/>
- [4] Aalib: ASCII-Graphics library: <http://aa-project.sourceforge.net/aalib/>
- [5] Xvideo driver for ATI: <http://gatos.sourceforge.net/>
- [6] MPEG compression: MPEG-Kompression-<http://mjpeg.sourceforge.net/>
- [7] Alevt: <http://www.goron.de/~froese/>