Installing and using MythTV

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Installing and using MythTV

Robert Kulagowski, mailto:rkulagow@rocketmail.com

Initially, installation of MythTV seems like a huge task. There are lots of dependencies, and various distributions seem to do the same thing different ways. This document will attempt to give general installation instructions, as well as including distribution−specific instructions where necessary.

---

14. MythGame.

15. MythMusic.

16. MythWeather.

17. MythVideo.

18. MythDVD.

19. MythNews.

20. Troubleshooting.

21. Miscellaneous.

22. Example Configurations.

1. How to obtain this document.

This HOWTO document is maintained at the primary MythTV website: http://www.mythtv.org by Robert Kulagowski mailto:rkulagow@rocketmail.com. (Please note that I am *NOT* the author/programmer of the MythTV application! See the contributors section below for a list of the programmers.)

This document is available as a single−page HTML document at http://www.mythtv.org/docs/mythtv−HOWTO−singlehtml.html or as a PDF at http://www.mythtv.org/docs/mythtv−HOWTO.pdf.

It's recommended that you join the user list at http://www.mythtv.org/mailman/listinfo/mythtv−users.

Searchable archives for the lists are available at http://www.gossamer−threads.com/archive/MythTV_C2/.

This HOWTO is for MythTV v0.13.

2. Introduction.

This HOWTO document will focus on manually building MythTV in a North American environment. If you have installation instructions for a different region or Linux distribution, please send them to the author so that it can be included in other versions of this document.

Hans Kruse has setup a website that focuses on running MythTV in a non−North American environment. See the website at http://mythtv.nicenemo.net.

Pre−compiled binaries are available for a number of distributions. See section Software::Pre−compiled packages for more information.

Custom mini−distributions are also available to make it easier to install MythTV. A mini−distribution removes many of the "general purpose" workstation / server software packages that may be installed by default if you use one of the big−name OS packages.

See http://linpvr.org/ if you'd like to install MythTV onto a VIA Epia M machine.

See http://mysettopbox.tv if you'd like to install a custom version of Knoppix optimized for MythTV.

The contributors to MythTV are:

- Lead developer: Isaac Richards, mailto:ijr@po.cwru.edu
- Developer: Matt Zimmerman, mailto:mdz@debian.org
- Developer: John Danner, mailto:jdanner@untzuntz.com
- Developer: Chris Pinkham, mailto:cpinkham@bc2va.org
- Overall structure of HOWTO and Mandrake 9.x documentation: Robert Kulagowski, mailto:rkulagow@rocketmail.com
- Gentoo documentation and package: Tony Clark, mailto:tclark@telia.com
- Gentoo documentation and package: Max Kalika, mailto:max@gentoo.org
- Debian package: Matt Zimmerman, mailto:mdz@debian.org
- Mandrake 8.2 documentation: Harondel J. Sibble, mailto:help@pdscc.com
- Mandrake 9.0 and 9.1 packages: Thac, mailto:thac@nyvalls.se
- Red Hat Linux 9 packages: Axel Thimm, mailto:Axel.Thimm@physik.fu−berlin.de
- Alternate setup program for MythTV and listings grabber for Germany: Ben Bucksch, http://www.bucksch.com/1/projects/various/mythtv/

3. Checking prerequisites.

You must ensure that any firewalls (either hardware, or a software firewall installed by your distribution) will not block access to the ports that will be used by the MythTV clients and servers on the "inside" LAN. The ports for MySQL (TCP port 3306) and mythbackend (TCP ports 6543 and 6544) must be open. It is *strongly* recommended that you do *not* expose the MythTV and MySQL ports to the Internet or your "Outside" LAN.

3.1 Hardware.

Hardware selection is a complex topic, one this HOWTO will only discuss briefly and in general terms. The following subsections offer some general guidance but stop short of offering specific recommendations.

For more detail about actual configurations that others have used, Mark Cooper has setup a hardware database at http://pvrhw.goldfish.org/. The website will let you browse what other users have reported as their hardware configuration, and how happy they are with the results.

If you have specific questions about the suitability of specific hardware choices, you can consult the archives of the mythtv−users mailing list at http://www.gossamer−threads.com/archive/MythTV_C2/Users_F11/ or post a question to the list.

CPU Type and Speed

Selection of CPU type and speed is one of the trickiest elements of hardware selection, mainly because there are so many tradeoffs which can be made. For example, if you have plenty of CPU, you can use higher bitrates or capture sizes, etc.

MythTV has two modes of operation. First, it can function as a software video encoder, which means that it uses a fairly generic "dumb" video capture card to get frames of video, encodes them using the CPU on your motherboard and writes them to disk. High−end video capture cards and devices like the Tivo and ReplayTV have dedicated encoder chips which use specialized hardware to convert the video stream to the MPEG−2 format without using the motherboard CPU. The main CPU has the responsibility of running the Operating System and reading and writing the encoded frames to the disk. These tasks have fairly low CPU requirements compared to encoding video, which is why a device like a Series 1 Tivo can run with only 16MB of RAM and a 54MHz CPU.

There are many variables that go into the question: "How fast a CPU do I need to run MythTV"? Obviously, the faster your CPU, the better your experience will be with MythTV. If you are using the software MPEG−4 encoder and performing the "Watch TV" function, where the CPU is both encoding and decoding video simultaneously to allow Pause, Fast Forward and Rewind functions for live TV requires more CPU then just encoding or decoding. MythTV also supports multiple encoder cards in a single PC, thereby increasing the CPU requirements if you plan on simultaneously encoding multiple programs. As a general guideline, plan on 1GHz per encoder.

Here are a few data points:

- A PIII/733MHz system can encode one video stream using the MPEG−4 codec using 480x480 capture resolution. This does not allow for live TV watching, but does allow for encoding video and then watching it later.
- The developer states that his AMD1800+ system can **almost** encode two MPEG−4 video streams and watch one program simultaneously.
- A PIII/800MHz system with 512MB RAM can encode one video stream using the RTJPEG codec with 480x480 capture resolution and play it back simultaneously, thereby allowing live TV watching.
- A dual Celeron/450MHz is able to view a 480x480 MPEG−4/3300kbps file created on a different system with 30% CPU usage.
- A P4 2.4GHz machine can encode two 3300Kbps 480x480 MPEG−4 files and simultaneously serve content to a remote frontend.

The second mode of operation is where MythTV is paired with a hardware−based video encoder, such as a Matrox G200 or a Hauppauge WinTV−PVR−250/350. In this mode, because the video encoding is being done by a dedicated video processor, the host CPU requirements are quite low. See the Video Capture Device section for details.

If you have a Via M10000 series or a Hauppauge PVR–350, MythTV can use the hardware–based video decoder for playback, which further reduces CPU requirements.

Memory

A MythTV host that is both a backend and a frontend and using software encoding with a single capture card should run adequately in 256MB of RAM. Additional RAM above 256MB will not necessarily increase performance, but may be useful if you are running multiple encoders.

Hard Disk(s)

Encoded video takes up a lot of hard disk space. The exact amount depends on the encoding scheme, the size of the raw images, and the frames per second, but typical values for MythTV range from 700 megabytes/hour to 2 gigabyte/hour for MPEG–4 and larger for MPEG–2 and RTJPEG.

NOTE: You *must* use DMA for hard drive access to prevent choppy or jittery video. Red Hat Linux 8.0 apparently does not enable DMA at boot time. See the Troubleshooting Section for instructions on how to do this.

Writing video to disk is sensitive to timing issues; RTJPEG requires less CPU with the tradeoff being larger files and needing to write to the disk faster. MPEG–4 requires more CPU, but the files are smaller. At the default resolution, MPEG–2 creates the largest files of all with almost no CPU impact.

See the Troubleshooting section for more information.

Filesystems

MythTV creates large files, many in excess of 2GB. You *must* use a filesystem that will allow you to create large files. Filesystems known to have problems with large files are FAT and FAT32.

Because of the size of these files, it may be useful to plan for future expansion right from the beginning. If your case and power supply have the capability for additional hard drives, read through the LVM and Advanced Partition Formatting sections for some pointers.

Video Capture Device

In order to watch live TV, MythTV will need one or more video capture devices with Linux drivers. For a list of video capture cards known to work with Linux, please see `/usr/src/linux/Documentation/video4linux/bttv` for a partial listing; even if your specific card is not listed, it may be that the vendor is actually using a standard reference design and placing their own name on it. See the video4linux mailing list (https://listman.redhat.com/mailman/listinfo/video4linux–list) for more information and for specific hardware questions.

The most common inexpensive cards available use the Bt848 or Bt878 video capture chip; examples are the "Hauppauge WinTV Go" card and the "AverTV Desktop PVR" card, both of which use the bttv kernel module.

```
NOTE:  The ATI TV Wonder series and the ATI All-in-Wonder series of cards
are not the same.  The All-in-Wonder cards will not work with MythTV.
```

Newer cards may use the CX23881 series of chips; a driver is available at http://bytesex.org/cx88/.

After you have installed a suitable capture device, you can check that the kernel sees it with `lspci`. Look for an entry labeled "Multimedia video controller". To get more detailed information about the card, use `lspci -v` or `lspci -vv`. Ensure that your system is loading the bttv modules by typing:

```
# lsmod |grep bttv
```

You want to see the `bttv` module listed.

While inexpensive video−capture cards just capture raw frames, leaving encoding to software, some higher−end cards incorporate hardware−level encoding. Using either a G200 MJPEG encoder card, or a MPEG−2 encoder card supported by the IvyTV project http://ivtv.sourceforge.net/ such as the Hauppauge PVR−250, PVR−350, Avermedia M179, Hauppauge "Freestyle" or Yuan M600 cards will allow you to use dedicated hardware encoders rather than your CPU. Release 0.13 of MythTV is able to use these MPEG−2 encoders as an input device for live TV and for scheduled recordings. (The PVR−350 can simultaneously be used as an output device.) Using the onboard MPEG−2 encoder drastically reduces the CPU requirements for encoding.

Here are some data points for encoding:

- A Celeron 450 uses 2% CPU for encoding a 480x480 16Mbps MPEG−2 stream.

Here are some data points for decoding:

- An Athlon 1800XP can decode a 720x480 8Mbps MPEG−2 file using 10% CPU
- An Athlon 1GHz can decode a 720x480 16Mbps MPEG−2 file using 30−50% CPU, can decode a 480x480 16Mbps MPEG−2 using 30% CPU and approximately 30% for Live TV at 416x480.
- A P3−550 can decode a 480x480 16Mbps MPEG−2 file with 55% CPU.
- A Celeron 450 (no SSE) can decode a 480x480 16Mbps MPEG−2 file with 80% CPU.

See the Hauppauge PVR−250/350 hardware MPEG−2 encoder. section for information on installing a PVR−250 on Mandrake 9.1. The information contained in the section will also apply to other MPEG−2 encoder cards supported by the IVTV driver.

NOTE: The ATI All−in−Wonder cards (which are not the same as the ATI TV Wonder, TV Wonder VE or TV Wonder Pro) *will not* work as a MythTV capture device because the GATOS http://gatos.sourceforge.net drivers that are available provide only a limited subset of the V4L API. The TV Wonder series of cards are supported by the Bt8x8 Video4Linux driver.

Sound card

The system needs a sound card or an onboard equivalent on the motherboard to play back and in most cases, to record sound. Any sound card that can be operated by the ALSA (Advanced Linux Sound Architecture) kernel modules will work with MythTV. However, some cards and drivers will provide better quality or compatibility than others. In particular, many audio devices included on motherboards can be problematic.

The usual practice for capturing the audio associated with the video is to run a cable from an audio output on the video capture card to the Line input on a sound card. However, some video capture cards provide onboard

audio capabilities that work with the kernel `btaudio` module instead, thereby eliminating the need for a cable. This is useful if you will be using multiple capture cards in a single chassis, since each capture card will not need its own sound card. Note that a separate sound card is still required for playback when using `btaudio`, and that often the audio recorded in this way will be mono only. See the btaudio section for more information.

NOTE: The MPEG−2 cards supported by the IVTV driver do not require a sound card for audio capture.

NOTE: Plugging a Line−level device into the Mic input is not recommended. Line−level devices have higher volatages and can damage the soundcard. In addition, even if it doesn't break your card, you will be getting Mono sound. See the Linux MP3 HOWTO at http://www.ibiblio.org/pub/Linux/docs/HOWTO/other−formats/html_single/MP3−HOWTO.html#toc8 for additional information.

Video Display Card

MythTV will work with just about any video card. However, it is highly recommended that you use a card which supports XVideo (XV) extensions. If your card does not support XV, color conversion and scaling will be performed by your CPU rather than the video card. This is very CPU and memory intensive and will often result in dropped frames and a corresponding degradation of quality. Check the XFree86 documentation for details if you are uncertain about your preferred card. You may also run `xvinfo`; look for your video card to be listed as one of the adapters.

If you want to use MythTV with a standard television, you will need a physical connection from your video card to your TV set, which can either be a TV−out port on the card itself or an external adapter that converts the VGA signal to an appropriate video signal. "Appropriate" depends on a number of factors, such as video standard (NTSC vs PAL) type of input connection (Composite vs SVideo), etc.

Note that with some video cards and X drivers, Xvideo extensions are only supported on the VGA output, and not on the TV output.

Cards with TV out

The next section deals with a number of cards that are known to have TV−out ports. The list is unlikely to be complete, so if you know of others, please post a message to the mythtv−users mailing list so the information can be included in future versions of the HOWTO. The list is organized by manufacturer.

Reports here are based on what users of the cards have posted on the mythtv−users mailing list, so if you need configuration details, please search the archives at http://www.gossamer−threads.com/archive/MythTV_C2/Users_F11/ using the card name in your search string.

ATI

ATI makes many cards with TV−out capability, but ATI offers no official support for their use with Linux. The standard XFree86 ati driver does not support TV−out.

The enhanced ati.2 X driver created by the GATOS http://gatos.sourceforge.net project offers some support for TV−out, but only in its "experimental" version, available through CVS. There have been reports from people who say they have made this driver work with one or another ATI card. For example, Bruce Markey

mailto:bjm@lvcm.com writes (on the mythtv−users mailing list): "I got this to work. You can quote me on that. I've used TV−out on several models of ATI cards both All−In−Wonder and regular cards with TV−out." See the "Adventurous Setup" section of http://gatos.sourceforge.net/watching_tv.php for details. Also see http://www.retinalburn.net/linux/tvout.html for more information.

Specific cards:

- ATI Rage II+DVD (Mach 64). Ports not described. Reported to work, but no details reported.
- Rage 128 VIVO. Ports not described. Reported to work, but no details reported.
- Radeon 7500. Ports not described. Reported to work, but no details reported.

Matrox

The standard Matrox driver included with XFree86 does not support TV−out.

Older Matrox cards can be used with a proprietary X driver provided by Matrox, and it does simultaneously support TV out and XVideo on some cards. See http://www.matrox.com/mga/support/drivers/files/lnx_21.cfm for details.

Newer Matrox cards can be run with a set of kernel patches (for kernel 2.4.19) and a customized X driver, but this arrangement does not support XVideo on TV−out. See http://www.bglug.ca/matrox_tvout/g450_tvout_howto.html for details.

Specific cards:

- Matrox G400 AGP. Uses external adapter to provide Composite and SVideo ports. Reported to work with the proprietary Matrox X driver.
- Matrox Millennium 450GX AGP. Uses external adapter to provide Composite and SVideo ports. TV−out implementation does *NOT* support XVideo.
- Matrox Millennium 550GX AGP. Uses external adapter to provide Composite and S−Video ports. TV−out implementation does *NOT* support XVideo.

nVidia

Some nVidia cards with TV−out can be run using the standard nv driver in XFree86, combined with the userspace application **nvtv** to control the TV−out port. See http://sourceforge.net/projects/nv−tv−out/ for details. Recent versions of the nVidia driver have better support for overscan and other features useful with TV−Out, so the **nvtv** application may not be required.

Some nVidia cards can be run with a proprietary nVidia X driver made available by nVidia. See http://www.nvidia.com/view.asp?IO=linux for more information.

Specific cards:

- GeForce4 MX440−SE AGP. Has Composite and SVideo ports. Works using the nVidia X driver.
- GeForce2 GTS 64 MB DDR 4x AGP. Has SVideo port. Works using the nVidia X driver. Reportedly works using the nv driver plus nvtv.
- GeForce4 MX420 PCI (BFG Asylum). Has SVideo port.

Savage

The standard savage X driver supports TV out on some Savage cards. Unfortunately, XVideo support for the Savage 2000 card (a very common and inexpensive Savage card with TV out) is broken. See http://www.probo.com/timr/savage40.html for details.

Specific cards:

- Savage 2000 AGP. Has Composite and SVideo ports. Does not support XVideo.
- S3 Savage IX 8MB AGP. Has SVideo output. Reported to work.

Hauppauge PVR−350

MythTV supports the TV−out and MPEG−2 decoder functions in the IvyTV driver.

Other Options

Some devices with onboard TV−out capability, such as XBoxes converted to Linux and some laptops, can be used as MythTV frontends to display on a television screen. Please consult the mythtv−users mailing list for messages that report the details of these special arrangements.

External Adapters

External adapters convert standard VGA output to a form suitable for display on a television. The output format varies by region, since different countries have different TV standards. People on the mythtv−users list have mentioned these adapters:

- AITech Web Cable Plus, powered by external transformer or takes power from PS/2 keyboard connector, support resolutions up to 1024x768, outputs composite and SVideo, provides position adjustment.
- Averkey lite, powered by a USB port, has Composite, SVideo, YPbPr outputs; pan, brightness, overscan/underscan controls; supports up to 1024x768 outputs; and supports PAL and NTSC.
- ADS TV Elite XGA
- AverKey iMicro (comments are generally favorable)
- AITech Web Cable (comments are generally unfavorable, different than the "Plus" version above)
- TVIEW Gold (mentioned once, favorably)

3.2 Software.

There are a few ways of installing programs on Linux systems; you can either use a pre−compiled package, or install from a tarball after satisfying any prerequisites.

NOTE: you must have the MySQL database software installed on a system to store the master database. This does not necessarily mean that MySQL must run on one of the MythTV boxes.

NOTE: The following distributions have issues which may prevent a successful MythTV installation:

- Suse 8.2 – Broken Qt package
- Red Hat 8 – does not meet minimum version requirement for Qt (3.1)

Pre–compiled packages.

A number of people have created pre–compiled packages for MythTV that may make your installation easier.

BIG FAT WARNING: This HOWTO assumes that you have *not* installed MythTV from a package. All example command lines and file locations are based on the MythTV tarball defaults. Some packagers have modified the filenames, binaries and file locations to match what is commonly found in that distribution. Any issues with MythTV installed via a pre–compiled package **MUST** be raised with the packager.

If you use any of the pre–compiled packages you may not need to perform any additional configuration steps in this HOWTO. The next logical step is configuring MySQL, which you may or may not have to perform. See your package documentation.

Red Hat Linux

Red Hat Linux 9 packages for MythTV and most of its add–on modules and some themes have been packaged by mailto:Axel.Thimm@physik.fu–berlin.de and are available at http://atrpms.physik.fu–berlin.de/topic/multimedia/. All of the prerequisites for MythTV (such as XMLTV) are available as RPM packages. If you have problems with the RPMs, please send him email directly.

Given the large number of dependent RPMs you are advised to use tools like apt or yum for automatic retrieval and installation of the required RPMs. (http://atrpms.physik.fu–berlin.de/install.html) In this case a special meta–package called mythtv–suite will allow you to install all of MythTV and its add–ons, plus all dependencies.

If you don't have **apt** or **yum** on your machine, download and install the atrpms–kickstart package. Install the package with:

```
# rpm -Uvh atrpms-kickstart*
```

Then update your **apt** package listings:

```
# apt-get update
```

And finally:

```
# apt-get install mythtv-suite
```

These steps however, do NOT perform the installation of any drivers required for **ALSA**, capture cards, **lirc**, etc., nor do they set up your MythTV database. Check ATrpms for the drivers you need.

Further documentation on using ATrpms packages can be found in Jarod Wilson's mailto:jcw@wilsonet.com "HOWTO Build a MythTV System on Red Hat Linux 9 w/ATrpms" document, http://pvrhw.goldfish.org/tiki–page.php?pageName=rh9pvr250. Just like 3rd–party packages, any 3rd–party documentation problems should be brought up with the maintainer.

Mandrake

Thac has created RPMs for MythTV for Mandrake 9.0 and 9.1 which may be obtained from http://rpm.nyvalls.se/ If you have problems with the RPMs, please send him email directly at thac@nyvalls.se.

Debian

Debian packages for MythTV and most of its add−on modules are maintained by Matt Zimmerman
mailto:mdz@debian.org and are available at http://dijkstra.csh.rit.edu/~mdz/debian/dists/woody/mythtv/ (for
Debian 3.0/woody) and http://dijkstra.csh.rit.edu/~mdz/debian/dists/unstable/mythtv/ (for unstable). Be sure
to use the appropriate packages for your system; the woody packages cannot be installed on unstable, nor vice
versa. Installation instructions can be found on those pages as well. All of the prerequisites for MythTV are
available as Debian packages, most of them from the official Debian archive.

If you are running woody, be sure to follow the instructions on the above page to ensure that the MySQL
server starts up immediately after it is installed. If you are running unstable, this is the default behavior.

The Debian packages are configured such that MythTV programs should be run as the `mythtv` user, which is
automatically created during installation. This user has access to write new recordings to disk in the default
directory, read and write the database, access the audio and video devices, and everything else that MythTV
needs to do.

See `/usr/share/doc/packagename/README.Debian` for more information, including copies of the
MythTV documentation. The `mythtv-doc` package contains a copy of this HOWTO in
`/usr/share/doc/mythtv-doc`.

Manual installation

You may use the graphical tools that come with your distribution, or you can use command−line utilities.
Either system will get the job done, and it all depends on your comfort level with Linux.

In order to compile MythTV, we need to make sure that the software it needs is installed. As of 2003−03−15,
this list includes **mysql**, **gcc**, **freetype2−devel**, **XFree86−devel**, **qt−devel** and **lame**. If you're going to use a
remote control with MythTV, you're going to need the **cdialog** package in order to compile **lircd** if your
distribution doesn't have a pre−packaged **lirc**. If you are using **XMLTV** as a grabber, you will need **perl**.

```
NOTE:   Qt v3.1 is required.

NOTE:   If you are going to be using RPMs to install various
components, you should be aware that not all packages include the necessary
headers for compiling packages.  If you're having trouble compiling, ensure
that you've installed the -devel version of a prerequisite.

NOTE:   Other than where noted, Mandrake 9.1 users may follow the
Mandrake 9.0 instructions.
```

There appears to be issues with ALSA and simultaneously recording and playing sound, which is a core
requirement for MythTV. If you are not running ALSA 0.9.0rc6 or higher and experience lock−ups of your
system when trying to use MythTV, then please check the main ALSA website and upgrade to the latest
version, or see if your distribution has already packaged the latest ALSA drivers.

Graphical installation tools.

Mandrake 9.0

On systems running KDE, a tool which will allow you to see what packages are available and if they're installed on your system can be accessed by clicking K−>Configuration−>Packaging−>KPackage

While the graphical tools are nice, the command line tool is actually easier to use. It's strongly recommended that you use the command line tool.

There are many inter−related modules, so make sure that you've got the main package for each one, and not a sub package for an unrelated item. For example, there are many modules that have "perl" in them; some are a part of the apache web server, others are a part of the perl distribution. In each case, we're looking to make sure that the primary module is getting installed.

Click the magnifying glass or press **CTRL−F** and enter "perl" as your search term. You should find it under the "Perl" folder. You need at least the perl 5.80 base package installed. If not, click on the "Mark" button.

If you are going to be installing mysql on this system, search for mysql − it should be under "Databases". Make sure that mysql and mysql−clients are installed. If not, click on "Mark".

Search for gcc. You're going to need gcc and gcc−c++ installed.

Search for freetype2. Make sure that you've got freetype2 and freetype2−devel installed.

Search for XFree86−devel.

Finally, check for qt. You'll need libqt3−devel (under folder "KDE and Qt") and libqt3 ("Libraries").

```
NOTE for Mandrake 9.1 users:  Mandrake 9.1 has removed the MySQL modules
from the default installation.  Ensure that you install libqt3-mysql.
```

Make sure that cdialog is installed if you want to use MythTV with a remote control if you need to compile **lirc** from source.

alsa−utils is needed for setting the volume. MythTV now has native volume support, but you may have a special configuration that necessitates an external program to adjust your volume settings. If this is the case, then you can disable native volume support.

If you marked anything to install, click on Install Marked. Click "Yes" or OK if you get a message about satisfying dependencies.

Red Hat Linux 8.0

Red Hat Linux 8.0 does not have the same graphical package manager used by other distributions, so you will need to use Red Hat's package manager. This is the same utility you saw upon installation from the Red Hat Linux 8.0 installation CD, and it allows you to select packages at installation time, or once the system is up and running. You can access this program by selecting

```
RedHatMenu>System Settings>Packages
```

If you did a typical Desktop install of Red Hat Linux 8.0 make sure you have selected:

- KDE Desktop Environment
- SQL Database Server – you will have to click on "details" and make sure "mysql–server" is checked. (only required on the master system)
- Development Tools
- X Software Development
- KDE Software Development
- Kernel Development

Click Install to continue.

You can not install qt–MySQL, a necessary prerequisite, graphically. To install this package using the command line do

```
$ up2date --solvedeps qt-MySQL
```

NOTE: ensure that the correct symlinks have been created for the XFree–86 development libraries:

```
# ln -s /usr/X11R6/lib/libXext.so.6 /usr/X11R6/lib/libXext.so
# ln -s /usr/X11R6/lib/libX11.so.6 /usr/X11R6/lib/libX11.so
```

Command–line installation.

This section details the various methods for installing prerequisites from the command line.

Mandrake

urpmi is the simplest tool for installation of packages from the command line, but properly configuring it can be difficult. The following website http://plf.zarb.org/~nanardon/urpmiweb.php will allow you to choose a mirror site and then present the command–line configuration text for that mirror. You will most likely need to add a "Contrib" mirror to your setup. If you add a site from the "Penguin Liberation Front", you will be able to load the `lame` library without compiling from source.

Special instructions for Mandrake 8.2

NOTE: These instructions will be removed on 2004–01–15. Contact the HOWTO maintainer if you wish to create up–to–date docs for Mandrake 8.2.

Mandrake 8.2 does not have a pre–packaged libqt3–devel, so you must compile and install this manually. The following command will install the prerequisites for qt3, so don't skip this step. The simplest tool for installing packages from the command line is urpmi.

Open a shell, and execute the following. You may get asked a number of questions regarding dependencies. It's best to answer "YES".

```
$ urpmi mesa-common-devel libmng1-devel mysql-devel libunixodbc2-devel
$ urpmi postgresql-devel xfree86-static-libs
```

Once all of the prerequisites have been installed, the next step is to obtain and compile the source RPM.

Go to http://www.rpmfind.net and search for `qt3`. There will be a number of different versions available. As of 2002–12–02, the latest version for Mandrake is called `qt3-3.0.5-7mdk.src.rpm`. Unless you are

running **Mandrake Cooker** (the beta version of the next release of Mandrake), make sure to download the standard "Mandrake" release. Download the `.src.rpm` file to a directory that you can find.

```
$ su
# cp qt3-3.0.5-7mdk.src.rpm /usr/src/RPMS/SRPMS
# cd /usr/src/RPMS/SRPMS
# rpm --rebuild qt3-3.0.5-7mdk.src.rpm
# cd ../RPMS
# ls -l
```

The "ls –l" command will show you a directory listing. There should be two files, one will be named "qt3" and the other will be "qt3–devel". The full filenames will vary, so one you know the filename, you can install them with

```
# rpm -Uvh [filename1]
# rpm -Uvh [filename2]
```

Once this has completed (it can take a while, so please be patient), continue by following the Mandrake 9.0 instructions.

Mandrake 8.2 and 9.0

The simplest tool for installing packages from the command line is **urpmi**.

Open a shell, and execute the following, all on the same line. You may get asked a number of questions regarding dependencies. It's best to answer "YES".

```
$ su
# urpmi mysql gcc gcc-c++ freetype2-devel cdialog alsa-utils
# urpmi XFree86-devel perl

# urpmi libqt3-devel
```

```
NOTE for Mandrake 9.1 users: execute the following command.

# urpmi libqt3-mysql
```

However, you might get this when you execute the command above:

```
[root@pvr root]# urpmi mysql gcc gcc-c++ freetype2-devel libqt3-devel alsa-utils XFree86-de
everything already installed
```

In that case, you're ready to move to the next section.

Gentoo.

If Qt has not been installed on your system: Edit `/etc/make.conf` and locate the "USE" variable. If the line is commented out, remove the comment. The line should have at least:

```
USE="mysql alsa"
```

Next you need to build Qt. If you don't plan on using the ebuilds as described in the Gentoo section then you also need to install lame.

```
# emerge lame mysql qt
```

If you have already installed Qt: you will need to rebuild because the default installation doesn't include mysql support, a requirement for MythTV. To enable SQL support, add "mysql" to your USE variable in `/etc/make.conf` and rebuild Qt by running

```
# emerge qt
```

All the necessary files will be downloaded and built. Even on a fast machine this may take a lot of time if you need to do a full Qt build.

4. System Configuration Requirements for Compiling MythTV

Before you compile MythTV from the current source tarball or from CVS, you may need to modify your system configuration in a few ways.

In general, if you install MythTV from pre−packaged binaries for your Linux distribution/version, you don't need to be too concerned about the issues in this section of the HOWTO – the install script for the packages should take care of them. However, this section is still recommended reading which may help if the packager skipped a step in their packaging.

4.1 Software requirements for compiling MythTV

General requirements

MythTV is written in C++ and requires a fairly complete, but standard, compilation environment, including a recent g++ compiler, `make`, and appropriate headers files for shared libraries. Any standard Linux distribution should be able to install a suitable compilation environment from its packaging system. Section 3.2 of this HOWTO provides some details of how to install the required environment for many distributions.

Subsequent sections of this chapter address the few oddities that you may have to adjust by hand before you compile MythTV.

The reference compilation system for MythTV is Debian.

Distribution−Specific Notes

None

4.2 Shared−Library Requirements for MythTV

Modifying /etc/ld.so.conf

The runtime manager for shared libraries, **/lib/ld.so**, gets information about the locations and contents of shared libraries from `/etc/ld.so.cache`, a file created by **ldconfig** from information in `/etc/ld.so.conf`. Because MythTV installs some shared libraries in `/usr/local/lib`, that directory needs to be added to the list of directories for **ld.so** to search when doing runtime linking of programs. You do

this, as root, by editing /etc/ld.so.conf, then running **ldconfig**. There are many ways to do this; one that works is to enter this series of commands:

```
$ su -
# echo /usr/local/lib >> /etc/ld.so.conf
# /sbin/ldconfig
# exit
$
```

Distribution–Specific Notes

Red Hat

It appears that Red Hat Linux 8.0 requires that the **ldconfig** command be run twice. Red Hat will also frequently "forget" about the path. It's recommended to run **/sbin/ldconfig** after installing every package. Failing that, always run it if you get an unexpected error about not finding libraries.

4.3 Environment variable Requirements for MythTV

General requirements

QT libraries and binaries

The compiler needs to be able to locate QT binaries and libraries in order to compile MythTV. QTDIR needs to be set and /usr/lib/qt3/bin must be added to your path. Your distribution may already be making these changes as a part of the installation of the software prerequisites detailed earlier.

One way to do this is as follows:

Open a shell and execute the following:

```
$ echo $PATH
/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/usr/games:/usr/lib/qt3/bin:/home/mythtv/bin:/
$ echo $QTDIR
/usr/lib/qt3
```

You should see a value like /usr/lib/qt3 for QTDIR and /usr/lib/qt3/bin should be in $PATH.

If you don't, do not proceed past this step until you have resolved this error. You may need to manually specify the QTDIR and PATH at the shell prompt before compiling.

Also, check that there has been a link created in /usr/lib/qt3/mkspecs called default. If not, you'll get errors during the compile. See the Troubleshooting Section for more information.

Distribution–Specific Notes

Mandrake

The following instructions work for Mandrake 9.0 and 9.1 using **bash** as the shell, and may be applicable for a distribution which uses /etc/profile.d.

As root, create the following file in `/etc/profile.d` The example filename is "mythtv.sh". Use what you feel is appropriate.

Open a shell, and switch to superuser mode. **NOTE**: ^D means press CTRL and d at the same time.

```
$ su
# cd /etc/profile.d
cat > mythtv.sh
export QTDIR=/usr/lib/qt3
export PATH=$PATH:/usr/lib/qt3/bin
^D

# chmod a+x mythtv.sh
# exit
$ exit
```

The last two commands are to exit out of the shell. This way, when you next open a shell your new commands will take effect.

Red Hat Linux 9/8.0

Red Hat Linux 9/8.0 use a default locale with UTF−8 encoding. Date::Manip, a perl module used by XMLTV, is not UTF−8 safe. As a result, whenever you run setup or mythfilldatabase you will get a lot of UTF−8 warnings. There is an http://membled.com/work/patches/DateManip/ updated version of Date::Manip at the XMLTV site fixing this and other XMLTV related # bugs.

Debian

Debian source packages are available from the same place as the binary packages. These can be used to build customized packages, and to automatically satisfy MythTV's build−time dependencies.

```
sh-2.05b# apt-get build-dep mythtv
Reading Package Lists...
Building Dependency Tree...
The following NEW packages will be installed:
  debconf-utils debhelper file fontconfig gettext html2text intltool-debian
  libaudio2 libexpat1 libfontconfig1 libfontconfig1-dev libfreetype6
  libfreetype6-dev libglib2.0-0 libjpeg62 libjpeg62-dev liblame-dev liblame0
  liblcms1 liblcms1-dev libmagic1 libmng-dev libmng1 libmysqlclient-dev
  libmysqlclient12 libogg-dev libogg0 libpng12-0 libpng12-dev libqt3-headers
  libqt3-mt-dev libqt3c102-mt libvorbis-dev libvorbis0a libvorbisenc2
  libvorbisfile3 libxcursor-dev libxcursor1 libxft2 libxft2-dev libxrender-dev
  libxrender1 mysql-common pkg-config po-debconf qt3-dev-tools xfree86-common
  xlibmesa-gl-dev xlibmesa-glu-dev xlibmesa3-gl xlibmesa3-glu xlibs xlibs-dev
  zlib1g-dev
0 upgraded, 54 newly installed, 0 to remove and 0 not upgraded.
Need to get 0B/24.0MB of archives.
After unpacking 78.7MB of additional disk space will be used.
Do you want to continue? [Y/n]
[...]
sh-2.05b# cd /tmp
sh-2.05b# apt-get source mythtv
Reading Package Lists...
Building Dependency Tree...
Need to get 3085kB of source archives.
Get:1 http://dijkstra.csh.rit.edu unstable/mythtv mythtv 0.11-4 (dsc) [507B]
Get:2 http://dijkstra.csh.rit.edu unstable/mythtv mythtv 0.11-4 (tar) [3071kB]
```

```
Get:3 http://dijkstra.csh.rit.edu unstable/mythtv mythtv 0.11-4 (diff) [13.9kB]
Fetched 3085kB in 0s (8120kB/s)
dpkg-source: extracting mythtv in mythtv-0.11
sh-2.05b# cd mythtv-0.11
[customize settings.pro, etc.]
[edit debian/changelog and increment the version number]
sh-2.05b# dpkg-buildpackage -rfakeroot -us -uc -b
[...]
dpkg-deb: building package `mythtv-backend' in `../mythtv-backend_0.11-4_i386.deb'.
dpkg-deb: building package `mythtv-frontend' in `../mythtv-frontend_0.11-4_i386.deb'.
dpkg-deb: building package `libmyth-0.11' in `../libmyth-0.11_0.11-4_i386.deb'.
dpkg-deb: building package `libmyth-0.11-dev' in `../libmyth-0.11-dev_0.11-4_i386.deb'.
 dpkg-genchanges -b
dpkg-genchanges: binary-only upload - not including any source code
dpkg-buildpackage: binary only upload (no source included)
sh-2.05b# ls -l ../*.deb
-rw-r--r--    1 root     root        31734 Sep  6 19:07 ../libmyth-0.11-dev_0.11-4_i386.deb
-rw-r--r--    1 root     root       408444 Sep  6 19:07 ../libmyth-0.11_0.11-4_i386.deb
-rw-r--r--    1 root     root      3324966 Sep  6 19:07 ../mythtv-backend_0.11-4_i386.deb
-rw-r--r--    1 root     root        15350 Sep  6 19:07 ../mythtv-common_0.11-4_all.deb
-rw-r--r--    1 root     root        17978 Sep  6 19:07 ../mythtv-database_0.11-4_all.deb
-rw-r--r--    1 root     root       422232 Sep  6 19:07 ../mythtv-doc_0.11-4_all.deb
-rw-r--r--    1 root     root      4900474 Sep  6 19:07 ../mythtv-frontend_0.11-4_i386.deb
-rw-r--r--    1 root     root        12372 Sep  6 19:07 ../mythtv_0.11-4_all.deb
```

If building from CVS instead of the source package, then the QTDIR environment variable must be set when building. On Debian woody, QTDIR should be set to `/usr/share/qt`. On Debian unstable, QTDIR should be set to `/usr/share/qt3`.

5. Downloading and compiling.

Get MythTV from the mythtv.org web site. There are two installation methods you may choose from. The first is to download the latest release in tarball format and compile. The tarball release of MythTV should work on a wide variety of systems and should be the preferred method for new users. If you wish to use the CVS version of MythTV you may obtain it from CVS this way:

```
$ mkdir mythtv
$ cd mythtv
$ cvs -d :pserver:mythtv@cvs.mythtv.org:/var/lib/mythcvs login
Logging in to :pserver:mythtv@cvs.mythtv.org:/var/lib/mythcvs
CVS password: mythtv
$ cvs -z3 -d :pserver:mythtv@cvs.mythtv.org:/var/lib/mythcvs checkout mythtv
```

The other MythTV modules may be downloaded from CVS the same way; replace "mythtv" with "mythweather", "mythgallery", "mythmusic", etc.

If you wish to stay up–to–date with CVS, you may replace "checkout mythtv" with `update mythtv` after you've done your initial download.

NOTE: If you are going to use CVS to compile MythTV rather than using the distribution tarball, it is **strongly** recommended that you join the http://www.mythtv.org/mailman/listinfo/mythtv–commits/ and http://www.mythtv.org/mailman/listinfo/mythtv–dev/ mailing lists.

Get XMLTV from http://xmltv.sourceforge.net. Download version 0.5.25 if you're in North America.

```
NOTE for Mandrake users:  If you have added a "PLF" mirror, you may skip the
```

```
next step and type:

# urpmi libmp3lame0 libmp3lame0-devel

NOTE for Red Hat users: If you are installing with apt or yum, you may skip this step, it will be
of. Otherwise get lame and lame-devel from FreshRPMS: http://freshrpms.net/rpm/lame/

After downloading, be sure to install both:
# rpm -Uvh lame*
```

Get lame from http://www.mp3dev.org/mp3. Download the source code to v3.93.1 by following the links to sourceforge.net

5.1 Notes on compiling on a system with non–Intel processors.

If you've got a system with an AMD K6–2 or a VIA C3 processor, make the following change in the `settings.pro` file.

Look for

```
-march=pentiumpro
```

and replace it with

```
-march=i586
```

before you compile.

Recent AMD processors, like the Duron, Thunderbird and Athlon don't need this change.

5.2 Building LAME.

Open a shell and switch to the directory where you saved lame.

```
        $ tar -xzf lame-3.93.1.tar.gz
        $ cd lame-3.93.1
        $ ./configure
        $ make
        $ make test
        $ su
        # make install
```

Check that it worked:

```
        # ls -l /usr/local/lib
        -rw-r--r--    1 root     root          381706 Nov  4 14:22 libmp3lame.a
        -rwxr-xr-x    1 root     root             674 Nov  4 14:22 libmp3lame.la*
        lrwxrwxrwx    1 root     root              19 Nov  4 14:22 libmp3lame.so ->
        libmp3lame.so.0.0.0*
        lrwxrwxrwx    1 root     root              19 Nov  4 14:22 libmp3lame.so.0 ->
        libmp3lame.so.0.0.0*
        -rwxr-xr-x    1 root     root          360197 Nov  4 14:22
        libmp3lame.so.0.0.0*
```

```
# exit
$
```

5.3 XMLTV.

Red Hat Linux 9:

RPMs for **XMLTV** and all of its dependencies can be obtained from
http://atrpms.physik.fu–berlin.de/name/xmltv/. The web page has a list of all the dependent packages you
must download and install.

```
# rpm -Uvh xmltv* perl*
```

If you install from this location you may skip to Manually building MythTV.

Mandrake

RPMs for **XMLTV** and all of its dependencies are located in Mandrake's "contrib". If you have added a
contrib mirror, try installing **XMLTV**:

```
# urpmi xmltv
```

If this does not work, it is possible that contrib for your Mandrake version does not have **XMLTV**, so you
may install the XMLTV prerequisites by typing:

```
# urpmi perl-xml-twig perl-xml-writer perl-datemanip perl-libwww-perl
```

and skipping straight to the XMLTV compilation step.

Other distributions and manual installation.

NOTE: Red Hat Linux 8.0 and SuSE 8.1 users, you *must* download and install an updated version of libexpat.
The version that ships with Red Hat Linux 8.0 and SuSE 8.1 (version 1.95.4) is buggy, and will cause
problems with XML::Twig. All other distributions may skip this step and proceed with untar'ing the xmltv
file.

If you get errors installing XML::Twig or other XMLTV prerequisites, try deleting the old versions of
libexpat

```
$ su
# rm -f /usr/lib/libexpat*
# /sbin/ldconfig
# exit
```

Download version 1.95.5 from sourceforge.net: http://sourceforge.net/projects/expat/

Open a shell prompt:

```
$ tar -xzf expat-1.95.5.tar.gz
$ cd expat-1.95.5
$ ./configure
```

```
$ make
$ su
# make install
# /sbin/ldconfig
# exit
$
```

Continuing on:

Untar the xmltv file:

```
$ tar -xjf xmltv-0.5.25.tar.bz2
```

Install the xmltv prerequisites:

```
$ su
# perl -MCPAN -e shell
cpan> install XML::Twig
cpan> install Date::Manip
Date::Manip is up to date.
cpan> install LWP
cpan> install XML::Writer
cpan> exit
```

Change to the XMLTV directory and compile it:

```
$ cd xmltv-0.5.25
$ perl Makefile.PL
```

You can answer "N" to the tv_check, tv_pick_cgi questions. Say "yes" to the grabber required for your location.

You may get errors about modules not being installed. They shouldn't matter if you're installing North America and have followed the instructions to this point.

```
$ make
$ make test
$ su
# make install
# exit
```

5.4 Manually building MythTV.

Unpack MythTV

```
$ tar -xjf mythtv-0.13.tar.bz2
$ cd mythtv-0.13
$ ./configure
$ qmake mythtv.pro
$ make
```

```
NOTE for Red Hat 8 users: You must manually modify a Makefile
within the themes directory, otherwise your on screen display (OSD) will be
"generic" rather than themed.  The solution is to edit the Makefile within
```

```
the themes directory and comment out (using a # symbol before each line)
every line that starts with the word "strip". Make sure you switch back to
the MythTV build directory before executing the next step.
```

Switch to superuser:

```
$ su
# make install
# exit
```

NOTE: subsequent configuration steps assume that you are within the MythTV directory that you `cd`'d to above.

Frontend–only configuration

Since MythTV uses a client/server architecture, multiple frontend computers can simultaneously access content on a Myth system. Live TV, watching and scheduling recordings, etc. are all possible from multiple frontends.

To get a better picture of what is needed to run a frontend, note the following:
 • You do NOT need the MySQL server installed on your remote frontend
 • You do NOT need XMLTV installed on your remote frontend
 • You do NOT need to run the MythTV setup program on your frontend machine

Other than the exclusion of the MySQL server and XMLTV, the MythTV compilation procedure is the same as when you're setting up both a backend and a frontend. However, you *will* need to install the database access libraries.

Once MythTV is compiled and installed, you must change your `/usr/local/share/mythtv/mysql.txt` or `~/.mythtv/mysql.txt` file's "DBHostName" field to point to your master backend's IP address.

Run the MythTV setup program on your master backend. Under the "General" menu, change the IP address of the current machine (by default, "127.0.0.1") to the real external IP address – 127.0.0.1 is the loopback address and no external machine can access it. Change the Master Server IP setting to the same IP address as well.

5.5 Gentoo.

Installation of MythTV on Gentoo consists of simply emerging the desired ebuild because all of the packages are now part of the official Portage tree.

```
$ su -
# emerge --sync # make sure portage is up to date.
# vi /etc/make.conf
```

Add mysql to your USE variable. i.e. `USE="mysql ...."`

```
# emerge mythtv
```

6. MySQL.

6.1 Mandrake 9.0

If this is the system maintaining the database, make sure that mysql is running and started at boot. Click on Mandrake Control Center–>System–>Services, find mysql and click the "On Boot" button and the "Start" button if the mysql status shows that it isn't running yet.

NOTE: There have been reports that mysql isn't starting at boot.

```
$ su
# chkconfig --level 35 mysql on
# /etc/rc.d/init.d/mysql start
# exit
```

6.2 Red Hat Linux

If this is the system maintaining the database, make sure that mysql is running and started at boot. Click on Redhat menu>Server Settings>Services and enter the root password when asked. Check "mysqld" and then click Start. Click Save, then close the window.

This can be done from the command line by typing:

```
# /sbin/chkconfig mysqld on
# /sbin/service mysqld start
```

6.3 Gentoo

After installing mysql you need to initialize the database by running mysql_install_db as root.

6.4 Setting up the initial database.

This step is only required on the system maintaining the database, which may or may not be one of your MythTV boxes. If the database is on a non–MythTV machine you'll need to copy the `database/mc.sql` file to it.

To setup the initial mysql databases:

```
$ cd database
```

Mandrake 9.0 and Red Hat Linux

```
$ mysql -u root < mc.sql
```

Debian 3.0

```
$ mysql < mc.sql
```

Gentoo

```
$ su
# mysql < /usr/share/mythtv/database/mc.sql
```

NOTE: It is good practice to set a root password for mysql. Instructions for doing so can be found on MySQL's web site at http://www.mysql.com/doc/en/Security.html.

Modifying access to the MySQL database for multiple systems

If you're going to have multiple systems accessing a master database, you must grant access to the database from remote systems. By default, the mc.sql script is only granting access to the local host.

To allow other hosts access to your master database, you can either set it up for no security at all, or with more granularity. Note that the "%" is the wildcard character in MySQL.

NOTE: The "no security" option is *very* dangerous unless you're in a controlled environment.

This example has no security at all, and allows access from any host.

```
$ mysql −u root mythconverg
mysql> grant all on mythconverg.* to mythtv@"%" identified by "mythtv";
```

For a more secure setup, you can restrict which machines or subnets have access. If you have a complete DNS system operational, you could do the following:

```
$ mysql −u root mythconverg
mysql> grant all on mythconverg.* to mythtv@"%.mydomain.com" identified by "mythtv";
```

Finally, if you just want to restrict by IP subnet:

```
$ mysql −u root mythconverg
mysql> grant all on mythconverg.* to mythtv@"192.168.1.%" identified by "mythtv";
```

7. Setting up ALSA and the mixer.

If your video doesn't appear to be in−sync with your audio, it could be because you are listening to the real−time audio from your video card rather than after it's been processed and synchronized to the video by MythTV. Because MythTV is a personal video recorder, "Live TV" isn't really live – to let you pause live TV, MythTV is actually encoding the video, saving to disk, and then playing it back. This procedure puts your MythTV "live" TV about 2 seconds behind real−time, so it's important that you're not listening to the live audio. However, if you're having an issue where the audio and video aren't synchronized by small but varying amount, it's most likely because the sound driver that you're using doesn't have the DSP_CAP_REALTIME capability. This was the case with ALSA (0.5), but not with newer versions (0.9). See the Troubleshooting Audio section for more information if you're having issues with sound. Also, ensure that no other programs are grabbing the audio output, like **arts** or **esd**.

What you need to do is to mute the "line−in" of your sound card and also set it as the recording source.

There are two ways to do this. Graphically, and from the command line.

7.1 Graphically setting up the mixer.

Mandrake 9.0 and Red Hat Linux

Open Kmix by clicking K–>Multimedia–>Sound–>Kmix for Mandrake, or

```
RedHat Menu>Sound & Video>Volume Control
```

on Red Hat.

Click on Settings–>Configure Make sure that "Tick Marks" and "Show labels" have "X"'s in them. This will make it easier to find the correct audio source. Click OK.

On the mixer page, look for Line–In on your sound card. You should see two LED's – a green one at the top, and a red one at the bottom. The green one at the top is for muting; you want to make sure that the green LED is a dark green, meaning that it's "off". You also want to click on the red LED so that it turns bright red, indicating that it's "ON"; this insures that the Line–in is used as the source. Click OK, and make sure that you save the settings so that this is your default.

Using OSS drivers.

Red Hat Linux 8.0 ships with OSS sound drivers rather than the ALSA drivers recommended by the MythTV team. The OSS drivers do work for many people, and for many cards. We recommend that you give OSS a try since it is already included in Red Hat 8. However, some people report problems (sync and jitter) with OSS, and in addition it does not support full duplex on some cards. Full duplex is required if you want to record and play sound using just one sound card. ALSA has good full duplex support, and also has drivers for some built–in (motherboard) sound cards that OSS does not. Installing ALSA is less painful than you might think, so if OSS does not work for you, installing ALSA is not too difficult. For those of you who do not wish to install ALSA you must do this:

To configure sound to work with MythTV, select

```
RedHat Menu>Sound &
Video>Volume Control
```

to open up a mixer. Make sure your global volume (on the far left) is up. Also make sure that the "line in" section has "mute" and "record" checked and that the gain is turned up. You may have to experiment with volume/gain levels to get the best sound.

Using ALSA drivers.

For those of you who do want to install ALSA please follow the directions below, compiled with much assistance from Gregorio Gervasio.

First you must find out the name of your driver, which isn't necessarily the same as the name of your sound card. Go to http://www.alsa–project.org/alsa–doc/, find your sound card on the list and click "details". Near the top it will say something like "The module options for snd–XXX", where XXX is the name of your driver. In the instructions below, replace "XXX" with the name of your driver.

Download the development release (0.9.x) from the ALSA web site http://www.alsa−project.org. You need at least three files:

- alsa−driver−0.9.x.tar.bz2
- alsa−lib−0.9.x.tar.bz2
- alsa−utils−0.9.x.tar.bz2

Start with the alsa−driver package:

```
$ tar -xjf alsa-driver-0.9.x.tar.bz2
$ cd alsa-driver-0.9.x
$ ./configure --with-cards=XXX --with-sequencer=yes
```

NOTE: Red Hat users, you may need to perform the following step to allow the compile to continue:

```
mkdir -p include/linux
touch include/linux/workqueue.h
```

Continuing on:

```
$ make
$ su
# make install
# ./snddevices
# exit
$
```

This will compile and install the kernel modules and initialization scripts, as well as create device nodes in /dev.

Install the rest of the packages:

```
$ cd ~
$ tar -xjf alsa-lib-0.9.x.tar.bz2
$ cd alsa-lib-0.9.x
$ ./configure
$ make
$ su
# make install
# exit
$ cd ..
$ tar -xjf alsa-utils-0.9.x.tar.bz2
$ cd alsa-utils-0.9.x
$ ./configure
$ make
$ su
# make install
```

Next, you must edit /etc/modules.conf so that the ALSA modules are loaded automatically. The first step is to remove your old OSS configuration to ensure that you don't end up with mysterious conflicts in the future. You should end up with a /etc/modules.conf file that has this configuration:

```
alias char-major-116 snd
alias snd-card-0 snd-XXX
# module options should go here
# OSS/Free portion
```

```
alias char-major-14 soundcore
alias sound-slot-0 snd-card-0
# card #1
alias sound-service-0-0 snd-mixer-oss
alias sound-service-0-1 snd-seq-oss
alias sound-service-0-3 snd-pcm-oss
alias sound-service-0-8 snd-seq-oss
alias sound-service-0-12 snd-pcm-oss
```

"XXX" should be the ALSA driver name for your sound card/chip. In some rare cases you need to supply extra options to the driver. These options are descried in the "INSTALL" file in the alsa−driver directory or at http://www.alsa−project.org/alsa−doc/.

For now we will assume that no extra options are required. The next step is to ensure that the ALSA drivers are started during boot−up.

```
$ su
# cd /etc/rc.d/init.d
# /sbin/chkconfig --add alsasound
```

The next set of instructions will cause the ALSA volume/mixer/etc. state to be saved when the machine is shut down properly. Note that the mixer settings won't get restored the next time you boot unless you ran the shutdown script or you've explicitly saved the state which you can do as root with:

```
# alsactl -f /etc/asound.state store
```

Reboot your machine. Login and adjust the sound levels using "alsamixer". You should increase the main volume (the first slider) and the PCM volume, and unmute them by hitting "m". For MythTV, if your tuner card audio output is patched to the Line−In input of your sound card, set the Line input for capture by pressing SPACE at the slider and mute it by pressing "M". If you've patched your tuner card to the Mic−In port of your sound card, follow the same instructions but make sure that you're on the Mic slider.

Test the driver by playing a simple file:

```
$ aplay /usr/share/sounds/info.wav
```

If it works, you're done installing ALSA!

7.2 Setting the mixer from the command line.

If you have installed the alsa−utils package, then the **amixer** program can be used to setup the mixer. The "Master" volume setting is only required on a frontend machine to ensure that the sound channels are unmuted and configured for outputting sound. The "Line" and "Capture" controls are required for your sound card to actually capture audio from the external Line−in. Not all sound cards have a "Capture" control, but if yours does and you don't set it then MythTV will not capture audio.

```
Note the spelling in the following commands.
```

```
$ amixer set Master,0 100%,100% unmute
$ amixer set Line,0 75%,75% mute captur
$ amixer set Capture,0 100%,100% mute captur
$ su
# alsactl store
# exit
```

```
$
```

If you have multiple sound cards, then use the `-c` parameter to specify which card to adjust. Note that the first card will be "0", the second will be "1", etc.

That takes care of setting the volume correctly, and the ALSA startup script will restore the volume after a reboot. If you find that your sound is distorted, it's possible that the levels in the above examples are too high for your particular hardware combination. Try reducing the percentages by 5–10% and checking again. Once you're satisfied, re–run the `alsactl store` command.

You may also use the **alsamixer** program to set the volume. First, start **alsamixer** from the command line. You should start out on the "Master" volume control slider. Use the up and down cursor to set the master volume to around 75%. Next, use the left and right cursor keys to move around on the screen until you find the "Line" slider. Press SPACE to set it as the capture source, set the level to around 50–75% and press "M" to mute it. You can now press ESC to exit out of the **alsamixer** program. You can also have MythTV manage all volume and mute settings, but this will only affect the "Master" or PCM volume, not the capture volume. See the mythfrontend setup page for options.

8. Setting up a remote control.

MythTV does not have native remote control receiver and decoder software built–in. Instead, remote control functions are implemented by cooperating with **lirc**, the Linux Infrared Remote Control program. **lirc** handles the IR hardware and passes keystrokes to MythTV, which then acts as if the user had pressed the keys on the keyboard. The file `keys.txt` describes the keys used to control MythTV.

NOTE:If you are running Mandrake 9.1, you may install **lirc** by executing: # `urpmi lirc lirc-remotes` and bypass the manual compilation steps described below by jumping to the Completing the lirc install section. See the contrib/mandrake91.etc.sysconfig.lircd file for an example of how to configure lircd.

NOTE: If you wish to use the remote control that comes with the Hauppauge PVR–250/350 cards, you will need to run the lirc 0.7.0pre2 code. See the example installation in the "Hauppauge PVR–250 remote and MythTV's native LIRC support" section.

Some IR devices require a kernel recompile, and some don't. However, all at least require having the kernel source available as a resource for the lirc build process.

8.1 Gentoo

To install lirc on Gentoo, all you need to do is:

```
# emerge lirc
```

8.2 Mandrake 9.0

lircd has two prerequisites: **dialog**, available in Mandrake through the "cdialog" RPM, and the kernel source. Mandrake 9.0 is using the 2.4.19 kernel, so either install the kernel sources from the installation CD or go to http://www.kernel.org and download the full 2.4.19 tar file from http://www.kernel.org/pub/linux/kernel/v2.4/.

Obtaining the kernel.

Install using urpmi.

```
$ su
# urpmi kernel-source
```

8.3 Red Hat Linux

You should already have the source from the "Kernel Development" package you installed in section 3.2. Ensure that a symlink has been created:

```
$ cd /usr/src
$ su
# ln -s linux-2.4 /usr/src/linux
# exit
```

8.4 Obtaining and compiling lirc.

You're going to need to download and compile **lircd**. Go to http://www.lirc.org and download lirc; as of 2002−11−07, the version available is 0.6.6. Grab the remotes.tgz file as well.

```
$ tar -xjf lirc-0.6.6.tar.bz2
$ cd lirc-0.6.6
$ ./setup.sh
```

You're going to need to know what sort of receiver you have and where it's connected. In the case of the Pinnacle Studio TV card, with the IR receiver connected to COM1 (/dev/ttys0), once the configuration menu comes up, perform the configuration by going to Driver Configuration−>Other Serial Port Devices−>Pinnacle Systems Receiver−>OK and on the next page select COM1−>OK.

Each remote is different; some remote receivers connect directly to your capture card and not to a serial port, so make sure that you've got the correct one.

You then click "Save Configuration and run configure" to continue.

Make sure you read the last text generated by the configure step. It will tell you if you require a kernel recompile, and what the name of your kernel module will be (if necessary). For instance a home−built receiver may require a kernel recompile, so you would be notified that you will have to load the lirc_serial module. If you did not get any such messages skip the kernel recompile steps below and go directly to making and installing the lirc driver.

Once the configuration step is complete:

```
$ make
$ su
# make install
# chmod 666 /dev/lircd
```

At this point, if you're using a serial receiver, check that there's a `lirc` device in `/dev`:

```
$ ls -l /dev/li*
```

Obtaining the kernel. 31

```
lr-xr-xr-x    1 root     root                 5 Jan 27 09:00 /dev/lirc -> ttyS0
srw-rw-rw-    1 root     root                 0 Jan 27 15:01 /dev/lircd=
prw-r--r--    1 root     root                 0 Jan 27 09:00 /dev/lircm|
```

As you can see, there's a link from /dev/lirc to ttyS0, a.k.a. "COM1", which is appropriate for the Pinnacle Systems PCTV Pro. However, you may notice something like this:

```
crw-------    1 root     root      61,   0 Dec 31  1969 lirc
```

Some IR receivers (including some homebrew units) use a character device as their data interface as opposed to a link to a serial port. If the `make install` step has created a character device for you, don't replace it with a link to a COM port.

So, if the link or character device was not created (but should have been), ensure that you ran the `make install` step as root. If it still doesn't work, then there are three options. The first option is to re−read the **lirc** documentation to determine whether your IR receiver is a character device or should be a link to a serial port and to create the link/character device manually. In this example, the IR device is connected to ttyS0. If it were connected to "COM2", then use ttyS1, etc.

```
$ su
# cd /dev
# ln -sf ttyS0 lirc
# exit
$
```

NOTE: The above example assumes that your receiver uses the standard serial driver. Some receivers do not, including receivers that plug into a TV capture card. Check the lirc documentation, but it may be necessary to replace the link created above with a a character pipe:

```
# mknod /dev/lirc c 61 0
```

See the lirc documentation for additional information. The lirc installation *should* create this for you, so manually creating it indicates that your lirc installation may have other issues.

The second option is to post your issue to the *lirc* list, not the mythtv−users list. The lirc programmers will be the ones that can assist you best.

The third option is to dispense with lirc altogether by purchasing an IR keyboard (various options exist, although Chicony appears to work for some people) and a learning remote control. The IR keyboard receiver plugs into the PS/2 keyboard port on your PC and you would train your learning remote to emulate the various keystrokes from `keys.txt` of your IR keyboard. Using this method removes lirc entirely from the picture − your remote will be sending keypresses that your PC "sees" on the keyboard port.

Recompiling your kernel on Red Hat Linux

Don't Panic! Red Hat makes a kernel recompile very easy. They provide configuration files that make it so you can recompile your kernel with Red Hat defaults. And as a side benefit, your kernel will be compiled specifically for your architecture. For a more detailed description (or if you run into trouble) of how to do this go to http://www.redhat.com/docs/manuals/linux/RHL−8.0−Manual/custom−guide/ch−custom−kernel.html

NOTE: Suitable Red Hat kernels for MythTV (i2c update, v4l2 api, CONFIG_SERIAL modularized) are available from ATrpms: http://atrpms.physik.fu−berlin.de/name/kernel/. You will also find suitable kernel

modules for ALSA, lirc, bttv, ivtv, saa7134 etc. there. See also the main Red Hat section.

For the impatient: Make sure you have an emergency floppy boot disk available!

```
$ cd /usr/src/linux
$ su
# make mrproper
# cd configs
# ls
```

at this point you must look at the file names and determine which config file is right for you. For instance if you have an Athlon XP you should choose "kernel−2.4.18−athlon.config" but if you have a dual processor P4 you might choose "kernel−2.4.18−i686−smp.config"

```
# cp yourconfig.config ../.config
# cd ../
# make xconfig
```

At this point you will get a graphical configuration utility. All that you must do to make lirc work is go to the "character devices" section and change "Standard/generic (8250/16550 and compatible UARTs) serial support" from "y" to "m". Now if you want you can have a look around. For instance, you might wish to turn off ham radio support, or perhaps turn off PCMCIA support if you are not on a laptop. But be careful! Only change things you know you can change. If you are not sure, just stick to the one required change. Click "save and exit". Next:

```
# make dep
# make clean
# make bzImage modules
# make modules_install
# make install
```

if you are using **grub**, that should be it, reboot and select your custom kernel upon boot. If you are using **lilo**, change your lilo configuration according to http://www.redhat.com/docs/manuals/linux/RHL−8.0−Manual/custom−guide/ s1−custom−kernel−bootloader.html Remember, if you have any custom kernel modules (ALSA, etc) you will have to recompile/reinstall those.

8.5 Completing the lirc install.

NOTE to Mandrake 9.1 users: skip to the manual start paragraph below.

If the lirc configure program / compile did not mention anything about a kernel module, then you are finished. If it did mention a kernel module, you must edit the `/etc/modules.conf` file. Add this line as the first thing in the file. It must come first, or it may not work.

```
alias char-major-61 XXX
```

replace XXX with the name which you determined earlier, which in this example was "lirc_serial"

```
$ su
# modprobe lirc_serial
# /sbin/ldconfig
```

Next, we're going to manually start lircd the first time. Mandrake 9.1 users, type: #
`/etc/rc.d/init.d/lircd start` instead of:

```
# /usr/local/sbin/lircd
```

If this fails, complaining of a missing `lircd.conf` file, then you must find or make one. First look for a
premade configuration file at http://lirc.sourceforge.net/remotes/. Mandrake 9.1 users, look in
`/usr/share/lirc-remotes`. If you find one your remotes either on the website or in `/usr/share`,
download or copy the file, name it `lircd.conf` and put it in your `/etc` directory. If you couldn't find your
remote, you must make your own `lircd.conf` file.

To make your own `lircd.conf` file

```
$ irrecord myremote
```

Follow the on–screen directions to train your remote and define keys. If your remote ends up working well,
you should consider submitting your `lircd.conf` file back to the lirc developers. Once finished:

```
$ su
# cp myremote /etc/lircd.conf
```

now try to start lircd again:

```
# /usr/local/sbin/lircd
```

Now, we're going to add the commands necessary for lircd to run each time we boot. Mandrake 9.1 users, you
can execute:

```
$ su
# chkconfig --level 35 lircd on
# exit
```

All other distributions:

```
# cd /etc/rc.d
# cat >> rc.local
echo "Starting lircd"
/usr/local/sbin/lircd
^D
# exit
$
```

This takes care of the lircd portion, which "listens" for the IR signals. If everything went well, the install script
for lircd put an appropriate configuration file for your remote into `/etc/lircd.conf` This file maps the
buttons on the remote control to the IR pulses coming from the receiver.

The next step is to convert those signals into something that can be used to control MythTV. MythTV now
includes native support for lirc and can interact directly with

```
$ cd ~/mythtv-0.13/configfiles
$ cp lircrc.example ~/.lircrc
```

or

```
$ cp lircrc.example.pinnaclestudiopctv ~/.lircrc
```

if you've got a Pinnacle Studio PCTV remote.

```
$ irw
```

Start pressing the keys on your remote; **irw** will print the name of the button as it is defined in your `/etc/lircd.conf`. If you don't see anything at this point, you need to troubleshoot further by going back to the lirc home page and investigating from there.

If it is working, then press **CTRL−C** to abort the program. Once you know that your remote is working, you can either recompile MythTV with native lirc support (edit the `settings.pro` file and remove the "#" from the lirc option, then make `distclean` and `make`) or you need to run the **irxevent** program, which takes the key presses and sends them to MythTV. If you use native lirc support, you don't need to run **irxevent**. If you are going to use irxevent, then you need to run it like this:

```
$ irxevent &
```

If **irxevent** isn't running, then MythTV will not respond to your remote control unless you're using native lirc support.

8.6 Additional information for lirc.

Take a look at the lircrc.example files in the `configfiles/` directory. In my case, (Pinnacle Studio card) the channel up and down functions weren't working, due to the fact that the button names were different than the default `lircrc.example` file that came with MythTV.

The `lircrc.example` file has this:

```
begin
    prog = irxevent
    button = ChannelUp
    config = Key Up CurrentWindow
end

begin
    prog = irxevent
    button = ChannelDown
    config = Key Down CurrentWindow
end
```

but the `/etc/lircd.conf` that comes in the lircd package defines the buttons for the Pinnacle Studio PCTV as:

```
            channel+                    0x0000000000000017
            channel-                    0x000000000000001C
```

rather than "ChannelUp" and "ChannelDown". I added the following to my /home/[yourusername]/.lircrc file:

```
begin
    prog = irxevent
    button = channel+
    repeat = 3
    config = Key Up CurrentWindow
```

```
        end

        begin
            prog = irxevent
            button = channel-
            repeat = 3
            config = Key Down CurrentWindow
        end
```

which took care of basic functionality. Because the PCTV Studio remote has additional buttons, look at the `configfiles/lircrc.example.pinnaclestudiopctv` for an example of how to define additional buttons, and how to debug potential button name conflicts between the `lircrc.example` file and how **your** remote defines the button names.

By examining the button names defined in `/etc/lircd.conf` and using the **irw** program to make sure that your remote is working, you can create the appropriate mappings in `.lircrc` to get excellent remote functionality with MythTV.

Note the **repeat =** parameter. This informs the `irxevent` program to pass through every third keypress. By default, `lirc` will only send one keypress to the application, even if you're holding down the key. The actual **repeat =** number will vary from system to system, so experiment and see which value works best for you.

8.7 Configuring lirc for use with an IR blaster.

By Carlos Talbot, mailto:carlos@talbot.net

Lirc has support for various IR transmitters. A popular model is the Actisys IR−200L http://store.yahoo.com/snapstreammedia/irblasbun.html. It was originally designed for IRDA communication, but can be used to transmit A/V remote control codes. By using the lirc SIR driver, this device can easily be integrated with MythTV. I have tested this device with an AT&T DCT2000 digital cable box but the instructions can be used to configure other IRDA devices and A/V remotes.

Follow the steps in the previous section. When you run setup.sh, select option 1, driver configuration. From here select option 6, IrDA hardware. Select your appropriate device and the corresponding serial port, then Save configuration & run configure from the main menu. Once configure is done type:

```
        $ make
```

Please note: unlike the Pinnacle receiver above you will be compiling lircd in addition to a kernel module for the SIR transmitter. Depending on whether you have your serial port driver configured as a kernel module you might see the following message during make:

```
        lirc_sir.c:56:2: warning: #warning
        "*****************************************"

        lirc_sir.c:57:2: warning: #warning "Your serial port driver is compiled into "

        lirc_sir.c:58:2: warning: #warning "the kernel. You will have to release the "

        lirc_sir.c:59:2: warning: #warning "port you want to use for LIRC with:"

        lirc_sir.c:60:2: warning: #warning "setserial /dev/ttySx uart none"

        lirc_sir.c:61:2: warning: #warning
```

```
"*****************************************"
```

If you do receive this statement make sure to run the **setserial** command before you load the lirc_sir module. Follow this with the install:

```
$ su
# make install
```

You will notice that lirc installs the kernel module in `/lib/modules/uname -a/misc`.

The configuration for starting **lircd** differs if you're going to be sending and receiving IR versus just receiving.

```
# cd /etc/rc.d
# cat >> rc.local
echo "Starting lircd"
setserial /dev/ttySx uart none        # (if required)
modprobe lirc_sir
/usr/local/sbin/lircd
^D
# exit
$
```

At this point you have to populate the `/etc/lircd.conf` file with the proper codes for your A/V remote. You should be able to find your remote within the lirc remote tar file located at http://www.lirc.org/remotes.tar.bz2. In my case I extracted the file from remotes/motorola/DCT2000 (gi−motorola−dct2000)

To test the lirc_sir module you can run **irw** to verify the codes are being received. If everything is configured correctly you should see something similar to the following:

```
$ irw
0000000000007ff0 00 1 gi-motorola-dct2000
000000000000bff8 00 2 gi-motorola-dct2000
000000000000f7f0 00 ENTER gi-motorola-dct2000
```

Once you've verified lirc is working you can press **CTRL−C** to exit **irw** and configure the channel changing script.

The path to the channel changing script will need to be entered on the mythbackend setup screen for Input Connections.

This csh script will be called each time MythTV needs to change the channel. Below is a copy of the script followed by the corresponding perl script. Make sure both are in your path. Also make sure you leave the #!/bin/csh setting and not change it to Bourne or bash. This will create a frustrating symptom to diagnose where MythTV cannot open /dev/device. Unlike Bourne or bash, csh scripts automatically close parent file descriptors before they start.

```
$ cd /usr/local/bin
# su
# cat > change_channel.csh
#!/bin/csh
echo "changing to $1"
/usr/local/bin/channel.pl $1 &
```

8.7 Configuring lirc for use with an IR blaster.   37

```
^D
# chmod a+x change_channel.csh
# exit
$ exit
```

See `contrib/channel.pl` for the actual file. Copy it to `/usr/local/bin/`

The last statement within the perl script is the lirc rc command. This is the command that transmits the code to your cable/DSS box. Make sure to have the IRDA device within a few feet of the box.

9. Configuring MythTV.

By this point, all of the prerequisites have been installed, **mysql** is running and has had its initial database setup. It's now time to configure MythTV.

9.1 Configuring the master backend system.

Open a shell and decide where you will store your video files. By default, MythTV will assume that you will be using `/mnt/store/`, but if this directory doesn't exist, or it exists and you don't have write privileges to it then MythTV will fail when you attempt to make a recording. The following example is specific for `/var/video`, but the same instructions would apply if you want to keep the default but don't have the directory structure setup yet. See the Advanced Partition Formatting section for hints on creating a partition for MythTV.

If you create a `/var/video` subdirectory, change `/mnt/store/` to `/var/video/` in the setup screens.

```
$ su
# mkdir /var/video
# chmod a+rwx /var/video
# exit
```

The first thing to configure is the master backend system. If you are running multiple backend systems, the master backend will make all decisions about which programs will be recorded on which tuners. If you have only one backend, then it will be its own master.

The master will always choose the first available tuner in the same order as you add cards through "setup". In other words, the second card you add will only be used when there are two overlapping recordings, the third when there are three, and so on. Therefore, you will want to have your largest disk space on the master because the tuner on the master will always be the first choice. You will then want to add your other backends in the order of your preference for recording.

See Advanced Backend Configurations for information on configuring multiple backend systems in various ways.

Because MythTV now uses a database to store most configuration variables, part of the bootstrap of MythTV is to indicate the location of the MySQL database server. If the frontend, backend and MySQL database server are all going to be running on the same box, you can continue to the next step. If not, you'll need to edit the `/usr/local/share/mythtv/mysql.txt` file and change the first line to the IP address of the database server. All other configuration is within the MythTV GUI screens.

NOTE: Users that have been running the frontend and the backend on different machines using Red Hat Linux 8.0 have stated that they have been having issues with remote access to the MySQL database. The following instructions may or may not work. Add the following to `/etc/my.cnf` on the backend machine and restart MySQL.

```
skip-innodb
set-variable=thread_stack=256k
```

Switch to the setup directory and run the setup program:

```
$ cd setup
$ ./setup
```

The backend setup program will start and offer you a number of choices. It is *strongly* recommended that you go through them in order.

The first question will ask if you wish to clear out your existing configurations for your capture cards. Initially, you should say "YES" so that there are no surprises later.

The next question will ask you if you wish to clear out your video source information. You should answer "YES" to this as well.

Once the graphical setup starts, you'll see that there are four choices.

1. General
2. Capture Cards
3. Video Sources
4. Input Connections

Use the arrow keys to move around, and press the space bar to select which option you wish to configure.

General

The first screen of the General configuration deals with IP addresses of the system that you're running setup on and any master backend you may have. If you've only got one machine, then the default values are fine and you can move to the next page by pressing the space bar. If you need to move around the screen, use the arrow keys to move focus between settings, not the mouse.

If you will have multiple backends, then *do not* use the "127.0.0.1" IP address.

It is very strongly discouraged that you make any changes to the port settings from the defaults.

Once you're satisfied with the values, move the focus down to Next and hit the space bar.

The next screen details the Host−specific Backend setup. This is where you will set the specific directory paths for this particular backend. Make sure that you've followed the steps at the beginning of this section and created a directory that exists and that MythTV will have write privileges to. When you're done, press Next to continue, taking you to the Global Backend Setup.

On the Global Backend Setup configure your backend with the appropriate settings. Use the left and right arrow keys to iterate through the choices available on each setting, and the up and down keys to move

between settings. Move to Finish when you're done and press the space bar, taking you back to the main configuration screen.

Capture Cards

You should have no capture cards defined, so the highlight will be on (New Capture Card). Press space to begin.

Choose the appropriate settings for your particular tuner. Use the arrow keys to move around and to make your choices, and press RETURN when complete. Pressing RETURN will take you back to the Capture Cards screen; if you have additional capture cards in this machine, press the space bar when the highlight is on the (New Capture Card) row to define another card. Once you have no additional cards to setup, press ESC.

Video Sources

When you start, the highlight should be on (New Video Source). Press the space bar to begin. The first field asks for the name of the video source. You may choose something easy to remember, like "Antenna" or "Cable". Once you've chosen a name, press the down arrow to move to the next field. If you're in North America, move to the Zip/postal code field and put in the appropriate value.

If you're outside of North America, then some manual interaction will be required with XMLTV.

NOTE: If you are in North America, and you enter your Zip/postal code and nothing seems to happen for a very very long time, it is possible that XMLTV is waiting for you to provide some input. Use ALT−TAB or whatever key combination your window manager uses to switch programs and look for a terminal window running XMLTV that may be waiting for input.

Once you have chosen your provider, press RETURN to continue. XMLTV will now begin collecting the initial data for your location. The screen may blank for a few seconds to several minutes, depending on the load of the listings provider and the speed of your connection to the Internet. Be patient!

You will then be returned to the Video Sources screen. If you have multiple video sources available, such as Antenna, Cable, etc, go ahead and define them all, even if they're not all going to be physically connected to the master backend server. Once you're done, press ESC to return to the main screen.

Input Connections

The final configuration item is Input Connections. On this screen, you will associate the various video sources you defined earlier with a physical input to a encoder card. It's entirely possible that you have multiple tuners, and each tuner has a different input, so on this screen you let MythTV know which device will connect to which input source.

When you start this screen, you should see a listing of the various input connections available on each of the Capture cards you defined earlier. For example, you may have a capture card with a tuner, a SVideo and a Composite connection. If you wanted to associate the tuner (a.k.a., "Television") with an "Antenna" source you defined in Video Sources, you would move to the /dev/videodevice (Television) −> line and press the space bar. Using the left and right arrow keys will show you the various choices you have already created for video source. In our case, you would use the left/right cursor keys until "Antenna" was shown in the Video Source field. Press down to move to the next setting, all of which should be self−explanatory.

Once you're done, press RETURN to go back to the Input Connections screen. You would then finish associating the video sources to any other hardware devices you have available.

NOTE: Don't add a video source to a hardware input if you don't actually have anything connected there. For example, adding "Cable" to the Tuner and to the Composite inputs without having something connected to Composite will lead to blank recordings.

Press ESC to return to the main menu, and press ESC again if you have no further items to configure, thereby returning you to the command line.

9.2 Post−configuration.

Run the `mythfilldatabase` program as directed. The master backend will obtain guide data for all the video sources you defined during setup.

Once `mythfilldatabase` has finished, start the master server before continuing.

```
$ mythbackend
```

mythbackend will print information about connections and what it's doing to the console. If you'd like to see the options that are available for mythbackend, type `mythbackend −h` for help.

As of 2003−10−03, the available options are:

```
$ mythbackend −h
Valid options are:
−l or −−logfile filename      Writes STDERR and STDOUT messages to filename
−p or −−pidfile filename      Write PID of mythbackend to filename
−d or −−daemon                Runs mythbackend as a daemon
−v or −−verbose debug−level   Prints more information
                              Accepts any combination (separated by comma)
                              of all,none,quiet,record,playback,
                              channel,osd,file,schedule,network
−−printexpire                 List of auto−expire programs
−−printsched                  Upcoming scheduled programs
```

Running mythbackend as a daemon and using the logfile option will allow you to have mythbackend automatically start up during the boot process. For example, you may add this to your `rc.local` file, or you can follow the steps outlined in the section called I'd like to automatically start mythbackend at system boot time.

9.3 Configuring a non−master backend.

Configuration of a non−master backend follows the same general procedure as that of the master backend, with the exception that you skip over the "Video Sources" step. All possible video sources need to be defined on the master backend system; only the master backend will query a listings provider to obtain guide data for all the slave backends.

Make sure that the IP addresses on the General setup screen are accurate. If the slave backend can't communicate with the master backend due to IP address misconfiguration then MythTV will not function properly.

Ensure that you've granted access to the master MySQL database for remote backends as discussed in the section titled Modifying access to the MySQL database for multiple systems and that you've edited the `/usr/local/share/mythtv/mysql.txt` file and put the correct IP address for the database server.

NOTE: Do not run `mythfilldatabase` on a non−master backend.

9.4 Configuring and running mythfilldatabase.

NOTE: **mythfilldatabase** might take a while to complete, depending on any number of factors, most of which you can't control. It's best to just let the program run to completion.

mythfilldatabase −−manual is another option; the manual option will allow you to fine tune channel frequencies and specify which channels will be added to the database. If you are not using the tv_grab_na grabber, you must use the −−**manual** option.

mythfilldatabase −−file is an option if there isn't an XMLTV grabber for your country, but you **do** have an XML formatted listings file created by some other program.

mythfilldatabase −−xawchannels is an option if you have used **xawtv** to fine−tune your channels and would like to import the fine tuning offsets into MythTV.

Setting mythfilldatabase to run from cron

In order to keep your database filled, **mythfilldatabase** should be run once a day.

There are two ways to accomplish this: either as a user−based cron job, or as a system based cron. If you use a user−based cron, you can select the time that the cron job will be run. A system−based cron job will run at the same time as any other system cron jobs.

If you are a Mandrake 9.0 user and wish to use a system cron job, you can perform the following steps:

```
$ cd configfiles
$ su
# cp mythfilldatabasecron /etc/cron.daily
```

If you want to know what time your job will run, you can do the following:

```
# cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root nice -n 19 run-parts /etc/cron.hourly
02 4 * * * root nice -n 19 run-parts /etc/cron.daily
22 4 * * 0 root nice -n 19 run-parts /etc/cron.weekly
42 4 1 * * root nice -n 19 run-parts /etc/cron.monthly
# exit
$
```

We can see from the display that the cron.daily job will run at 0402 every morning.

If you wish to specify a custom time, then the user–based cron job is the better method.

The default **configfiles/mythcrontab** file will run the **mythfilldatabase** program every day at 0430. Placing a copy of the default **configfiles/mythcrontab** into your ~/.mythtv directory will allow you to edit your copy and maintain the original if you get into trouble. To change the time used to start the **mythfilldatabase** process, open the ~/.mythtv/mythcrontab file in your favorite editor and modify it as you like. Once you're satisfied with the time that **mythfilldatabase** will run, save and exit, and return to the shell prompt.

```
$ cp ~/mythtv/configfiles/mythfilldatabasecron ~/.mythtv
$ cp ~/mythtv/configfiles/mythcrontab ~/.mythtv
$ cd ~/.mythtv
$ crontab mythcrontab
$ crontab -l  NOTE: this is a lowercase L.
# The following will run the mythfilldatabase shell script at 0430
# every day.  Format is
# minutes hour day_of_month month day_of_week
# man 5 crontab for more information
# Need to use a shell script because the QTDIR environment variable isn't
# being honored by crontab?
# The next text should be all on the same line.
30 04 * * * /bin/bash $HOME/.mythtv/mythfilldatabasecron
$
```

As you can see, **mythfilldatabase** will now be run at 0430 every day.

If you decide to alter the time that the **mythfilldatabase** runs, you can use the crontab command to open your existing job and edit it in place.

```
$ crontab -e
```

The **crontab -e** command will allow you to edit the existing crontab. **NOTE**: By default, the crontab will use **vi** as your editor. If you don't like to use **vi**, make sure that you've set VISUAL to something other than **vi**.

```
$ export VISUAL=/usr/bin/joe
$ crontab -e
```

This will cause **crontab** to use **joe**, a visual editor that uses the Wordstar command sequence. You may use **emacs**, or any other editor that you're comfortable with. You may also add the EXPORT command to your .bashrc or to the /etc/profile.d/mythtv.sh file.

10. Configuring mythfrontend.

Once you have completed configuration of your backend systems, the next step is to configure the frontend client. You should run the MythTV setup program before using any other modules such as MythMusic, etc. to correctly initialize the database.

When you start mythfrontend, you should have a number of choices. Before doing anything, go to TV, then to Setup and configure the frontend client.

10.1 General

The General screen has configuration items that don't really fit anywhere else. The first few configuration items asks you to indicate the number of seconds to record before or after a program, which is useful if the

broadcast network or your system clock are out of sync and will help prevent you missing the beginning or end of a program.

To change the value, use the left and right arrow keys to increment and decrement the number of seconds. When you're satisfied with the result, use the down arrow to put the input focus on the Next button or press RETURN to continue to the next page.

The next page has a number of options to do with how channels are displayed on your system. The help text will give you more information. Move the focus to Next and press the space bar to continue.

The final General page sets up some final configuration items. See the help text for more information.

10.2 Appearance

This set of screens is mostly concerned with how MythTV will look on your system. From here, you can choose different themes and set the resolution of your system.

10.3 Program Guide

Fairly self explanatory. Note that the alternate program guide does not use the same font settings as defined in Appearance, so if the EPG is unreadable this is where you make the adjustments to fonts, number of elements displayed, etc.

10.4 Playback

The one configuration item which may cause problems on your system is the "Deinterlace playback" setting. MythTV uses a linear blend algorithm for deinterlacing, which will improve how the image looks on your screen. Deinterlacing requires that your processor support SSE. (Streaming SIMD Extensions, aka "MMX2"). Early Intel Celeron (those that don't use the Coppermine 0.18um core and are usually <600Mhz), Pentium Pro and Pentium II CPUs do not have SSE, so make sure that you haven't enabled deinterlacing if your processor doesn't support it. If you enable it, and your processor doesn't support SSE, you will get "Illegal Instruction" errors.

To determine if you've got SSE on an Intel processor, you can:

```
$ cat /proc/cpuinfo
[snip]
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
cmov pat pse36 mmx fxsr sse
```

Notice the **sse** at the end of the line – this tells you that this processor will be able to deinterlace correctly.

On an AMD processor, look for "3dnow" in the cpuinfo line; "3dnow" is AMD's implementation of SSE instructions, so if your processor has 3dnow you shouldn't have any issues with deinterlacing.

10.5 Recording

Depending on your capture card, MythTV offers different video encoders. The following types of hardware encoding cards are supported:

- MJPEG – Zoran–based cards; see http://mjpeg.sourceforge.net
- MPEG2 – iTVC15 based cards (Hauppague PVR–250/PVR–350); see http://ivtv.sourceforce.net
- HDTV – pcHDTV cards; see http://pchdtv.com
- DVB – cards supporting DVB; see http://linuxtv.org

For cards without hardware encoding capabilities (all cards supported by V4L not listed above), Myth includes two methods for software encoding: RTjpeg and MPEG4. RTjpeg has significantly fewer CPU demands than MPEG4, but it generates larger files than MPEG4 for a given recording.

For DVB and HDTV cards, no further configuration is required after setting up the card using the 'setup' program. For all other cards, configuration is done through MythFrontend. Selecting 'Recording Profiles' from the 'TV Settings' screen will list the profiles currently available for the cards in your system. Depending on what types of cards you have installed you may see:

```
(Create new profile group)
Software Encoders
Hardware MPEG Encoders
Hardware MJPEG Encoders
Trancoders
```

The '(Create new profile group)' option will allow you to create custom profiles in case you have multiple backends. Note that custom profiles are per backend and cardtype. If you have 2 MPEG–2 encoders in a given backend system, creating a custom profile will affect both of them. This option should not be needed otherwise.

The 'Transcoders' group is a little different from the others. Selecting this group will result in a menu with the following options: 'RTjpeg/MPEG4' and 'MPEG–2'. These types indicate what transcoder options will be used for a given input type (i.e. the 'MPEG–2' settings would be used to transcode MPEG–2 files into MPEG–4). The source of the MPEG–2 stream (DVB, HDTV, or PVR–x50) does not matter. Configuration of the options is the same as below (although any resolution settings will be ignored).

Selecting any of the other options will show a new screen with a list of four profiles:

- Default
- Live TV
- Low Quality
- High Quality

The Default profile will be used for any recording which does not otherwise have a specific profile assigned. The 'Live TV' profile will be used when watching TV. The remaining two profiles are available for customizing to allow for more precise control over what quality is used for a given program.

Selecting a profile will allow you to adjust the relevant options for that card. The most significant setting is the recording resolution, but you can also choose encoding format, audio format, and tweak other encoder specific properties.

NOTE: although the width and height can be changed to almost any value, if you start MythTV and don't see video or you get "segmentation fault" errors, it is likely that the video4linux (v4l) subsystem did not like the height and width parameters specified. It's best to leave the default as–is until you're sure that MythTV is operational.

NOTE: although the width and height can be changed to almost anything, if you start MythTV and don't see video or you get "segmentation fault" errors, it is likely that the video4linux (v4l) subsystem did not like the height and width parameters specified. It's best to leave the default as−is until you're sure that MythTV is operational.

10.6 Xbox Frontends

MythTV is able to control the LED on the Xbox to indicate backend recording status.

To control the LED, you will need the **blink** program from the xbox−linux project, which is installed as `/bin/led` on GentooX. On Xebian (the new Ed's Debian) you must install it yourself. On other distributions it may or may not be installed as a program called **blink** and should be located in your path. (Type `which blink` to see if the program is available.) If you do not have **blink**, you may obtain it from the Xbox−Linux project site at http://xbox−linux.sf.net/. The program you need is part of the `eds_i2c_staff` module in CVS. Note the spelling.

Once you have installed **blink** you will need to set permissions. **blink** needs write permission to the i2c device to function properly. There are three methods to accomplish this. First, you could run `mythfrontend` as root, which is the simplest method, but could potentially be a security risk. Next, you may make the **blink** binary setuid root, which allows non−privileged users to run a program with root capability. This is done by typing the command:

```
$ su
# chmod u+s /path/to/blink
```

The final technique would be to set the `/dev/i2c/0` device read/write for all users, but this is the least preferred method.

Now it's time to setup MythTV for Xbox hardware. Enter Setup −> General. On the second page check the 'Enable Xbox Hardware' option. Upon reentering the settings, you should have a new option named 'Xbox'. Within this option you may select the distribution, LED colors for recording and the update interval. If you select GentooX as the distribution **led** will be used as the **blink** binary name, otherwise, **blink** is used. Colors should be self explanatory. The update interval determines how often the frontend should poll the backend to determine if the status has changed.

11. Using MythTV.

The `keys.txt` file describes what the various keyboard commands are.

NOTE to Red Hat users:Red Hat Linux 9/8.0 ships with Gnome as the default desktop environment. However, Gnome seems to have issues with window focus and window switching which sometimes cause mythfrontend to obscure the video. KDE does not seem to have any such issues. Therefore you will need to switch to KDE by selecting RedHatMenu>Extras>System Settings>Desktop Switching Tool and choose "KDE".

11.1 Using themes with MythTV.

MythTV is "themeable", meaning that the visual appearance of the program can be modified by the user without re−compiling or altering the program functionality. The MythTV website has a Themes section; pick

a theme you'd like to use and download it to your system.

Once it's on your machine, you must move the file and untar it:

```
$ su
# cp funhouse.tar.gz /usr/local/share/mythtv/themes
# cd /usr/local/share/mythtv/themes
# tar -xzf funhouse.tar.gz
# exit
```

The theme will now be available in the mythfrontend Appearance section.

NOTE: if the theme file you downloaded is a `bz2` file, use `tar -xjf` rather than `tar -xzf`.

11.2 Adding DirecTV information to the database.

A script for adding DirecTV information into the database has been written by tarek Lubani mailto:tarek@tarek.2y.net and is available at http://tarek.2y.net/myth/ppv.pl. This is currently an external user–supplied program, so if you have issues with the script, please contact the author.

11.3 Adding support for an external tuner.

MythTV supports changing the channel on an external tuner. If you have an external tuner, such as a DirecTV or digital cable set top box, you should add `/usr/local/bin/changechannel` to your Input Connections in the mythbackend configuration GUI.

The **changechannel** program is not supplied with MythTV, so this is going to be dependent on what sort of external tuner you have. Example scripts for Sony and RCA receivers can be obtained from http://tarek.2y.net/myth/. This is currently an external user–supplied program, so if you have issues with the script, please contact the author.

Feel free to browse some of what sort of hardware is available at http://shop.store.yahoo.com/snapstreammedia/cablepacks.html, or if you wish to assemble your own, rather than purchase, the following may be helpful: http://www.snapstream.com/products/irblaster/sonydss.htm for cable pinouts for RCA DSS receivers.

12. MythWeb.

The MythWeb allows you to use a web page to control various aspects of your MythTV system. MythWeb is a separate application, but it's dependent on MythTV being installed and operational.

12.1 Installation and prerequisites.

Download mythweb from http://www.mythtv.org/ and save it to a location where you can find it. The next step depends on whether your distribution has a web server and if you have PHP support.

Mandrake 9.0

Mandrake 9.0 has **apache** and **PHP** pre−packaged, so installation is quite simple. Once you have installed the two programs, either with the graphical installer (see instructions in Graphic Install Tools) or through the command line (section Command Line), and you've ensured that the **apache** server will start at boot using the Mandrake Control Center.

NOTE: Mandrake 9.1 users, perform the following:

```
# urpmi apache2 apache2-mod_php php-mysql
# chkconfig --level 345 httpd on
# /etc/rc.d/init.d/httpd restart
```

12.2 Completing the installation.

```
$ tar -xjf mythweb-0.13.tar.bz2
$ cd mythweb-0.13
$ su
# mkdir /var/www/html/mythweb
# cp -r . /var/www/html/mythweb
# exit
$
```

Edit the `/var/www/html/mythweb/config/conf.php` file if required.

By default, MythWeb uses an Apache `.htaccess` file to restrict access to the website and to configure some variables.

To create the password file for Apache (if your system doesn't already have one), you could do something like this:

```
# cd /var/www
# htpasswd -c htpasswd mythtv
New password:
Re-type new password:
Adding password for user mythtv
```

See the man page for **htpasswd** for more examples.

To access the web page, open a web browser and use **http://[name or ip address]/mythweb/**

NOTE: Make sure that you have a trailing slash on the URL, otherwise you will get a `404 Page not Found` error.

13. MythGallery.

MythGallery is a photo and slideshow application. MythGallery is a separate application, but it's dependent on MythTV being installed and operational.

13.1 Installation and prerequisites.

Download MythGallery from http://www.mythtv.org/ and save it to a location where you can find it.

```
$ tar -xjf mythgallery-0.13.tar.bz2
$ cd mythgallery-0.13
$ qmake mythgallery.pro
$ make
# su
# make install
# exit
$
```

The configuration for MythGallery is accessed through the main Setup option in mythfrontend. Make sure you set your pictures directory to wherever you're storing your photos.

- Use the arrow keys to move around the thumbnails.
- Use SPACE or ENTER to maximize a photo.
- Use p or P to start the slideshow of all the files in the directory. "Play" or "Pause" on your remote may also work.
- Use SPACE or ENTER to advance to the next photo.
- Use ESC to go up one level, or to exit out of the module if you're at the first level.
- Use the [and] keys to rotate the image.
- Use M to bring up the setup dialog.
- Use I to import pictures; when a photo is maximized this will bring up photo information.
- 7 and 9 are used to Zoom Out and in, 2, 4, 6 and 8 are used to move around while in zoom mode, 5 centers and 0 exits zoom.

13.2 Importing Pictures

The import path in the setup dialog is a colon separated list of directories and/or executable files. When the import key is pressed, a new directory (the destination directory) under the current directory will be created and the import path will be searched. If the item in the import path is a directory (the source directory), the contents of that directory will be copied to the destination directory. If you would like the source directory to be that of a removable device, it might be a good idea to use autofs. See the automount howto at www.linuxdoc.org for info on how to get it working.

If the item in the import path is an executable file, MythGallery will attempt to execute it with the destination directory as its sole argument. Be careful when using executable scripts that the script runs unattended (doesn't need user intervention) and returns properly, otherwise it could create the appearance of MythGallery hanging (e.g. running **smbclient** and prompting for password). Also be sure that scripts have executable permissions set.

Here is an example script that a user may want to run on import:

```
#!/bin/csh

if ($#argv == 0) then
        echo "Usage: $0 dest_dir"
        exit
endif

cd $argv[1]
```

```
# get stuff over the network
wget http://www.somesite.dom/dir/file1.jpg
wget http://www.somesite.dom/dir/file2.jpg
wget http://www.somesite.dom/dir/file3.jpg

# stuff that requires manual module loading and/or fs mounting
modprobe camera_module
mount /dev/camera /mnt/camera
cp /mnt/camera/* $argv[1]
umount /mnt/camera
rmmod camera_module

# perform some processing
foreach pname (`ls *.jpg`)
        jpegtran -flip vertical $pname > $pname.new
        mv $pname.new $pname
end
```

14. MythGame.

15. MythMusic.

MythMusic has a number of prerequisites that must be satisfied before it is operational. Depending on your distribution, some of these prerequisites can be satisfied through the various package managers. If your distribution doesn't offer pre−compiled versions of the software below, then follow the generic instructions for manually compiling and installing the software.

The prerequisites for MythMusic are:

- MAD
- id3tag
- libvorbis−devel
- FLAC
- libcdaudio
- CDParanoia

15.1 Manual installation of prerequisites.

These instructions are for distributions which don't have pre−compiled versions of the software necessary to run MythTV.

Download MAD from http://www.mars.org/home/rob/proj/mpeg/ Save it to a directory you can find later.

```
$ tar -xzf mad-0.14.2b.tar.gz
$ cd mad-0.14.2b
$ ./configure
$ make
$ su
# make install
# exit
$
```

Download libid3tag from http://www.underbit.com/products/mad/ Save it to a directory you can find later.

```
$ tar –xzf libid3tag-0.15.0b.tar.gz
$ cd libid3tag-0.15.0b
$ ./configure
$ make
$ su
# make install
# exit
$
```

Download FLAC from http://flac.sourceforge.net and install:

```
$ tar –xzf flac-1.1.0.tar.gz
$ cd flac-1.1.0
$ ./configure
$ make
$ su
# make install
# exit
$
```

Download libcdaudio from libcdaudio.sourceforge.net and install:

```
$ tar –xzf libcdaudio-0.99.9.tar.gz
$ cd libcdaudio-0.99.9
$ ./configure
$ make
$ su
# make install
# exit
$
```

Download cdparanoia from http://www.xiph.org/paranoia/down.html.

```
$ tar –xzf cdparanoia-III-alpha9.8.src.tgz
$ cd cdparanoia-III-alpha9.8
$ ./configure
$ make
$ su
# make install
# cd /usr/lib
# ln –sf libcdda_interface.so.0.9.8 libcdda_interface.so
# ln –sf libcdda_paranoia.so.0.9.8 libcdda_paranoia.so
# exit
$
```

15.2 Mandrake 9.0

Mandrake 9.0 has a number of the prerequisites available on the installation CD. Some of the software you're going to need will have to be obtained from the "contrib" or "cooker" development repositories. Applications downloaded from "cooker" come from the development branch, so there may be issues with some software. It isn't recommended that you mix cooker and release–level software.

urpmi is the simplest tool for installation of packages from the command line. The difficult part is the configuration, but this has been made easier at the following website: http://plf.zarb.org/~nanardon/urpmiweb.php The website will allow you to choose a mirror site and then present the command–line configuration text for that mirror. You will most likely need to add a "Contrib" mirror to your setup. Once you have done that, you can proceed. If **urpmi** prompts you about other modules

that need to be installed to satisfy dependencies, say "Yes".

```
# urpmi libmad0 libmad0-devel libflac4 libflac4-devel libcdaudio1 cdparanoia
# urpmi libcdda0 libcdda0-devel libvorbis0 libvorbis0-devel
# urpmi libcdaudio1-devel libid3tag0 libid3tag0-devel
```

Additional options with MythMusic

Additional visualizations have been added to MythMusic. If you wish to use these, there are some prerequisites you must install prior to compiling.

- fftw
- OpenGL
- SDL

`fftw` may be obtained from http://www.fftw.org/. In Mandrake 9.1 it may be installed by typing:

```
# urpmi libfftw2 libfftw2-devel
```

`OpenGL` should be installed on practically all distributions. However, you will need the devel module. In Mandrake 9.1 it may be installed by typing:

```
# urpmi libMesaGLU1-devel
```

`SDL` may be obtained from http://www.libsdl.org. In Mandrake 9.1 it may be installed by typing:

```
# urpmi libSDL1.2 libSDL1.2-devel
```

15.3 Red Hat Linux 9/8.0

Red Hat provides packages for several of the prerequisites, making installation very simple. Of the prerequisites, Red Hat provides packages for Vorbis, cdparanoia, SDL, and OpenGL (which you probably already have installed). To install these all at once, simply type (all on the same line):

```
$ up2date --solvedeps libvorbis libvorbis-devel vorbis-tools cdparanoia-devel cdparanoia-l
```

If you get the following message: "None of the packages you requested were found, or they are already updated" it probably means you already have all of those packages installed.

You must install the remaining packages, (MAD, FLAC, libcdaudio and optionally fftw) manually following the installation directions above. When installing fftw do not use the rpm package offered on the website because it will cause an error, so use the source package instead.

NOTE: you can use the instructions given at the automated installation section to install all of MythMusic in one step.

15.4 Compiling MythMusic.

Once all the prerequisites have been installed, you can proceed with getting MythMusic going. Make sure that you run `./configure` first. If you wish to use the new visualizations, make sure you install the prerequisites. Run `./configure --help` for help.

```
NOTE for Red Hat users: Before compiling, make the following modification
to settings.pro, otherwise the compile will fail:

Find the following text:
INCLUDEPATH *= /usr/local/include

immediately below that add
INCLUDEPATH *= /usr/include/cdda

save and close settings.pro and proceed with compiling MythMusic.
```

```
$ qmake mythmusic.pro
$ make
$ su
# make install
# exit
$ cd musicdb
$ mysql -u root < metadata.sql
```

15.5 Configuring MythMusic.

Configuration of MythMusic occurs in two places. The main mythfrontend Setup is for global MythMusic configuration. Go to the Setup/MythMusic/General Setup screen and adjust it for your particular setup.

The second configuration screen is within the MythMusic program and will allow you rescan your music library, etc.

Here's some explanation about the Ignore_ID3 and The NonID3FileNameFormat:

If Ignore_ID3 is set to TRUE, MythMusic will try to determine the Genre, Artist, Album, Track Number, and Title from the filename of the mp3 file. The NonID3FileNameFormat variable should be set to the directory/file format where the mp3 files are stored. For instance, I store mine in the above shown Genre/Artist/Album/Track format. MythMusic will then use this information to fill in the proper fields when it populates the musicmetadata table rather than searching for an ID3 tag in the mp3 file.

The files can be laid out in any format, such as:

Genre/Artist/Album/Title Artist/Genre/Album/Title Artist/Album/Title (with Genre left as Unknown)

The track number is optional but can be specified with the title by using the TRACK_TITLE keyword instead of TITLE. If TRACK_TITLE is used, then the filename can have a space, hyphen, or underscore separating the track number from the track title. Keywords are case insensitive, so if you specify GENRE it's the same as Genre in the format field.

The Ignore_ID3 option does not disable the code that determines the track length, just the portion that tries to read ID3 info.

15.6 Using MythMusic.

MythMusic is fairly simple to use. It is recommended that you insert the CD before selecting "Import CD". You should also ensure that your system doesn't try to automount the CD and begin playing it automatically.

Another item to consider: there are some CDs that contain computer data that runs as a "CD Extra" when inserted into some Windows PCs and Macintoshes. As of 2003–06–10, MythMusic doesn't support track skipping or individual track selection, so if you have a CD with "CD Extra" data you will not be able to encode it; MythMusic will hang attempting to encode the non–audio data.

Here's some information on playlist management:

Q: How do I create a new playlist? A: Using the MythMusic "Select Music" menu option, setup the playlist as you normally would by adding songs or other playlists as needed. When you are ready to save the new playlist, highlight "Active Play Queue" at the bottom of the selection tree and hit the "i" key. This will pop up a menu allowing you to name and save the new playlist. You can also hit Enter to bring up the popup on the Active Play Queue. This does not work on the playlists above, as Enter is obviously bound to checking/unchecking the boxes. Any number (ie. keypad on remote) will also bring up the menu in both cases.

Q: How do I enter the playlist name in the text field without a keyboard? A: Use the keypad number keys (bound to your remote) to select letters quasi–cell phone style. Keys 2–9 work pretty much like any cell phone text entry. 1 cycles through a few special characters, delete, and space. 0 is like a CAPS LOCK. Hard to describe, fairly easy to use. You will soon be able to specify the cycle timing in a Setup screen. You can type fairly quickly through a combination of jumping around the number keys AND hitting a non–number key (right arrow is particularly good for this) to force the current character.

Q: How do I edit a playlist? A: Highlight the playlist in the selection tree and hit the "i" key then select "Move to Active Play Queue" in the popup. You can now modify the "Active Play Queue" like normal, adding songs and playlists by selecting them from the song tree. When you are done, highlight the "Active Play Queue" in the selection tree and hit the "i" key then select "Save Back to Playlist Tree". And whatever you were editing as your Active Queue before you moved an existing playlist "on top" of Active reappears. Think of Active has having a push on, pop off capability, but with a depth of only 1.

Q: How do I delete an item from a playlist? A: Highlight the item in the selection tree and hit the "d" key.

Q: How do I rearrange the songs in my playlist? A: Highlight a song and hit the "space" bar, the song will now have pair of red arrows in front of it. Use the up and down arrow keys to move it around in the playlist. When you have it where you want it, hit the "space" bar again.

Q: How do I delete a playlist? A: Highlight the playlist in the selection tree and hit the "i" key then select "Delete this Playlist" from the popup.

15.7 Troubleshooting MythMusic.

You may run into errors when running MythMusic.

When I run MythMusic and try and look up a CD, I get an error message

The full text of the message will say:

databasebox.o: Couldn't find your CD. It may not be in the freedb database. More likely, however, is that you need to delete ˜/.cddb and ˜/.cdserverrc and restart mythmusic. Have a nice day.

If you get this message, you should go to the home directory of whatever user MythMusic is running as and type:

```
rm .cdserverrc
rm -rf .cddb/
```

These files aren't automatically deleted because of a conscious design decision by the author that programs that automatically delete things are bad.

The files are used to locally cache CD lookups. If you are re–inserting CDs, your machine will not actually have to go out to the Internet to determine what is on them. However, the URL used to access the freedb database has recently changed, so the stale information in the files from previous runs of MythMusic would cause the error above. Once the files have been deleted the stale information will be gone and your local database will be rebuilt as you use CDs.

16. MythWeather.

The MythWeather module will obtain the weather information for the location that you specify. You must be running MythTV v0.10 or later in order to use MythWeather.

```
NOTE for Red Hat Linux 8.0 users: Red Hat 8 users must make the following
modification, or several weather icons will fail to be installed.

From the mythweather build directory move to the mythweather sub-directory
and edit the Makefile there.

Find and comment out (using a '#' symbol) the following line:

strip "$(INSTALL_ROOT)/usr/local/share/mythtv/mythweather/images/"
```

Change into the MythWeather directory, then make and make install:

```
$ cd mythweather
$ qmake mythweather.pro
$ make
$ su
# make install
# exit
```

MythWeather uses MSNBC.com as its source for weather data and weather.com for its radar image.

These are the keyboard commands for MythWeather:

```
Left Key            Goes back one page, and extends the time spent
                    on the page you are on.
Right Key           Goes forward one page, see above.
Space               Pause, wait on the current page until space is hit
                    again.
Numeric Keys        You can check other weather by keying in other ZIP codes.
Enter Key           Switch between Celsius and Fahrenheit. Can also
                    be used a way to force a data update.
"m" Key             Resets the location to the database default, then updates the data.
"i"                 Enter / Save settings
ESC                 Exit the settings screen without saving / Exit the program
```

MythWeather also has an "Aggressiveness" setting. This affects how long MythWeather waits for data from the msnbc.com website before timing out. If you are on a slow connection, or have a slow DNS, or MythWeather just doesn't seem to be working and you've already tried everything else, then try increasing the

aggressiveness level parameter. This parameter is inverse; a higher number actually means that MythWeather will be less aggressive, and will therefore wait longer before timing out.

MythWeather will print debugging information on the terminal. If you wish to see additional debugging information while MythWeather is running, run mythweather from the command line with as `mythweather --debug`

You may also force mythweather to re−run the configuration by starting it on the command line as `mythweather --configure`. These two options are mutually exclusive.

17. MythVideo.

The MythVideo application will allow you to use an external program to watch media files that are not directly supported by MythTV.

Change into the MythVideo directory, then make and make install:

```
$ cd mythvideo
$ qmake mythvideo.pro
$ make
$ su
# make install
# exit
```

The next step is to create the create the videometadata table in the database. Simply do the following:

```
$ cd videodb
$ mysql −u mythtv −p < metadata.sql
```

See MythVideo's `README` file for additional information.

18. MythDVD.

MythDVD is an application which rips DVDs and makes them available for use with MythVideo. You may also transcode the DVD content from MPEG−2 to other formats which should greatly reduce the amount of space the DVD material takes up on your hard drive.

MythDVD has a number of prerequisites to enable transcoding functionality. If you only wish to play DVDs rather than convert them to something like MPEG−4 or xvid you may skip the prerequisite installation step.

18.1 Manual Compilation of Prerequisites.

18.2 Pre−compiled binaries.

Mandrake users may install the prerequisites this way:

```
# urpmi libdvdread3 libdvdread3-devel a52dec liba52dec-devel
# urpmi mplayer ogle xine
```

Assuming that you've added a PLF mirror, you may also load the rest of the prerequsites using the following

command:

```
# urpmi xvid xvid-devel fame libfame0.9-devel transcode libdvdcss
```

Next comes the configuration and compilation. If you don't want to transcode, then the first command in the next example can simply be `./configure`

```
$ ./configure --enable-transcode
$ qmake mythdvd.pro
$ make
$ su
# make install
```

Once that's done, ensure that you've added the appropriate table structures to the database:

```
$ cd dvddb
$ mysql -u root < metadata.sql
```

19. MythNews.

20. Troubleshooting.

20.1 Illegal Instruction.

This error can happen for the following reasons:

- You have an AMD K6–2 or VIA C3 processor. If you have either of these, make sure that you you followed the instructions in this section.
- You have an early production Intel Celeron, Pentium Pro or Pentium II. If you have any of these processors, make sure that your deinterlace variable is set correctly. Your CPU needs to support SSE instructions for deinterlace support. You may check this by doing a `$ cat /proc/cpuinfo` and looking for "sse" in the processor flags section.

20.2 mythfilldatabase failing.

If mythfilldatabase suddenly appears to be failing, check to see what version of XMLTV you're running. Versions prior to 0.5.4 need to have the following change made: edit the `/usr/lib/perl5/site_perl/5.8.0/XMLTV/ZapListings.pm` file and perform a global search and replace, searching for "tvlistings2" and replacing it with "tvlistings". **NOTE**: It is highly recommended to run the latest version of XMLTV available. In North America, zap2it, the listings provider, has been making numerous changes to their website which negatively impact XMLTV.

20.3 MythTV seems to disappear; can't see video.

If you're using Gnome as your window manager, try switching to KDE. For Red Hat Linux 8.0 users, edit `/etc/sysconfig/desktop` and change "GNOME" to "KDE".

20.4 Fast CPU, choppy or jittery video.

First, you should check that your kernel has been enabled for DMA:

```
[mythtv@pvr mythtv]$ dmesg |grep DMA
    ide0: BM-DMA at 0xd800-0xd807, BIOS settings: hda:DMA, hdb:DMA
    ide1: BM-DMA at 0xd808-0xd80f, BIOS settings: hdc:DMA, hdd:pio
hda: 156301488 sectors (80026 MB) w/2048KiB Cache, CHS=9729/255/63, UDMA(33)
hdb: 80043264 sectors (40982 MB) w/2048KiB Cache, CHS=4982/255/63, UDMA(33)
```

From the listing above, you can see that hda, hdb and hdc are set for DMA, and hdd is set for pio. If your kernel is not reporting DMA being enabled, you may need to recompile your kernel. Check your motherboard's chipset (look in the "ATA/IDE/MFM/RLL support" section in "make menuconfig") for more information.

Next, check that the hard drive has DMA enabled. Use the **hdparm** program to check and enable DMA.

```
# hdparm -d /dev/hd?
```

will tell you the DMA status for your hard drives. If you run **hdparm** with the -d1 parameter, it will turn DMA on.

You may also setup your PC to do this at boot time, either by adding the command to your /etc/rc.local file, or by adding files to /etc/sysconfig.

On Mandrake and other distributions, if you install **hdparm** from an RPM you will most likely get a /etc/sysconfig/harddisks file installed. This file will be parsed by the /etc/rc.sysinit script. If you use the default harddisks file, your changes will affect all IDE devices (including CD ROMs). If you wish to use different parameters for various devices, rename and/or copy the file to harddiskhda, harddiskhdb, etc. Edit the file to your liking and on the next reboot your setting will be preserved.

20.5 I keep losing all of my settings every time I compile!

MythTV will overwrite any settings in /usr/local/share/mythtv. If you want to make sure that you don't lose your modifications, make a directory ~/.mythtv and copy your mysql.txt file to this location.

20.6 Compile errors.

Some compile errors are worse than others. If you get an error that doesn't abort the compilation, and says something like:

```
cc1plus: warning: changing search order for system directory
"/usr/local/include"
cc1plus: warning:   as it has already been specified as a non-system
directory
```

then it shouldn't be a problem.

If you get an error like /usr/bin/ld: cannot find -lXext, the compiler is telling you that you don't have XFree86-devel installed, or that your distribution hasn't set it up correctly. This needs to be fixed before MythTV will compile.

20.7 My screen goes blank, but comes back when I wiggle the mouse or use the keyboard.

This is due to DPMS, the Display Power Management System, which is used to save power by turning off your monitor when the system decides that it's not being used or to a screensaver that has defaulted to a blank screen. Since it's likely that watching TV will not generate keyboard or mouse events for a time, you need to turn off DPMS and the screensaver. There are a few ways to do this. You may also need to check your BIOS for power saving modes and disable screen blanking there as well.

Edit your `/etc/X11/XF86Config-4` file, and look for:

```
Section "ServerFlags"
    #DontZap # disable <Crtl><Alt><BS> (server abort)
    #DontZoom # disable <Crtl><Alt><KP_+>/<KP_-> (resolution switching)
    AllowMouseOpenFail # allows the server to start up even if the mouse doesn't work
EndSection
```

Add `Option "NoPM" "true"` and `Option "BlankTime" "0"` to the ServerFlags section.

Also, look for:

```
Section "Device"
    Identifier "device1"
    VendorName "nVidia Corporation"
    BoardName "NVIDIA GeForce 256 (generic)"
    Driver "nv"
    Option "DPMS"
EndSection
```

In this case, you would need to either delete the `Option "DPMS"` line, or change it to `# Option "DPMS"` to comment it out. The next time you start XFree this change will take effect.

Finally, check:

```
Section "Monitor"
    Identifier "monitor1"
    VendorName "Plug'n Play"
    HorizSync 30-85
    VertRefresh 50-160

    # Sony Vaio C1(X,XS,VE,VN)?
    # 1024x480 @ 85.6 Hz, 48 kHz hsync
    ModeLine "1024x480"   65.00 1024 1032 1176 1344   480  488  494  563 -hsync -vsync

    # TV fullscreen mode or DVD fullscreen output.
    # 768x576 @ 79 Hz, 50 kHz hsync
    ModeLine "768x576"    50.00  768  832  846 1000   576  590  595  630

    # 768x576 @ 100 Hz, 61.6 kHz hsync
    ModeLine "768x576"    63.07  768  800  960 1024   576  578  590  616
EndSection
```

Ensure that there isn't an `Option "DPMS"` in the Monitor configuration.

You can also turn off DPMS from the Command Line, but this will not survive a reboot.

```
$ xset -dpms
```

Using `xset +dpms` will turn it back on.

Another technique to try, which will turn off the screensaver:

```
$ xset s off
```

You may also combine the command to turn off DPMS and the screensaver:

```
$ xset -dpms s off
```

Finally, depending on your distribution, you may be able to turn it off from within the control panel.

20.8 I get segfaults / MythTV isn't doing anything.

20.9 Debugging with GDB.

Without details, the developers will not be able to determine if you have discovered a genuine code–bug, or if the problem is with your system. In order to determine what's going on, you must recompile MythTV with debugging support and run MythTV within **gdb**, the GNU debugger.

Edit the settings.pro file. Make sure that the top of the file looks like this:

```
$ cat settings.pro
CONFIG += debug
#CONFIG += release
```

Now, you need to clear out the old versions of the software to ensure that you're running with the debugging code, then compile and install.

```
$ make clean distclean
$ ./configure
$ make
$ su
# make install
# exit
```

At this point, you now have debug–enabled software ready. Let's assume that the problem you're having is in the `setup` program.

```
$ cd setup
$ gdb ./setup

GNU gdb 5.3-1mdk (Mandrake Linux)
Copyright 2002 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain
conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show warranty" for details.
This GDB was configured as "i586-mandrake-linux-gnu"...
(gdb) handle SIGPIPE nostop
Signal        Stop      Print    Pass to program Description
```

```
     SIGPIPE      No       Yes     Yes          Broken pipe
```

gdb has a number of options, read the `man` page for more information.

Once at the `(gdb)` prompt, type `run` to start program execution. When the program segfaults or appears to lock−up (press CTRL−C), type

```
    (gdb) thread apply all bt full
```

The backtrace information for all the threads should be posted to the mythtv−dev mailing list, along with the steps you followed to get the program to crash.

If you're trying to troubleshoot and you can't get back to the gdb window for some reason, it may be easier to use two systems or to start mythfrontend from the text console.

If you're going to troubleshoot from a remote system, connect to the machine that you're going to test using **ssh** or **telnet**. Next, type `$ export DISPLAY=localhost:0.0`. This will allow the graphics to be displayed on the X console (usually ALT−F6 or ALT−F7) and still give you output and control of **mythfrontend**, either from the **ssh** session, or by switching back to the text console by pressing CTRL−ALT−F1. You can now continue troubleshooting using **gdb** as detailed in the instructions.

20.10 MythTV makes my system crash.

MythTV *can not* crash your system − it is a user−level program. If your system is crashing when you run MythTV, then you have some issue with the drivers for your capture card or other hardware, or the CPU fan has fallen off/broken and your system is overheating when asked to perform a CPU intensive task like encoding video.

20.11 Troubleshooting audio.

By Bruce Markey, mailto:bjm@lvcm.com **NOTE**: the following instructions do not apply to PVR−250/350 encoders; the MPEG−2 file will have the audio embedded in the stream so it is not accessible using `/dev/dsp`. See this in the PVR−250 section for more information.

Audio appears to be one of the bigger issues that users run into on the mailing list. If the audio isn't configured correctly, then MythTV will often appear to hang, when in fact it is trying to manipulate the audio subsystem and failing. You may or may not receive error messages indicating that the source of the error is the audio subsystem.

You can not use **xawtv** to determine if your audio is working correctly, since **xawtv** is simply using the the analog sound patched through line−in to line−out. It doesn't need to digitize the sound unless you are using the recording function.

A better test to verify that sound will work for MythTV (and recording with **xawtv** for that matter) is to startup **xawtv**, mute the line−in then run `aplay /dev/dsp`. You should hear the recorded audio slightly delayed behind the realtime video. Once this test succeeds, MythTV should work correctly because it writes to and read from /dev/dsp in the same way that **aplay** does.

To record audio along with video the audio signal must be digitized by a DSP so that the audio data can be stored in a file. On playback, the audio data is written to /dev/dsp and converted back to an analog signal. This

analog signal should then be sent to your speakers. Here is what is needed in alsamixer:

CAPTUR source – the analog source to be sent to the DSP. This should be set to the input source from the tuner card to the sound card. In most cases this is Line but this could also be Aux, CD, Mic, etc., depending on how you connect the input cable. This source should be muted to prevent patching through the analog sound. The volume of this source will not affect the record level.

Capture mixer – this sets the level for the analog to digital recording. While a volume of 100% is recommended for testing, distortion may occur. Lowering this level to 75% to 85% may result in better audio quality. "Capture" should be marked as the CAPTUR destination.

PCM mixer – this sets the level for the digital to analog playback. While a volume of 100% is recommended for testing, distortion may occur. Lowering this level to 75% to 85% may result in better audio quality.

Master mixer – sets the level for the analog signal sent to line–out or the speakers.

You may also want to ensure that `/dev/dsp` hasn't already been grabbed by another process, like **esd** or **artsd**. If `/dev/dsp` isn't available, then MythTV won't work. Check your window manager documentation for instructions on disabling the sound manager. If you wish to see what application is grabbing a resource, you can use the `fuser` command:

```
# fuser -v /dev/dsp
```

To disable aRts in KDE, go to KDE–>Control Center–>Sound–>Sound System and uncheck the "Start aRts soundserver on KDE startup" box. Run # `killall artsd` from the command line to stop the artsd program.

If you're using multiple sound cards and multiple tuners, use `alsamixer -c 1` to work with the second sound card. The first card is #0, the second card is #1, etc.

20.12 I can't see parts of the On Screen Display.

If you're a Red Hat 8.0 user, make sure that you've followed the instructions which tell you how to modify the Themes makefile. Link.

20.13 The mythbackend program told me to look at this section.

mythbackend does a check to see if your sound device is capable of full duplex operation. If it's not, it's most likely that you're going to run into issues when you try to record and play sound at the same time. If your backend is a separate machine than your frontend, then there's no problem, since you're only going to be doing one thing at a time with the card. Likewise, if you're running the frontend and backend on the same machine, but you're using btaudio or a Hauppauge PVR–250 as your recording source, and using the playback function of your soundcard, then you also shouldn't have an issue, since the sound card isn't being asked to perform two functions at once.

If you can't get your soundcard to go full duplex and need it to, then check your distribution for updated sound drivers. If your soundcard is not capable of full–duplex operation, either because the drivers don't support it, or it has been designed that way, then you're pretty much out of luck and will either need to purchase a new

sound card, or will need to get btaudio operational.

20.14 My remote doesn't work / works sometimes and not others / "ghost" keypresses.

This can be due to a number of factors. The simplest case is the "ghost" keypresses. For me, it was due to compact fluorescent lights in the same room as the IR receiver, which the receiver was picking up as keypresses. Once the lights were switched to incandescent bulbs, the ghost went away.

You may have an issue with **lirc** misinterpreting IR commands from a different remote. I also have an issue where the Tivo "Peanut" remote will eventually cause **lircd** to stop responding; even though **lircd** is configured for the Pinnacle Systems remote, the Tivo remote IR patterns are being seen by the IR receiver.

If your remote has been properly configured, and **irw** and **irxevent** are working correctly, then it's highly likely that your window manager is not giving focus correctly to the various Myth programs as they run. The following window managers are known to work correctly:

- fvwm
- blackbox (using "Sloppy Focus" and "Focus New Windows")

20.15 My PVR−250 card doesn't show inputs in the setup screen.

See message http://www.mythtv.org/pipermail/mythtv−users/2003−April/002527.html

20.16 Where's my "canada−cable" entry gone to?

20.17 My channels are off by one.

There is no such thing as "Canada Cable"; Canada uses the same frequencies as the United States. "Canada Cable" was a hack that some people used when they would discover that their channels were off−by−one, ie, when tuning to channel 42, they might get channel 41 or 43. This is actually due to the tuner on the video capture device being mis−detected. You must manually specify the tuner type in your `/etc/modules.conf`. See the video4linux mailing list (https://listman.redhat.com/mailman/listinfo/video4linux−list) for more information.

20.18 Mythweb is showing a db_open error when I connect to it.

Find your php.ini file. Make sure you've got a line in it like this:

```
extension=mysql.so
```

Restart apache for it to take effect.

20.19 error: can't find a register in class 'BREG' while reloading 'asm'

This is due to the broken Qt that is being distributed by Suse. To work around this, edit `libs/libavcodec/Makefile` and remove any "–fPIC" you find there and then recompile.

20.20 make: *** No rule to make target /usr/lib/qt3/mkspecs/default/qmake.conf', needed by Makefile'. Stop.

This error happens when there's a missing link in the `/usr/lib/qt3/mkspecs` directory. There are two ways to fix this error:

1. Create the link manually:

```
$ su
# cd /usr/lib/qt3/mkspecs
# ln -sf linux-g++ default
```

and then restart the compile,

or

2. Run **qmake mythtv.pro** in the mythtv directory. Rerunning **qmake** will create a new Makefile for you, however this still doesn't fix the root cause of the issue, which is that your distribution didn't create the symlink for you when the qt3 package was installed. The first choice is the better solution.

20.21 make: *** No rule to make target /mkspecs/default/qmake.conf', needed by Makefile'. Stop.

You didn't set your `QTDIR`. Re–read the section on Setting up paths.

20.22 My mouse pointer disappears when placed over the MythTV windows.

This is the intended behavior. The MythTV interface is meant for use with a remote control or a keyboard.

20.23 What does "strange error flushing buffer" mean on my console?

Nothing, really. It's just lame (the mp3 encoder) complaining for some obscure reason. This seems to be fixed in more recent versions of the libmp3lame library.

20.24 I can't change the channel when watching Live TV.

Something's wrong with your program database. Did mythfilldatabase run with no major errors?

20.25 When trying to run setup, I get an error like this: "./setup: error while loading shared libraries:"

You didn't add `/usr/local/lib` to `/etc/ld.so.conf`. See the section on modifying /etc/ld.so.conf.

20.26 Troubleshooting the Hauppauge PVR−250 IR remote

The most likely issue is that you have incompatible versions of the **lirc_i2c** and **lirc_dev** modules and the various **lircd** programs. See the section called Hauppauge PVR−250 remote and MythTV's native LIRC support. for examples on finding and removing old versions of lirc.

20.27 MySQL not connecting correctly.

Your **MySQL** installation may have networking turned off. Check that `/etc/mysql/my.cnf` *does not* contain `skip-networking`. If it does, remove it, and restart **MySQL**.

20.28 My screen goes black when I try to play something.

MythTV prints error and status messages to the shell that was used to start the application. If nothing seems to be happening when you try to view a program, try switching back to the shell and look for error messages there.

21. Miscellaneous.

21.1 I'd like to watch the files without using MythTV / I'd like to convert the files to some other format.

MythTV comes with a utility called **mythtranscode** which can decode nuv files into raw format for use with other applications. This command−line utility was not designed to be used by the end−user, but instead to be called by other applications or scripts. Programs like **mkmovie** (http://www.icelus.org/) and **nuvexport** (http://forevermore.net/myth/) are better suited for the end user. However, since **mythtranscode** can be a useful tool, directions on using it follow.

mythtranscode creates raw streams, which means that they do not contain any container information such as resolution, frame−rate, or audio sampling rate. In order to process the output, you must supply this information to the processing utility. **mythtranscode** provides the relevant information on stdout.

There are two modes in which **mythtranscode** can create raw streams. The first has no synchronization and assumes that the processing utility will read audio and video at a constant rate. This method is useful when a single application will be processing the raw output, such as **mencoder** or **ffmpeg**. The second method assumes that two separate applications will be processing the audio and video streams independently, and there is no rate control between them which means that the two programs don't coordinate their efforts to

maintain synchronization.

mythtranscode example

First, start **mythtranscode**. You will need to determine the channel and the start time manually.

```
$ mythtranscode --chanid 1036 --starttime 2003-10-20T15:30:00 --profile \
autodetect --fifodir . &
```

When **mythtranscode** begins executing, it will create two fifos ("audout" and "vidout") in the directory specified (in this case ".", meaning the current directory) and will print out information about the video stream.

The next step is to start the processing application. The following assumes that the stream is NTSC 640x480 with 32kbps audio.

To use **mencoder** you would enter a command like:

```
mencoder -audiofile audout -audio-demuxer 20 -rawaudio rate=32000 \
-rawvideo on:w=640:h=480:fps=29.97 -ovc lavc -oac mp3lame -o out.avi \
vidout
```

NOTE: You must use mencoder 1.0PRE1 or later. **mencoder** version 0.9x *WILL NOT WORK!*

Using ffmepg:

```
ffmpeg -f u16le -ar 32000 -ac 2 -i audout -f rawvideo -s 640x480 -r 29.97 \
-i vidout -vcodec mpeg4 -b 2000 -acodec mp3 -ab 128 out.avi
```

Or to play directly using mplayer (again 1.0PRE1 or later is needed):

```
mplayer -audiofile audout -audio-demuxer 20 -rawaudio rate=32000 \
-rawvideo on:w=640:h=480:fps=29.97 vidout
```

If you wanted to write the raw data to separate audio and video files for later processing, the following would work (note the use of --fifosync for rate-control):

```
$ mythtranscode --chanid 1036 --starttime 2003-10-20T15:30:00 \
--profile autodetect --fifodir . --fifosync &
$ cat audout > audio.raw &
$ cat vidout > video.yuv
```

21.2 I'd like to use some other window manager.

MythTV is not dependent on any particular window manager. If you wish to run a lightweight window manager, the `configfiles/` directory has an example of a `.twmrc` and `.fvwmrc` file you may use.

21.3 What capture resolution should I use? How does video work?

While MythTV allows you to set various GUI and capture resolutions, not all combinations make sense.

First, analog video signals have a defined vertical resolution. In NTSC, the video standard specifies that there are 525 vertical scan lines. Once the "extra" lines are removed (they're used to synchronize the video signal, and encode closed captioning data), you have 480 vertical lines.

In PAL, there are 625 "raw" lines of resolution, with a net of 576 vertical lines.

Horizontally, the maximum you can expect from a Bt8X8 chip is 720.

With this in mind, there are certain commonly accepted values for resolution. While other values may be accepted, they will cause scan lines to be repeated or dropped.

From "best" to "worst", in NTSC:

- 720x480 (but more commonly, 704x480. "DVD" resolution)
- 640x480 4:3
- 544x480 (TiVo high resolution)
- 480x480 (Video CD resolution)
- 352x480 (ReplayTV "Standard" quality)
- 320x480
- 544x240
- 480x240
- 352x240
- 320x240

As you can see, the lower quality values are half of the better ones. 720x240 is possible, but isn't a good tradeoff relative to the number of vertical lines lost. In a PAL country, the you would use values like 720x576 or x288.

The higher resolutions will be more CPU intensive if you're using software encoding (PVR−250/350 will have minimal host CPU impact even if you're using 720x480). If the CPU is overtaxed, frames will be dropped causing uneven motion. You will likely see the best results at resolutions which average at least 10% CPU idle time. You can use system tools such as `top` or `sar` to check the CPU % idle while recording. If the CPU average usage is consistently exceeding 90%, frames will need to be dropped during peak times when more than 100% of the available CPU would be needed to process all of the frames.

If you'd like to read more on this, go to the vcdhelp website at http://www.vcdhelp.com/forum/userguides/94382.php.

21.4 MythTV GUI and X Display Sizes.

MythTV is designed to be run as dedicated full screen TV application but can also be run as a desktop application on a computer monitor. Here are a few consideration for configuring sizes to best suit you needs.

X Dimensions

For output to a Television, common resolutions are 640x480, 800x600, and some rare devices support 1024x768. Generally, higher resolutions are better. However, you may find that you prefer the picture quality at one of the lower resolutions. Everything in MythTV is scalable and should 'fit' regardless of the resolution you choose.

Edit your X configuration file, usually "XF86Config−4", so that the the resolution you want to use is listed first in the lists under "Screen". If this resolution is higher than the resolutions supported by your output device, you will see a 'panning' effect where moving the mouse to the edge will scroll around a desktop area which is larger than the display size. If this happens, edit your X configuration file to match the display size then restart X.

MythTV Dimensions

In 0.10 and later, from "mythfrontend" go to Setup−>Appearance. The default for the height and width is "0" − this will cause MythTV to automatically size itself to full screen.

If the MythTV GUI width and height are not 0, mythfrontend uses these GUI dimensions and is anchored to the upper left corner of the X Desktop. If the GUI X and/or Y are not 0, the upper left corner is positioned at the specified coordinates. If the "Run the frontend in a window" box is checked, the window will have a frame and can then be dragged to any position on the desktop.

NOTE: When the GUI is full screen, you may see windows rapidly flipping on top of each other. If this happens you will need to set your window manager to 'Click to Focus' for windows to stack properly.

The fonts for the GUI and OSD will scale to whatever sizes you use. Most font sizes can be changed in setup selections or in the .xml files under `/usr/local/share/mythtv/`. Make sure to use fonts large enough to be read on a TV screen from a distance.

The full screen TV size is based on the X display size. For Xinerama, you can specify a screen in Setup−>General. The TV picture will be stretched to fit the entire GUI area regardless of the capture resolutions used. However, during playback, the "W" key can to used to correct differences between 16:9 and 4:3.

Overscan Dimensions

Because picture edges can be ragged and screen edges aren't straight, Television is designed to project an image larger than the physical screen. This is called "overscan". Underscan is fitting the entire image inside the screen. Underscan is useful for computer monitors so that toolbars and scrollbars at the edges can be seen.

For best results, match the X display area as close as possible to the edges of the physical screen. This can only be adjusted by your tv−out device or by the settings for the television set. Many sets have these adjustments in a 'service mode'. If you cannot make these adjustments, there will be black borders around the edges of the X desktop, MythTV GUI and TV playback.

MythTV has settings for "Overscan" in Setup−>Playback. These can not, and do not, cause the image to display beyond the edge of the X display area. The purpose of these settings are to cut off rough edges and to expand the image so that objects will appear to be the same size as a normal overscanned TV picture.

21.5 I'd like to save or restore my database.

See the **mysqldump** manpage for more information.

```
$ mysqldump −u mythtv −pmythtv mythconverg −c > mythtv_backup.sql
```

To restore: (assuming that you've dropped the database)

```
$ mysql -u root
mysql>create database mythconverg;
mysql>exit
$ mysql -u mythtv -pmythtv mythconverg < mythtv_backup.sql
```

You may need to alter the mysql permissions if this database is being shared with multiple systems. See the Modifying access to the MySQL database for multiple systems section for more information.

21.6 I'd like to delete the mysql database.

NOTE: Performing this step will remove the entire database. You will lose all of your settings and will need to re–run the mc.sql script to setup the database structure before running the setup program.

```
$ mysql -u root
mysql> drop database mythconverg;
mysql> quit
```

21.7 btaudio

btaudio allows you to obtain the audio data from your tuner card directly over the PCI bus without using a soundcard. This is useful if you would like to use multiple tuner cards in a system without adding a soundcard for each one, or if your existing soundcard is not capable of full–duplex operation.

In order to use btaudio, your tuner card will need certain hardware installed on it, and that hardware must be wired correctly. The chip that will allow you to use the btaudio module is the MSP34xx. However, having a MSP34xx is no guarantee that you will be able to use the btaudio module.

As of 2003–03–31, this is the current list of cards and their status: Works with btaudio:

- Hauppauge WinTV–radio with dbx–TV stereo, model 401
- Hauppauge WinTV–Theater, model 495, 498 (Europe)
- ATI TV Wonder

The following cards do not work:

- Pinnacle Studio PCTV Pro – note: this has a MSP34xx, but it's not wired correctly to the BT878 chip.
- ATI TV Wonder VE
- Leadtek Winfast 2000 XP (PAL, UK and NTSC)
- I/O Magic PC–PVR. No MSP34xx chip.

The following cards have been reported to work, but have issues:

- Avermedia AVerTV Studio (no digital DSP output, "whiney noise" on analog)

See `contrib/example.modules.conf.for.btaudio` for an example file on how to configure btaudio on your system.

Once btaudio loads, it should register additional `/dev/dsp` and `/dev/mixer` devices. Typing `$ dmesg` will let you know what's going on.

21.8 Removing unwanted channels.

If **mythfilldatabase** grabbed a channel which you do not want to include in your TV listings, you can remove the entries from the xmltv config file and the mysql database. This often happens with premium channels; for example, HBO or Showtime may be available on your cable TV system, but is scrambled because you're not a subscriber to that channel. Since you can never watch it, you want to get rid of it.

First, comment out the channel from the /.mythtv/<sourcename>.xmltv file by inserting a "#" in front of the unwanted entry. This will prevent **mythfilldatabase** from grabbing future listings.

Next, delete the unwanted item from the channel table so that it will not appear in the EPG or when changing channels. The channels will always have "1000" added to them, so for example, assuming that HBO is channel 15:

```
$ mysql -u root mythconverg
mysql> delete from channel where chanid = '1015';
```

Old program data will be removed over the course of a week. However, you may want to immediately delete any current program listings for the channel that has been removed:

```
$ mysql -u root mythconverg
mysql> delete from program where chanid = '1015';
```

21.9 I'd like to use NFS.

You may want to use a central server to store your files.

On the host machine, (in this case, the hostname is "masterbackend") you'll want to edit your /etc/exports file and use something like:

```
/var/video (rw)
```

To export the /var/video directory with read / write privileges.

On the "slave" machine, you'll want to edit the /etc/fstab file and add something like:

```
masterbackend:/var/video /var/video nfs rsize=8192,wsize=8192,soft,nfsvers=3
```

Then run # fstab -a to re-read the file to mount the file system.

In this case, the source is a machine called "masterbackend" which is exporting the directory "/var/video", which we're mounting locally at "/var/video". The rsize and wsize options are used to increase the performance of NFS; soft mean that NFS will eventually timeout on an error, and the nfsvers is required for filesizes over 2GB.

21.10 Automatically starting mythfrontend at system boot time.

Here's an example submitted to the mythtv-dev list by Pat Pflaum mailto:pat@netburp.com using fvwm:

```
$ cat > .xinitrc
fvwm &
mythfrontend
^D
$ cat > .fvwmrc
Style myth* NoTitle, NoHandles, Sticky, WindowListSkip, SloppyFocus, GrabFocus, BorderWidth
^D
$
```

The following also works with blackbox:

```
$ cat > .xinitrc
xset -dpms s off &
irxevent &
mythfrontend &
blackbox
```

Make sure that your `.blackboxrc` file has:

```
session.screen0.focusNewWindows:        True
session.screen0.focusModel:      SloppyFocus
```

in it.

21.11 Automatically starting mythbackend at system boot time.

Red Hat And Mandrake

Here's a method for automatically starting mythbackend submitted by Mike Thomson (mailto:linux@m–thomson.net) and Stu Tomlinson (mailto:stu@nosnilmot.com).

Copy the files from the MythTV `contrib` directory or from Mike's web site (http://m–thomson.net/mythtv/) as follows:

`etc.rc.d.init.mythbackend` should be made executable and copied to `/etc/rc.d/init.d/`:

```
$ cd contrib
$ su
# chmod a+x etc.rc.d.init.d.mythbackend
# cp etc.rc.d.init.d.mythbackend /etc/rc.d/init.d/mythbackend
```

`etc.sysconfig.mythbackend` should be copied to `/etc/sysconfig/`:

```
$ cd contrib
$ su
# cp etc.sysconfig.mythbackend /etc/sysconfig/mythbackend
```

Edit `/etc/sysconfig/mythbackend` if you want to change the defaults (the userid that should start mythbackend, location of the logfile and (if required) the name and location of the mythbackend binary).

Use **chkconfig** to make sure the script is called when entering runlevels 3, 4 or 5:

```
$ su
# chkconfig --level 345 mythbackend on
# exit
$
```

Log files

By default, the log file for mythbackend will be written to `/var/tmp/mythbackend.log`. This has been tested and is known to work on Mandrake and Red Hat, but many people prefer to place logs under `/var/log/`.

To do this, create a group called `mythtv` (or anything you prefer) and add your usual MythTV users to that group. If you changed the user that starts mythbackend from the default of root you *must* perform this step.

Create the directory `/var/log/mythtv` and set its permissions as follows:

```
$ su
# mkdir /var/log/mythtv
# chown root:mythtv /var/log/mythtv
# chmod 0775 /var/log/mythtv
# exit
$ ls -ld /var/log/mythtv
drwxrwxr-x   2 root     mythtv        4096 Apr 28 21:58 /var/log/mythtv/
$
```

Mandrake 9.x

Mandrake adds one more twist in the form of the `msec` utility, which runs regularly and (at the default or any higher security level) sets permissions on many files, including those under `/var/log`.

To tell msec about the MythTV log files and their directory, you need to edit the `/etc/security/msec/perm.local` file to include the following:

```
# /etc/security/msec/perm.local
# Local overrides to the msec program
#
# Full file path              user.group          permissions
/var/log/mythtv/              root.mythtv         775
/var/log/mythtv/*             root.mythtv         664
```

A copy of the above has been included in the contrib/ directory. You may add it by typing:

```
$ cd contrib
$ su
# cat etc.security.msec.perm.local >> /etc/security/msec/perm.local
# exit
```

Finally run the `msec` tool to check and implement your changes.

```
$ su
# msec
# exit
$
```

NOTE: msec can only *reduce* the permissions of files, so if you don't get the results you expect, check that you're not asking `msec` to add missing permissions to the files or directories you created.

Gentoo

The portage file for MythTV has scripts that will allow you to run mythbackend at startup.

To run mythbackend as a daemon which starts at boot time:

```
# rc-update add mythbackend default
```

To stop mythbackend as a daemon:

```
# /etc/init.d/mythbackend stop
```

To obtain a list of options:

```
# /etc/init.d/mythbackend
```

21.12 Advanced Backend Configurations.

MythTV is flexible in the way that you define multiple backend tuner configurations. The only hard–and–fast rule is that the Master Backend *must* have a capture device defined, but shouldn't imply that the capture device in the MBE must be the first capture card defined in the database.

One example of an advanced configuration is the round–robin scheme. Rather than defining all of the cards on the master, you could first go into setup on the master to define globals such as the general configuration and the channel lineup but not the host–specific configuration item like the capture card. In this example, we will use a 4 tuner configuration, where two slaves have one card each and the master has two.

1. Add the first capture card on one of the slaves. Complete the configuration, connecting the input source to the card. This will get cardid #1 in the database. Exit setup.
2. Configure the first capture card on the master backend. This will get cardid #2 in the database. Exit setup.
3. Configure the first capture card on the second slave. This will be cardid #3 in the database. Exit setup.
4. Configure the second capture card on the master backend. This will get cardid #4 in the database. Exit setup.

Using this scheme, the master backend will not use both capture cards until one of the following happens:

- There are four recordings scheduled for the same time
- Both slaves are not available

The scheduler in MythTV checks whether an encoder is available; if a slave backend isn't running, its encoder isn't available, so the scheduler will look for the next available encoder. This makes MythTV very flexible; slave tuners can come and go, and as long as there are enough tuners for what you'd like to record it doesn't matter which tuner in particular is going to be used.

Using this round–robin scheme along with a shared storage directory like NFS and enabling the Master Backend Override setting will allow you to view content even if the slave backend that recorded a program is

not available.

21.13 What is this transcoder thing, and how do I use it?

The transcoder re−encodes files from one MythTV format to another. The main purpose of the transcoder is to allow users with hardware encoders (PVR−250) or systems that can only record in RTJpeg due to performance reasons (multiple capture cards, slow system, etc) to create MPEG−4 streams to save space.

The transcoder can be used in two ways:

- Automatically re−encode every file once it has completed recording
- After marking commercials, the transcoder can be run manually to delete the commercials from the file (thus further saving space)

The two methods are independent. The first is an automatic process that can be enabled/disabled through the setup program; the second is a manual method which must be invoked on each recording individually.

The second method can be used on files that have already been transcoded (or files which were are already in the correct format), so only the frames immediately following a cut section will be re−encoded, resulting in a very minimal loss of quality.

To enable automatic transcoding, do the following: start the setup program under the host−specific settings: set the Transcoder Auto−run checkbox

For either manual or automatic transcoding: start mythbackend start mythfrontend select setup, and the Transcoding recording profile. now select either RTJPEG or MPEG−4 (selecting any of the hardware encoders will result in the transcoder not working). set the other parameters as you'd like. For best performance, you should match the audio to the 'Default' profile. If you plan to only use the manual transcoder, you can choose to set the video to be the same as the default profile too (as long as it is MPEG−4 or RTJpeg)

Everything should now be setup properly. If you elected to use the Auto−Run feature, the transcoder will automatically launch after each recording is complete. The transcoder thread runs at a low priority, so it should not impact any critical tasks or other recordings.

If you want to manually transcode a program, simply press 'X' while watching a recording (you should have already finished marking all commercials). If you change your mind, hitting 'X' again will stop the transcoding.

Once the transcode is complete, mythbackend will replace the old file with the new as soon as it is no longer in use.

22. Example Configurations.

This section contains configurations which you may find useful.

22.1 Hauppauge PVR−250/350 hardware MPEG−2 encoder.

NOTE: THE FOLLOWING DOCUMENTATION IS PROVIDED AS–IS AND IS NOT GUARANTEED TO WORK WITH YOUR DISTRIBUTION. FOR IVTV DRIVER ASSISTANCE, SEE THE IVTV MAILING LIST. https://lists.sourceforge.net/lists/listinfo/ivtv–devel SEE THE IVTV FAQ IF YOU HAVE QUESTIONS OR ISSUES. http://ivtv.sourceforge.net/FAQ.html

These instructions work for Mandrake 9.1 on a backend machine which is hosting a single PVR–250 capture card.

Install the Mandrake kernel source. If you are using a different Mandrake kernel level (ie, not 2.4.21–0.16mdk as in the example below, then alter the **urpmi** command appropriately.)

```
$ uname -a
Linux pvr 2.4.21-0.16mdk #1 Fri Apr 11 06:51:54 CEST 2003 i686 unknown
unknown GNU/Linux

$ su
# urpmi kernel-source
```

NOTE: You do not need to recompile your kernel.

Download the IVTV driver from CVS:

```
$ cvs -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/ivtv login
<enter> as password
$ cvs -z3 -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/ivtv co ivtv
```

This should create an ivtv directory. Switch to it and perform the following commands:

```
$ cd ivtv/ivtv/utils
$ wget http://hauppauge.lightpath.net/software/pvr250/pvr250_17_21226.exe
$ su
# ./ivtvfwextract.pl pvr250_17_21226.exe
# exit
```

The **ivtvfwextract** program extracts the firmware required for the card. You may want to go to the Hauppauge website and download the latest Windows driver if the **wget** command fails.

Next, compile the driver:

```
$ cd ~/ivtv/ivtv/driver
$ make
```

Compile the **test_ioctl** program:

```
$ cd ../utils
$ make
$ su
# cp test_ioctl /usr/local/bin
# exit
```

NOTE: You may get warnings about i2c during the compile. Ignore them.

Add the following to /etc/modules.conf:

```
alias char-major-81     videodev
```

21.13 What is this transcoder thing, and how do I use it?   75

```
alias char-major-81-0   ivtv
alias char-major-61 lirc_i2c
options ivtv debug=1
options tuner type=2
options msp3400 once=1 simple=1
add below ivtv msp3400 saa7115 tuner
add above ivtv lirc_dev lirc_i2c
```

NOTE: You must specify the tuner type manually. See `~/ivtv/ivtv/driver/tuner.h` for a listing of tuners. In the above example, `tuner type=2` is a Phillips NTSC. Use an appropriate tuner type for your system and video standard. The `debug=1` parameter on the ivtv line actually means less debugging output will be printed.

Next, copy the files that were compiled in the driver directory into the appropriate modules directory:

```
$ su
# make install
# depmod -a
```

Finally, load the ivtv driver:

```
# modprobe ivtv
# exit
```

Check that the card is being recognized on the PCI bus:

```
$ lspci -v
<snip>

00:0c.0 Multimedia video controller: Internext Compression Inc iTVC15 MPEG-2
Encoder (rev 01)
        Subsystem: Hauppauge computer works Inc.: Unknown device 4801
        Flags: bus master, medium devsel, latency 32, IRQ 11
        Memory at d0000000 (32-bit, prefetchable) [size=64M]
        Capabilities: <available only to root>
```

If you don't see your card on the PCI bus, make sure that it's been installed in a bus–master slot. For some motherboards, only the PCI slots closest to the AGP slot are bus–mastering.

Check that the ivtv driver has created a new video device:

```
$ ls -l /dev/v4l/vi*
crw-------    1 mythtv   sys          81, 224 Dec 31  1969 vbi0
crw-------    1 mythtv   sys          81,   0 Dec 31  1969 video0
crw-------    1 mythtv   sys          81,  32 Dec 31  1969 video32
```

In this case, the ivtv device is `video0`.

Go into `setup/setup` on the machine hosting the hardware MPEG–2 card. When you add a new capture card, ensure that you are using the `video0` device.

NOTE: As of 2003–04–25 the ivtv driver does not support VBI, so closed–captioning is not available.

Leave the VBI device, audio device and audio sampling rate limit at the default values. Change the default input to an appropriate setting for your configuration. Change the card type to "Hardware MPEG Encoder

Card" and press ENTER.

The remaining setup is just like a standard V4L card; define a video source if you have not already done so, and assign the video source to a video card input on the Input Connections screen.

Some people report issues with the ivtv module not being loaded when required by MythTV. To get around this, you can **modprobe** the ivtv driver before using it. The simplest technique is to add the **modprobe** command to /etc/rc.d/rc.local:

```
# echo "/sbin/modprobe ivtv" >> /etc/rc.d/rc.local
```

NOTE: some people complain that their PVR–250/350 card is not capturing audio. The most likely source for this is that your system is not loading the correct msp3400.o file; ivtv uses a custom msp3400.c, and if the resulting module isn't loaded you will not get sound. Switch to your /lib/modules directory and rename the existing msp3400.o file, then copy the msp3400.o from the ivtv driver directory and re–run **depmod**. If that still doesn't work, then check the ivtv mailing list for more information. You may wish to perform an **updatedb** as root and then do a locate msp3400 to find all the msp3400.o.gz files that may be on your system. For example, Mandrake has a msp3400.o.gz in the 3rdparty subdirectory which will take precedence over the custom version created by ivtv. You can check if you have the correct msp3400 loaded by typing (as root): modinfo msp3400 | grep "parm: *standard". If you don't get a line that says: "parm: standard int", then you're not loading the correct msp3400.

Check to ensure that you're loading the correct msp3400 module.

```
# modinfo msp3400
filename:     /lib/modules/2.4.22-9mdk/kernel/drivers/media/video/msp3400.o
```

Check that the path is correct for your kernel version, and that the driver being loaded is in /media/video and not some other directory.

NOTE: Some people report "ghosting" issues with their PVR–250s, where there is a faint copy of whatever is on the screen appearing to the right of the image. If you have this issue, it's fairly obvious. This is a firmware issue, apparently. There are two ways around this:

1. Run at 720x480 resolution
2. Turn off DNR

To turn off DNR, add the following to your rc.local:

```
/path/to/ivtv/utils/test_ioctl --set-codec-params=dnr_mode=0,dnr_temporal=0
```

22.2 Hauppauge PVR–250 remote and MythTV's native LIRC support.

NOTE: THE FOLLOWING DOCUMENTATION IS PROVIDED AS–IS AND IS NOT GUARANTEED TO WORK WITH YOUR DISTRIBUTION. FOR LIRC DRIVER ASSISTANCE, SEE THE LIRC MAILING LIST. http://lists.sourceforge.net/lists/listinfo/lirc–list

These instructions work for Mandrake 9.2 on a frontend machine which is hosting a single PVR–250 capture card using MythTV's native LIRC support.

First, you need to ensure that you've installed the kernel source appropriate for the kernel that you're running. In this case, we're using kernel 2.4.22−9, so when installing the kernel−source ensure that you've got the correct version.

```
$ su
# uname −a
Linux frontend 2.4.22-9mdk #1 sam sep 13 06:54:11 CEST 2003 i686 unknown
unknown GNU/Linux
# rpm −qa|grep kernel−source
kernel-source-2.4.22-9mdk
```

If you've already installed LIRC from the Mandrake packages, it's going to be version 0.6.6, and that won't work. Uninstall it:

```
$ su
# urpme lirc lirc−remotes liblirc−devel
# rm −rf /dev/lirc*
# exit
```

Check to ensure that you don't have other versions of lirc modules and programs on your system:

```
$ su
# updatedb
# locate lirc_i2c|grep `uname −r`
/usr/src/linux-2.4.22-9mdk/3rdparty/lirc/lirc_i2c.c
/lib/modules/2.4.22-9mdk/kernel/3rdparty/lirc/lirc_i2c.o.gz

# locate lirc_dev|grep `uname −r`
/usr/src/linux-2.4.22-9mdk/3rdparty/lirc/lirc_dev.c
/usr/src/linux-2.4.22-9mdk/3rdparty/lirc/lirc_dev.h
/usr/src/linux-2.4.22-9mdk/include/linux/modules/lirc_dev.stamp
/usr/src/linux-2.4.22-9mdk/include/linux/modules/lirc_dev.ver
/lib/modules/2.4.22-9mdk/kernel/3rdparty/lirc/lirc_dev.o.gz

# locate irw|grep "/usr"
/usr/bin/irw
/usr/local/bin/irw
/usr/local/man/man1/irw.1

# locate lircd|grep "/usr"
/usr/src/linux-2.4.22-9mdk/3rdparty/lirc/lircd.conf.RM-050
/usr/share/man/man8/lircd.8.bz2
/usr/share/doc/lirc-0.6.6/html/lircd.html
/usr/sbin/lircd
/usr/local/sbin/lircd
/usr/local/man/man8/lircd.8

# locate irxevent|grep "/usr"
/usr/share/man/man1/irxevent.1.bz2
/usr/share/doc/lirc-0.6.6/html/irxevent.html
/usr/share/doc/lirc-0.6.6/irxevent.keys
/usr/bin/irxevent
/usr/local/bin/irxevent
/usr/local/man/man1/irxevent.1
```

As you can see, there were a number of different versions of lirc already installed on this system. They must be removed. We don't care about the source code or the man pages, only the executables and the kernel modules.

22.2 Hauppauge PVR−250 remote and MythTV's native LIRC support. 78

```
$ su
# rm -rf /lib/modules/2.4.22-9mdk/kernel/3rdparty/lirc
# rm -rf /usr/sbin/lircd /usr/local/sbin/lircd
# rm -rf /usr/bin/irxevent /usr/local/bin/irxevent
# rm -rf /usr/bin/irw /usr/local/bin/irw
```

Download the lirc−0.7.0pre2 tarball:

```
$ wget http://lirc.sourceforge.net/software/snapshots/lirc-0.7.0pre2.tar.bz2
$ tar -xjf lirc-0.7.0pre2.tar.bz2
```

Mandrake now includes i2c version 2.8.0 which changes the semantics of some low−level calls. You will need to patch LIRC in order for it to work correctly. Download the patch and install:

```
$ cd lirc-0.7.0pre2
$ wget http://delvare.free.fr/i2c/other/lirc-CVS-i2c-2.8.0.patch
$ patch -p1 < lirc-CVS-i2c-2.8.0.patch
```

Next, run the LIRC setup program:

```
$ ./setup.sh
```

Select "1" (Driver configuration), then scroll down to "5" (TV Card), press ENTER, then scroll down to "f" (Hauppauge TV card) and press ENTER again. Once back at the main menu, press "3" for Save and run configure.

If your build environment is ready, LIRC will create a Makefile for you. Switch to root and compile:

```
$ su
# make
# make install
```

As a part of the `make install` process, LIRC will create the appropriate device for you in `/dev`. Check to make sure:

```
$ ls -l /dev/lirc*
crw-r--r--   1 root     root       61,   0 Sep 18 15:36 /dev/lirc
srw-rw-rw-   1 root     root             0 Sep 18 15:38 /dev/lircd=
```

Your `modules.conf` file should have already been modified if you've followed the instructions above for installing the PVR−250. **modprobe** the lirc_i2c driver, run the **lircd** program and then check your `/var/log/messages` and `/var/log/lircd` files.

```
$ su
# modprobe lirc_i2c
# lsmod
Module                  Size  Used by     Not tainted
lirc_i2c                5124   0
lirc_dev               10096   1  [lirc_i2c]
...
# lircd
# tail /var/log/messages
Sep 18 15:38:26 frontend kernel: lirc_i2c: chip found @ 0x18 (Hauppauge IR)
Sep 18 15:38:26 frontend kernel: lirc_dev: lirc_register_plugin:sample_rate:
10
...
```

22.2 Hauppauge PVR−250 remote and MythTV's native LIRC support.

```
# tail /var/log/lircd
Sep 18 15:24:52 frontend lircd 0.7.0pre2: lircd(hauppauge) ready
```

To check that your remote is working correctly, run the **irw** program and start pressing buttons. If nothing is happening, you must begin troubleshooting. Things to check:

- Are there batteries in the remote?
- Does the remote work? You may be able to see the flashes of IR coming from the remote if you look at it with a video camera.
- Is the IR dongle cable plugged in?
- Did you remove all of the old LIRC device drivers? Run "updatedb" as root, then "locate lirc_i2c". The only lirc_i2c should be the one in the misc/ directory for your kernel version.
- Did you run depmod?
- Is your modules.conf setup correctly?
- Did you modprobe lirc_i2c?
- Did you run lircd?
- Do you have an /etc/lircd.conf?
- Is /dev/lirc a character mode file, major 61, minor 0? Is it readable by all groups?
- If you think you may have had old LIRC device drivers you may need to reboot to ensure that they're totally out of the system. Try running modprobe –r lirc_i2c as root first, then modprobe lirc_i2c and try again.
- What path does modinfo lirc_i2c and modinfo lirc_dev show? It should be in /lib/modules/{your kernel version}/misc/

If none of the above works, and **irw** still isn't showing keypresses, then you will need to ask your question on the LIRC list.

Assuming that **irw** is showing keypresses, you can continue: To enable native LIRC support within MythTV, you will need to modify your `settings.pro` file and recompile. Your `settings.pro` should look like this:

```
# Native lirc support
CONFIG += using_lirc
EXTRA_LIBS += -llirc_client
```

Recompile and install:

```
$ make distclean; make
$ su
# make install
```

If you're not using native LIRC support, ensure that the button names that come up with **irw** match the ones in the `.lircrc` file in your home directory. Note that if you are not using native LIRC support, the `.lircrc` file is in your home directory and has a "." as the first character.

Since we're using native LIRC support within MythTV, copy the `configfiles/hauppauge-lircrc-nativelirc` file into your `.mythtv/` directory and call it `lircrc`. It has a slightly different format than what you would use if you were using **irxevent**. Also note that it is in your `.mythtv` directory and does *not* have a "." in the filename.

```
$ cp configfiles/hauppauge-lircrc-nativelirc ~/.mythtv/lircrc
```

Add the commands to load the LIRC device drivers and start **lircd** to your `rc.local`:

```
$ su
# echo "modprobe lirc_i2c" >> /etc/rc.d/rc.local
# echo "/usr/local/sbin/lircd" >> /etc/rc.d/rc.local
# exit
```

You should now have native LIRC support within MythTV.

22.3 Logical Volume Manager (LVM).

LVM greatly increases the flexibility you have in managing your storage than traditional physical partitions. This section will provide some brief notes on how to use LVM to create storage space for your video files and how to add additional disk space in the future. There's lots more that can be done with LVM, so check the LVM HOWTO http://tldp.org/HOWTO/lvm−HOWTO/ document for details.

Make sure your kernel configuration includes LVM support or that it's available as a module. Today, most vendors include this by default. You'll also want to ensure that you have a copy of the LVM utilities; check your distribution, or get the latest versions from http://www.sistina.com/products_lvm.htm and build them manually.

Check that the **vgscan** program is being run at some point during your boot sequence – most distributions do this by default. Look for a message during boot up that looks like this: `vgscan -- reading all physical volumes (this may take a while...)`

LVM uses a few concepts you should be familiar with before starting.

- PV (Physical Volume). The actual partition on the hard drive.
- VG (Volume Group). The aggregation of all the PVs make a VG.
- LV (Logical Volume). Subdivision of the pool of space available in the VG into individual chunks, like /usr, /var/video, etc.

The following example assumes that you want to create a LVM partition from a chunk of space in /dev/hda5, using a reiserfs filesystem and mounted on /var/video. You later decide to extend this filesystem by adding a new disk: /dev/hdb.

You need to create at least one LVM partition for a physical volume. Use **fdisk** or your favorite partition editor to set the type to LVM (0x8e). If you're using an entire disk, create one big partition rather than using the device itself. e.g. use /dev/hdb1 not /dev/hdb.

`# fdisk /dev/hda` .... create partition 5, save partition table and reboot if you have to

Create the LVM physical volume from the partitions (repeat if you have multiple partitions to use):

```
# pvcreate /dev/hda5
```

Create a LVM volume group out of this physical volume called "VGforMyth" that is allocated in chunks that are a multiple of 64MB

```
# vgcreate --name VGforMyth -s 64m /dev/hda5
```

Create a logical volume of 20GB called "video" and then create the reiserfs filesystem and mount it:

```
# lvcreate --name video --size 20G VGforMyth
# mkreiserfs /dev/VGforMyth/video
# mount /dev/VGforMyth/video /var/video
```

Now create a 5GB volume for mythmusic files if you like:

```
# lvcreate --name music --size 5G VGforMyth
# mkreiserfs /dev/VGforMyth/music
# mount /dev/VGforMyth/music /var/music
```

Display the volume group status: `# vgdisplay -v`

Now, lets suppose you want to add a 60GB hard disk to the system as hdb and allocate 50GB of it to video storage.

First, create a single partition /dev/hdb1 covering the whole disk and make it type 0x8e using your partition editor.

`# fdisk /dev/hdb` .... create partition, set type, save and reboot if it says you have to

Create the new LVM physical volume:

`# pvcreate /dev/hdb1`

Add the new physical volume to the volume group:

`# vgextend VGforMyth /dev/hdb1`

NOTE: You may get errors at this point stating that there are no physical volumes available for adding to the LV, even though you know for a fact that there are. You may need to specify the physical volume in the `/dev/ide/host/bus/target/lun/etc` format.

Once you've completed one of the following two procedures, use **df** to check that you've got more space.

ReiserFS

Make the logical volume used for video bigger:

`# lvextend --size +50G /dev/VGforMyth/video`

Unmount, resize and remount the filesystem. Technically, you don't need to unmount and remount the ReiserFS.

```
# umount /var/video
# resize_reiserfs /dev/VGforMyth/video
# mount /dev/VGforMyth/video /var/video
```

ext2 or ext3

LVM comes with a program called **e2fsadm**.

Unmount, resize and remount the filesystem. The filesystem *must* be unmounted during this procedure.

```
# umount /var/video
# e2fsadm --size +50G /dev/VGforMyth/video
# mount /dev/VGforMyth/video /var/video
```

22.4 Advanced Partition Formatting.

MythTV creates large files. The partitions that your distribution sets up for you may not be optimized for large files. Using LVM in conjunction with the following techniques can be quite useful.

Ext3

With Ext3, your biggest gain is that in case of a crash and reboot you don't have to wait very long for your partition to be remounted. With Ext2, the **fsck** may take a long time to run on multi−gigabyte partitions.

When formatting the partition, the following command line should be better than the default. This example assumes that /dev/hdb1 has already been created using **fdisk**. If you're using LVM, /dev/hdb1 may be something like /dev/VGforMyth/video.

```
# mkfs.ext3 -T largefile4 -m 0 /dev/hdb1
```

The "−T largefile4" option creates one inode per 4 megabytes. The "−m 0" (zero, not "oh") parameter sets the amount of spaced reserved for root to zero. By default, this is 5%, which on a multi−gigabyte drive is a lot of reserved space.

You may also modify some aspects of a filesystem such as the reserved block percentage after it has been created using the **tune2fs** program:

```
# tune2fs -m 0 /dev/hdb1
```

You can check on your filesystem using the **dumpe2fs** program. See the **man** page for details.