Installing and using MythTV

Table of Contents

Table of Contents

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.

1. First things first.

2. Introduction.

3. Checking prerequisites.

4. System Configuration Requirements for Compiling MythTV.

5. Downloading and compiling.

12. MythPlugins.

13. MythWeb.

14. MythGallery.

15. MythGame.

16. MythMusic.

17. MythWeather.

18. MythVideo.

19. MythDVD.

20. MythNews.

21. Troubleshooting.

22. Miscellaneous.

23. Example Configurations.

1. First things first.

1.1 What is MythTV?

MythTV is a suite of programs that allow you to build the mythical home media convergence box on your own using Open Source software and operating systems.

MythTV has a number of capabilities. The television portion allows you to do the following:

- You may pause, fast–forward and rewind live Television.
- You may install multiple video capture cards to record more than one program at a time.
- You can have multiple servers, each with multiple capture cards in them. All servers are centrally managed and all programs are scheduled by the Master backend.
- You can have multiple clients (called "frontends" in MythTV parlance), each with a common view of all available programs. Any client can watch any program that was recorded by any of the servers. Clients can be diskless and controlled entirely by a remote control.
- You may use any combination of standard analog capture card, MPEG–2, MJPEG, DVB or HDTV capture devices. With appropriate hardware, MythTV can control set top boxes, often found in digital cable and satellite TV systems.
- Program Guide Data in North America is downloaded from Zap2It.com, a subsidiary of Tribune Media Services. This free service is called DataDirect, and provides MythTV almost two weeks of scheduling information. Program Guide Data in other countries is obtained using XMLTV. MythTV uses this information to create a schedule that maximizes the number of programs that can be recorded if you don't have enough tuners.

Other modules in MythTV include:

- MythGallery, a picture–viewing application
- MythVideo, a media–viewer for content not created within MythTV
- MythDVD, a DVD viewer / ripper
- MythMusic, a music playing / ripping application which supports MP3 and FLAC
- MythGame
- MythWeather
- MythNews, a RSS news grabber
- MythBrowser, a web browser
- MythWeb, which allows you to control your MythTV system using a web browser. With MythWeb, you can schedule recordings, delete files, change keybindings and more. With proper security, you may even schedule a program over the Internet and have it immediately acted on by the scheduler.
- MythPhone, phone and videophone using SIP.

1.2 QuickStart

Custom mini–distributions are 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://mysettopbox.tv if you'd like to install a custom version of Knoppix optimized for MythTV.

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

See http://bit.blkbk.com/ if you'd like to install MythTV on a Xbox.

See http://wilsonet.com/mythtv/ for instructions tailored to RedHat's Fedora Core distribution.

1.3 Upgrading from previous versions

The upgrade from previous versions should be transparent. Any changes to the database structure should be applied automatically.

It is *strongly* recommended that you back up your database before installing a new version of MythTV.

See Saving or Restoring the database for instructions.

1.4 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.

STOP **NOTE**: Please note that I am *NOT* the author/programmer of the MythTV application! I can not give you personalized installation support. If you are having issues installing MythTV you should examine the archives, or post your question to the MythTV−users mailing list.

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.

This HOWTO is for MythTV v0.18.1

1.5 Document conventions

The following conventions are used throughout this document.
boldface – used for program names.
`typewriter` – used for program paths.
emphasis – Pay attention here.

Pay more attention.

STOP Ignore at your own peril.

1.6 Mailing lists / getting help

It's recommended that you join the user list at http://www.mythtv.org/mailman/listinfo/mythtv−users. The developer list is at http://mythtv.org/cgi−bin/mailman/listinfo/mythtv−dev. Please keep the developer list strictly for development−related issues.

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

1.7 IRC

There are two IRC channels dedicated to MythTV which can be found on irc.freenode.net

- mythtv
- mythtv−users

The `mythtv` channel is where the developers discuss code. It is *not* a user−support channel. Please don't ask non−development related questions there.

Really. Even if there's no one in the mythtv−users IRC group or everyone seems to be ignoring you.

1.8 Bugzilla

If you feel you need to contribute to a bug database, use the MythTV Bugzilla at
http://www.mythtv.org/bugs/.

1.9 Contributing to this document

Contributions to the HOWTO are welcome, especially if you find a grammatical or spelling error, or if the wording of something is just plain confusing.

If you'd like to make a new contribution, feel free to email it to me mailto:rkulagow@rocketmail.com, but as the maintainer I retain the right to edit or choose not to use your submission.

If you would like to submit an update or contribution, please send it as either SGML or as plain text. *NO HTML*. The source used to create the HOWTO is in SGML / Linuxdoc. See the Linuxdoc HOWTO at http://www.tldp.org/HOWTO/Howtos−with−LinuxDoc.html for information on the linuxdoc format itself, or look at the `mythtv-HOWTO.sgml` file as an example.

Do not be afraid of SGML. A quick look at the source of this HOWTO will show that it is not difficult, so at least try to submit as SGML.

SGML patch files are the easiest. Run a **diff −u origfile newfile > doc.patch** and mail it to the address above.

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.

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 a good MythTV experience, you must understand that MythTV exercises your hardware more than a typical desktop. Encoder cards generate DMA across the PCI bus. The CPU is busy encoding / decoding video. Hard drives are constantly reading and writing data. Building a MythTV system on older / "spare" hardware may be an exercise in frustration and can waste many hours of valuable time.

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/lists/mythtv/ 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 if you are doing software−based encoding, less if you are using a hardware−based 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.
- A 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−150/250/350/500. 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 for MPEG−4, 2 GB/hour for MPEG−2 and RTjpeg and 7 GB/hour for ATSC HDTV.

NOTE: You *must* use DMA for hard drive access to prevent choppy or jittery video. Not all distributions 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 MythTV creates very large files, a filesystem that does well at deleting large files is important. Numerous benchmarks show that XFS and JFS do very well at this task. You are strongly encouraged to consider either of these for your MythTV filesystem.

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

Video Capture Device

In order to capture video, MythTV will need one or more video capture devices with Linux drivers. There are a number of classes of hardware available for capturing video.

Frame Grabbers.

This class of card is the simplest and is usually the cheapest. There is no on−board encoding of the analog video; hardware known as a Digital−Analog Converter (DAC) takes the video and presents it to the computer in an essentially raw digital form.

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.
```

STOP STOP 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.

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.

Hardware MPEG−2 encoders.

While inexpensive video−capture cards simply capture raw frames, leaving encoding to software, some higher−end cards incorporate hardware−based 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−150/250/350/500, Avermedia M179, Hauppauge "Freestyle" or Yuan M600 cards will allow you to use dedicated hardware encoders rather than your CPU. (The PVR−350 can simultaneously be used as an output device.) Using the on−board MPEG−2 encoder drastically reduces the CPU requirements for encoding.

NOTE: Motherboards with the Via chipset are notoriously bad with DMA and have caused numerous issues with ivtv, including hard locks. See the ivtv website http://ivtv.sf.net for the latest information on what works and what doesn't.

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.

DVB capture cards.

DVB is a video standard primarily found in Europe. MythTV supports DVB; see the pages maintained by Martin Smith martin@ethics−gradient.net at http://www.ethics−gradient.net/myth/mythdvb.html for more information. To see if your DVB card is supported, see the list of cards at http://www.linuxtv.org/download/dvb/CARDS for more information.

In the United States, you may use a card such as the TwinHan to obtain unencrypted Free−To−Air satellite channels.

HDTV.

Currently there are two HDTV card with Linux drivers which are known to operate in the United States – the HD−2000/HD−3000 cards available from http://www.pchdtv.com/ or http://mythic.tv/ and the Air2PC−ATSC−PCI available from http://www.cyberestore.com/product_info.php?cPath=28&products_id=103 or http://mythic.tv/.

The Air2PC is supported using the linux−dvb drivers.

To playback HDTV content, plan on a powerful CPU. "How powerful?" depends on a number of factors, such as the capture resolution, whether the video is progressive or interlaced, and whether your display card has hardware−assist support for Linux. For 720p content (1280x720), a 2.4GHz P4 should be sufficient.

For 1920x1080i−>1920x1080p with the better deinterlacing methods done in real time a 2.4GHz CPU is taxed, but should work if you use "Bob and Weave" deinterlacing, or if you have an NVIDIA card with MPEG−2 hardware acceleration. If you enable the hardware acceleration, you may be able to use a 1.8GHz processor.

Firewire.

You may use the Firewire output of the DCT6200 or the SA3250.

USB Capture Devices.

The Plextor ConvertX PVR devices are supported through Linux drivers available from http://www.plextor.com/english/support/LinuxSDK.htm. MythTV uses the Plextor to capture hardware encoded MPEG−4, so the host CPU requirements are low.

Hardware known NOT to work and other issues

- Hauppauge WinTV−D or −HD (no driver)
- Hauppauge WinTV−USB series

- Hauppauge WinTV−PVR−usb (model 602), or WinTV−PVR−PCI (model 880) cards (no driver − this is not the PVR−250/350 series of cards supported by the IvyTV driver)
- Hauppauge WinTV−PVR−USB2 (driver available at http://justiceforall.free.fr/pvrusb2.html, no MythTV support)
- ATI All−in−Wonder series

There are no known consumer−level capture devices which will allow you to capture the HDTV output (either DVI or HDMI) from a set−top box commonly found with digital cable systems or satellite systems.

There are no known consumer−level capture devices which will allow you to capture component output from a set−top box.

Sound card

The system needs a sound card or an on−board 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 on−board 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 voltages and can damage the sound card. 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/lists/mythtv/ 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@1vcm.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.

The PVR−350 is unique in that it also supports audio output. However, as of MythTV 0.17, the IvyTV drivers do not support volume control. There are two courses of action you may take:

1. Take the audio output from the PVR−350 and plug it into an input on a sound card on your machine. You may then use MythTV's internal audio controls.
2. Take the audio output from the PVR−350 and connect it directly to your television / audio system. You must indicate that you are using external audio control on the PVR−350 setup page.

Other Options

Some devices with on−board 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/Fedora Core

Red Hat Linux (9, 8.0 and 7.3) and Fedora Core (2 and 1) packages for MythTV and all of its add−on modules and some themes have been packaged by mailto:Axel.Thimm@ATrpms.net and are available at http://ATrpms.net/topic/multimedia/. All of the prerequisites for MythTV (such as XMLTV) are available as RPM packages. If you have problems with the RPMs, please contact the ATrpms lists at

http://lists.ATrpms.net/ or file a bug against http://bugzilla.ATrpms.net/.

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.net/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 from http://ATrpms.net/name/atrpms−kickstart/. Install the package with:

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

Then run:

```
# apt-get update
# apt-get dist-upgrade
# 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 kernel modules**, 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 at http://wilsonet.com/mythtv/ Just like 3rd−party packages, any 3rd−party documentation problems should be brought up with the 3rd−parties (maintainer, lists, bugzillas etc.).

Mandrake

Thac has created RPMs for MythTV for Mandrake 9.0 and above 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/unstable/mythtv/. 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 have followed the instructions on the above page you should have added

```
deb-src http://dijkstra.csh.rit.edu/~mdz/debian unstable mythtv
```

to your /etc/apt/sources.list file. Running **apt−get update** and then executing **apt−get build−dep mythtv** should install all the pre−requisites required to compile MythTV.

NOTE: Due to the requirement for Qt 3.1+, there are no packages for Debian woody/stable.

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 or higher 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.
```

Graphical installation tools

Mandrake

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 users: In versions after Mandrake 9.1, Mandrake 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.

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://addmedia.linuxfornewbies.org/ 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.

Mandrake 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. Once you have completed installing the pre–requisites, exit out of the shell and start a new one to ensure that any environment variables setup by the installation have a chance to take effect.

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.

Debian.

See http://dijkstra.csh.rit.edu/~mdz/debian/dists/unstable/mythtv/ for up–to–date information for Debian.

Build–dependencies for MythTV can be satisfied by adding the following to your `/etc/apt/sources.list`

```
# MythTV
deb http://dijkstra.csh.rit.edu/~mdz/debian unstable mythtv
deb-src http://dijkstra.csh.rit.edu/~mdz/debian unstable mythtv

# Christian Marillat's packages (mplayer, lame)
deb ftp://ftp.nerim.net/debian-marillat/ unstable main
```

and executing:

```
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]
```

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.

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, if it is not already there. 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
$
```

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 the directory holding the QT binaries needs to 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
$ which qmake
/usr/lib/qt3/bin/qmake
```

For Mandrake, you should see a value like `/usr/lib/qt3` for QTDIR and `/usr/lib/qt3/bin` should be in $PATH.

For Gentoo, you should see a value like `/usr/qt/3` for QTDIR and `/usr/qt/3/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` (`/usr/share/qt3/mkspecs` for Debian) 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 using **bash** as the shell, and may be applicable for a distribution which uses `/etc/profile.d`.

STOP **NOTE**: Mandrake 10 installs a `/etc/profile.d/qtdir3.sh` file, but it doesn't include the addition of the PATH variable. If you're running Mandrake 10, don't create a `mythtv.sh` file as detailed below; edit the `qtdir3.sh` file and add the PATH statement within the if / fi block. 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

Red Hat Linux 9 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# 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 unstable, QTDIR should be set to `/usr/share/qt3`. Edit the `/etc/profile` file using your favorite text editor and add something like:

```
PATH="/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/games"
QTDIR="/usr/share/qt3"
.
.
.
export PATH QTDIR
```

Device Permissions

MythTV will need access to the video4linux devices on your system. By default, your distribution may restrict access to these devices to the logged–in user, so if you will be automatically starting **mythbackend** from a script rather than an interactive terminal session you will need to make some adjustments.

NOTE: The following instructions are accurate for Mandrake.

Check for a file called `/etc/security/console.perms`. Open the file in your favorite text editor and look for a line that has:

```
<console>  0600 <v4l>        0600 root.video
```

and replace it with

```
<console>  0666 <v4l>        0666 root.video
```

What we're doing is allowing read and write access to the files in the video4linux directory.

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
```

NOTE: You may get an error about cvs not finding a `.cvspass` file the very first time you run it. This is normal. The other MythTV modules may be downloaded from CVS the same way; replace "mythtv" with "myththemes" or "mythplugins".

MythTV also maintains a CVS branch that incorporates bugfixes for issues discovered after a release has been packaged into a tarball.

If you would like to use code from this release branch, you would execute the following:

```
$ 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 co -r release-0-18-fixes myth

$ cvs -z3 -d :pserver:mythtv@cvs.mythtv.org:/var/lib/mythcvs co -r release-0-18-fixes myth
```

If you wish to stay up−to−date with CVS, go into the MythTV subdirectory and type:

```
$ cd mythtv
$ cvs -z3 update -PAd
```

after you've done your initial download.

STOP **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.

If you are in North America you will use the DataDirect grabber which is built−in to MythTV. You *do not* need to install XMLTV, but you need **wget** version 1.9.1 or higher.

Get XMLTV from http://xmltv.sourceforge.net. Download the latest version (0.5.39).

```
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 taken care of. Otherwise get lame and lame-devel from
<url url="http://ATrpms.net/name/lame/" name="http://ATrpms.net/name/lame/">.

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

Get lame from http://lame.sourceforge.net/. Download the source code to v3.96.1 by following the links from "Using" through "Download...".

5.1 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
$
```

5.2 XMLTV

Red Hat Linux and Fedora Core:

RPMs for **XMLTV** and all of its dependencies can be obtained from http://ATrpms.net/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 xmltv-grabbers
```

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 skip straight to the XMLTV compilation step.

Manual installation

Untar the xmltv file:

```
$ tar -xjf xmltv-0.5.39.tar.bz2
```

Install the xmltv prerequisites. The following prerequisites are the minimum required; when you actually start running the xmltv setup program it may alert you to other modules that are required.:

```
$ 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-0.5.39
$ 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. You will need to resolve any missing dependencies at this point, or your grabber may not work correctly.

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

5.3 Configuring the Zap2It.com DataDirect service

Introduction

Zap2It DataDirect is a service provided by Zap2It, a subsidiary of Tribune Media Services. TMS is one of the primary providers for program listing data in the United States and Canada. There are a number of advantages to using DataDirect rather than the standard XMLTV `tv_grab_na` grabber.

- The `tv_grab_na` grabber is no longer supported and will not be fixed when problems occur.
- It's much faster, because the grabber isn't downloading hundreds of HTML pages and trying to parse the program data out of the HTML.
- It contains more program information.
- It's easier on the Zap2It servers, since they're not generating hundreds of custom HTML pages when you scrape their listings.
- The service is available free of charge by periodically filling in surveys similar to the one presented during registration, and the data provider has no intention of charging for it in the future.

If you wish to use Zap2It DataDirect, you'll need to establish a user account. Go to http://labs.zap2it.com and click on the "New User? Sign Up" link. Read the license terms, and if you agree with them click "Accept". The next screen is the subscription registration. In the "Certificate Code" field, enter **ZIYN–DQZO–SBUT**.

Fill in the survey, then proceed to the lineup choices and configure your account for your particular location and the channels that you have. This configuration will be imported into MythTV when you first run the **setup** program.

See the section called Migrating from XMLTV to DataDirect for instructions on moving to DataDirect if you are currently using XMLTV. If you are starting from a fresh installation, the simplest thing is to immediately start using DataDirect.

5.4 Manually building MythTV

Unpack MythTV

```
$ tar -xjf mythtv-0.18.1.tar.bz2
$ cd mythtv-0.18.1
$ ./configure
```

As of MythTV version 0.18, there is no need to edit the `settings.pro` file to set compile−time options. The **configure** script will configure MythTV appropriately for your system. If you wish to change options, run **./configure −−help** to see what is available and to override and automatically detected options. See the `config.log` file after running **configure** to see previous runs.

On a single CPU machine, you would type the following to compile MythTV:

```
$ qmake mythtv.pro
$ make
```

If you want to build MythTV on a multi−CPU machine (or with **distcc**), specify "−j numjobs". In the following example, we will have two concurrent jobs executing. Do not set the number of jobs too high, or your compile will actually take longer to complete than it would if you did a "normal" build.

```
$ qmake mythtv.pro
$ make qmake
$ make -j 2
```

If you are using **distcc**, and you had two other host machines (red, blue) participating, you would do something like:

```
$ export DISTCC_HOSTS='localhost red blue'
$ qmake mythtv.pro
$ make qmake
$ make -j 6 CXX=distcc
```

The actual speed−up, if any, is dependant on a number of factors, such as number of CPUs / hosts, etc. The **distcc** documentation recommends using a −j value of twice the number of CPUs available to keep all of them busy.

Some timing information. The following should only be used for illustration; your actual results may vary. The test involves a complete `make distclean` to the final binary.

- P4 3.2Ghz HT: "standard" make: 12m 49s
- P4 3.2Ghz HT: make −j 2: 11m 24s

In the above example, we see that with a single CPU, a multi−stage **make** does not significantly decrease compile time.

Once the compile is done, 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.

Enabling real–time scheduling of the display thread.

MythTV supports real–time scheduling of the video output thread if it is installed SUID root.

This is an optional, but recommended step, if you understand the potential negative impact. As a positive, It can make the video display more smoothly, especially if you are decoding HDTV.

🛑 **NOTE**: Please note that this opens a security hole in your system.

If the tradeoff is acceptable:

```
$ su
# chmod a+s /usr/local/bin/mythfrontend
# exit
$
```

NOTE: Making this change will affect your ability to run **mythfrontend** within **gdb** for debugging purposes. If you need to debug, you will need to run as root.

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:
 • 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.
 • Run the mythfrontend program on your frontend machine, and a "Database Configuration" screen should appear. Set the "Host name" field to point to your Master backend's IP address.

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 Distribution–specific information

Mandrake

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. If this is happening to you, try running the following commands.

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

Red Hat Linux and Fedora Core

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
```

Gentoo

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

6.2 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 and Red Hat Linux/Fedora Core

```
$ mysql −u root < mc.sql
```

Debian 3.0

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

Gentoo

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

STOP **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.

STOP **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";
mysql> flush privileges;
```

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";
mysql> flush privileges;
```

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";
mysql> flush privileges;
```

7. Configuring Sound.

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 and Red Hat Linux/Fedora Core

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

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

on Red Hat/Fedora.

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 and Fedora Core 1 ship 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 the vendor distribution. 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.

If you'd like to use ALSA, you'll need to correctly setup your `asoundrc` file. Configuring this file is beyond the scope of this HOWTO. Once ALSA is working correctly, change the output sound device in mythfrontend–>setup–>Audio from `/dev/dsp` to `ALSA:default`. This field may be edited to suit your ALSA requirements.

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 PCM,0 100%,100% unmute
$ amixer set Line,0 75%,75% mute captur
$ amixer set Capture,0 100%,100% 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. If you are using an ALSA version after 1.0.6, use **alsamixer –V all** . 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, 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

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/Fedora Core

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*
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 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/Fedora Core

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.net/name/kernel/. You will also find suitable kernel modules for ALSA, lirc, bttv, ivtv, saa7134 etc. there. See also the main Red Hat/Fedora Core 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 pre−made 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-0.18.1/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 by enabling it in **configure** 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

Lirc has support for various IR transmitters. A popular model is the Actisys IR−200L http://store.snapstreamstore.com/accessories.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 &
^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 compile−time prerequisites have been installed, **mysql** is running and has had its initial database setup. It's now time to configure MythTV.

STOP **NOTE**: If you're running Debian unstable and you have compiled MythTV from source, you will need to install an additional package before you will be able to run MythTV. Execute the following to install the MySQL driver for QT.

```
$ su -
# apt-get install libqt3c102-mt-mysql
# exit
```

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 backend 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 the greatest amount of disk space on the Master backend because its tuner will always be the first choice. You will then want to add your other backends in the order of your preference for recording.

NOTE: It is possible to *not* have the cards on the Master backend be the first ones used. However, if you are new to MythTV it is easier to configure the Master backend first before moving on to the Slaves, at least until you become more familiar with the MythTV system. See Advanced Backend Configurations for information on configuring multiple backend systems in various ways.

Because MythTV now uses a database to store all 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 change the Host Name in the "Database Configuration" screen of the mythfrontend program.

NOTE: Users that have been running the frontend and the backend on different machines 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
```

Run the setup program:

```
$ mythtv-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 five choices.

9. Configuring MythTV. 41

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 be deploying multiple backends, or if your backend is on one system and you're running the frontend on another machine then *do not* use the "127.0.0.1" IP address.

NOTE: If you modify the 127.0.0.1 address and use a "real" IP address, you must use real IP addresses in both fields, otherwise your frontend machines will generate "Unexpected response to MYTH_PROTO_VERSION" errors.

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.

If you have made a mistake, you can delete a card by highlighting it and pressing the 'D' key, or you can highlight it and press the RETURN or 'E' key to edit it.

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, change the grabber to "DataDirect", then continue pressing the down arrow to move to the next field. Fill in the username and password that you have established with DataDirect, then move to the "Retrieve Listings" button and press the space bar.

NOTE: You need **wget** version 1.9.1 or higher to use DataDirect.

The setup program will contact the Zap2It servers and get your account information. Once you're done, you may click the Finish button and skip the next few paragraphs in this document since they only apply to users that are using the external XMLTV script to get their guide data.

If you wish to continue using the XMLTV grabber, then 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.

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.

If you have multiple input connection types (Such as Tuner 0 and Tuner 1), it's very, very likely that it is the lowest numbered input that is actually available on the device.

 NOTE: If you have a Hauppauge PVR−500, you must think of it has two PVR−150's on a single PCI card. For example, if you have a single PVR−500 card, it will appear as `/dev/video0` and

/dev/video1. Each /dev/video device will have a Tuner 0 and Tuner 1 input, so you would choose Tuner 0 for **both** devices.

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.

STOP **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.

Channel Editor

The channel editor is used to globally alter channel information, including items like hue, contrast, fine tuning and others. Users in North America shouldn't run the channel editor until after completing initial setup and running **mythfilldatabase** at least once to populate the database.

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.

NOTE: If you are using DataDirect and watching the output messages on the console or the log file it is normal to see a "401 Unauthorized" error followed by a "200 OK" when the connection to Zap2It is being established.

```
From : Sun Jun 13 05:00:00 2004 To : Mon Jun 14 05:00:00 2004 (UTC)
--02:58:01--
http://datadirect.webservices.zap2it.com/tvlistings/xtvdService
           => -'
Resolving datadirect.webservices.zap2it.com... 206.18.98.160
Connecting to datadirect.webservices.zap2it.com[206.18.98.160]:80...
connected.
HTTP request sent, awaiting response... 401 Unauthorized
Connecting to datadirect.webservices.zap2it.com[206.18.98.160]:80...
connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/xml]

    [   <=>                                 ] 114,125       63.57K/s

02:58:03 (63.53 KB/s) - -' saved [114125]

Your subscription expires on 08/20/2004 12:00:00 AM
Grab complete.  Actual data from Sun Jun 13 05:00:00 2004 to Mon Jun 14
00:00:00 2004 (UTC)
```

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 MythTV v0.18.1, 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,commflag,audio,
                              libav,jobqueue
--printexpire                 List of auto-expire programs
--printsched                  Upcoming scheduled programs
--testsched                   Test run scheduler (ignore existing schedule)
--resched                     Force the scheduler to update
--nosched                     Do not perform any scheduling
--version                     Version information
```

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 Automatically starting mythbackend at system boot time.

9.3 Configuring a non−master backend

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 have the correct IP address for the database server in the "Database Configuration" screen of the setup application on this slave backend.

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.

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 non−master backends.

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 −−help** will give a full listing of the options available.

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.

mythfilldatabase −−**refresh−today** will only pull guide data for today (in case of late−breaking changes to the schedule).

Periodically running mythfilldatabase

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

To use MythTV's built−in capability, you'll need to run the **mythfrontend** Setup option. From the MythFrontend, enter the Setup>General screen and advance to "Mythfilldatabase", the fourth screen. Select the checkbox, then complete the options as you see fit. The **mythbackend** program will now run **mythfilldatabase** at the times you specify.

9.5 Grabbing channel icons for DataDirect users

While the DataDirect TV listings service has several advantages, it does not support grabbing logo icons for the stations you receive. However, there are utilities provided with MythTV which you may use to grab your initial set of icons and to keep them updated if your lineups change.

First, you need to generate or obtain an XML file with the information for your stations.

If you have XMLTV software installed, there is a perl script in MythTV's `contrib/` directory which will generate this file for you. Run the command:

```
$ perl mkiconmap.pl
```

You will be asked for your zip code and the service that you use. If there are no errors, the `iconmap.xml` file that you need for the next step will be created.

If you do not have XMLTV software installed and do not want to install it for the sake of this minor task, there is a generic `contrib/master_iconmap.xml` which you can copy and use but this may not be as complete as using the specific information for your service.

Once you have an `iconmap.xml` file, add the icon information to your database and grab any new icons with the command:

```
$ mythfilldatabase −−import-icon-map iconmap.xml −−update-icon-map
```

10. Configuring mythfrontend.

Once you have completed configuration of your backend systems, the next step is to configure the frontend client.

When you start mythfrontend for the first time, it will attempt to connect to a configuration database on the local machine. If there is none, a "Database Configuration" screen will appear, and you will need to fill in some details. The "Host name" field needs the backend or database server's IP address or DNS name, and the User or password fields may need to be set to match your database user accounts. After editing those fields, press enter twice to write these configurations on your local machine, and attempt to connect to the database. If you make any mistakes, the screens will pop up again.

Now that mythfrontend has started up, you should have a number of buttons/choices. Before doing anything, go to TV, then to Setup and configure the frontend client.

NOTE: You should go through the various setup screens in mythfrontend before using any other

modules such as MythMusic, etc. to correctly initialize the database.

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.

Video Filters

MythTV provides a facility to include video filters while recording and during playback. These filters can be used to improve or modify the video image, including hiding the effects of an interlaced image or reducing the effects of noise in a poor video signal. The following is a brief introduction to introduce you to the filters that are available in MythTV version 0.18.1.

Applying filters

One or more filters can be included in a "filter chain". The filters to be used are identified in a "filter string". A filter string is a group of filter names and parameters separated by commas. To include parameters, the filter name is followed by "=" and the parameter information. There should be no spaces in the filter string. Here is an example filter string:

With parameters: `kerneldeint=10:1,denoise3d=12`

Without: `kerneldeint,denoise3d`

Recording filters are set for each individual channel. These may be used when encoding in software (MPEG–4, RTjpeg) but do not apply when using a capture card with hardware encoding such as those supported by the ivtv driver, DVB, HDTV or MJPEG cards. You can run MythTV's "setup" program and select the "Channel Editor". On the first page for each channel, you can enter a filter string in the box titled "Video filters". If you are running "mythweb" on your web server, you can click on "Settings" then "Channels" and enter filter strings in the "videofilters" column.

Playback filters are per–host and apply to any recording you watch from the frontend where filters have been applied. Playback filtering can only work with software decoding so the viaslice, xvmc, and ivtv outputs ignore filters entirely. From "mythfrontend" go to Setup–>TV Settings–>Playback. Enter your filter string in the box titled "Custom Filters".

Currently Available Filters

"Deinterlace Playback" checkbox.

This implements special behavior needed for the "bobdeint" filter but can also be used to choose any of the deinterlace filters. If you prefer, you may leave this unchecked and include any of the deinterlace filters, other than "bobdeint", in your custom filter chain.

o The "invert" filter

Invert ignores any parameters and inverts the pixel values of the video frames. In other words, a negative image. This would rarely be useful but may be a good example to verify that your filter strings take effect.

o The "linearblend" filter

It is a simple deinterlacing filter that ignores parameters and works by blending adjacent lines. It replaces combing in interlaced video with a less distracting "ghost" image.

o The "bobdeint" filter

This filter splits the interlaced image into two separate fields that can be line doubled then displayed at twice the frame rate. If the display is at the same refresh rate as the recording (59.92Hz NTSC or 50Hz PAL) this will cause each refresh to show objects in motion in a new position with no jagged edges. However, if the display is not synchronous, it will cause flickering or the appearance of the picture moving up and down by one line.

NOTE: This filter requires the frame rate to be doubled and therefore can only be used with the "Deinterlace Playback" checkbox. Do not include this in your filter chain.

o The "kerneldeint" filter

Kerneldeint is a more complex deinterlacing filter which applies a filter kernel using input from several lines. It generally removes combing without a "ghost" image, sometimes leaving a faint outline of the image from the other field. It is considered to be less distracting to watch than linearblend or no filter at all. It accepts one or two integer parameters separated by a colon.

The first parameter is the filter threshold and defaults to 12. Adjacent lines differing by more than the threshold value are filtered. The second option defaults to 0. If set to a non−zero value, it will cause the filter to skip chroma, and filter only the luminance. It may be useful on some capture cards which do not capture the chroma fields of interlaced video correctly.

o The "onefield" filter

This is a simple one−field deinterlacing filter that uses only one field of the interlaced video. By default it keeps the top field, though passing the parameter "bottom" will cause it to keep the bottom field instead.

This filter is primarily useful for those who display 1080i HDTV signals with a video mode that has 540 pixels vertically. The advantage over other deinterlacing filters is that scenes with motion never show combing or ghosting.

o The "adjust" filter

This filter adjusts the digital values for luma and chroma to ensure that they will fall within the ranges specified in the ITU−R601 standard. By default, this corrects a known problem for the luma range used by bt8x8 chips which causes video to look washed out. If parameters are passed, there need to be exactly six. However, passing a single parameter of "−1" will disable the filter.

1: luma minimum input value (int) 2: luma maximum input value (int) 3: luma gamma correction (float) 4:

chroma minimum input value (int) 5: chroma maximum input value (int) 6: chroma gamma correction (float)

The default bt8x8 correction values are equivalent to "16:253:1.0:2:253:1.0". Output ranges are fixed at ITU–R601 values (16–235 luma, 16–240 chroma).

NOTE: If it is not already specified in the filter chain, this filter will be automatically applied when recording with the "bttv" driver.

o The "quickdnr" filter

A fast temporal denoiser. This can take 1, 2 or 4 parameters, each being a value from "0" for the least filtering to "255" for the greatest filtering. With one parameter, the filter will compute the values it should use for all of its variables. Two parameters will set the filter strength for luma and chroma independently. If you are interested in how the algorithm works, you may examine the source code to see how four parameter are used.

o The "denoise3d" filter

A slower denoiser that applies a spatial and temporal low–pass filter. The spatial filter can remove some noise that quickdnr can't, but a more powerful CPU is needed. This filter accepts 3 float parameters:

- luma spatial filter strength
- chroma spatial filter strength
- luma temporal filter strength

Reasonable defaults will be selected for omitted parameters. The chroma temporal filter strength is calculated from the other filter strengths.

o The "crop" filter

Covers edges of video with black bars. This helps improve video quality when the edges of the frame are distorted. By default, this removes 16 pixels from each edge. This can optionally take four parameters representing top:left:bottom:right. The number times 16 is the number of pixels to remove so, for example, the default is "=1:1:1:1".

o The "forceyv12" and "forceyuv422p" filters

These force the filter manager to use the given format. You can use one of these at the head of a filter chain to change the capture format. The most likely use would be forceyuv422p to use YUV422P capture on cards with known chroma interlacing problems with YV12.

There are some filters included in the MythTV source code that should not be used:

o The "forcergb24" and "forceargb32" filters

The two RGB formats should not be used because there is no conversion filter for them yet.

o The "convert" filter

It exists but don't use it. The filter manager uses this filter automatically when it is unable to match the input/output formats of two adjacent filters.

o The "postprocess" filter

While this exists in MythTV source code, it is currently not recommended for use.

Usage Considerations

There are trade−offs to consider when deciding if it would be wise to use a filter. Any processing will modify the original image so you should assess if the filter has made a noticeable improvement to the picture in order to justify the impact of the processing. Adding any filter will inherently increase CPU usage. The impact can vary dramatically depending on your CPU type and speed, the resolution of the recording, which filters you are using and other factors. You can only determine what is right for you through experimentation. However, as a starting point, here are some filter strings that you may find useful:

For typical broadcast stations: "kerneldeint,quickdnr"

For stations with poor signal quality: "linearblend,denoise3d=12"

For synchronous TV−out: check Deinterlace with "Bob (2x framerate)"

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
- MPEG−2 – iTVC15/16 based cards (Hauppauge PVR−250/PVR−350); see http://ivtv.sourceforge.net
- HDTV – pcHDTV cards; see http://pchdtv.com and the Air2PC−ATSC−PCI see http://www.cyberestore.com/product_info.php?cPath=28&products_id=103
- 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 MPEG−4. RTjpeg has significantly fewer CPU demands than MPEG−4, but it generates larger files than MPEG−4 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
Transcoders
```

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 card type. 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/MPEG–4' 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 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.

See the What capture resolution should I use? How does video work? section for more information.

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.

STOP **NOTE to Red Hat/Fedora users**: Red Hat Linux and Fedora Core ship 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 Keyboard commands

The `keys.txt` file describes what the various keyboard commands are. If you have loaded mythweb, you may change the default keys to your liking.

mythfrontend

Arrow keys	used to move the highlight point around
ALT–F4	exit out of the application
Space/Enter	take action on the item under the highlight point
P	play in both "Watch a Recording" and "Delete a Recording"
D	delete in both "Watch a Recording" and "Delete a Recording"
U	to view details for the currently selected show on the Watch or Delete screens, EPG, "Program Finder", "Fix Scheduling Conflicts" or search results screens
O	to list the upcoming episodes for the currently selected show on the EPG, "Program Finder", "Program Recording Priorities", "Fix Scheduling Conflicts" or search results screens
I	edit recording options from the EPG, "Program Finder", "Program Recording Priorities", or "Fix Scheduling Conflicts" screens. From the Playback and Delete screens, 'I' presents options for recorded shows such as Auto Expire or Stop Recording. Pressing 'I' while on the Recording Options screen will take you to the Advanced Recording Options screen.

Watching TV or a recording

Up or down	keys change the channel

num pad	Type a number to enter a channel number or jump amount (HHMM format)
P	pause / play. You may also add an explicit keybinding for 'Play' through MythWeb, returning you to normal speed if you are in slow motion, rewind fast forward or pause mode.
C	change inputs on TV Tuner card
ESC	quits
I	puts the On–screen Display up again. During playback, 'I' toggles between position and show description info. If a jump amount is entered, jump to that position.
M	brings up the electronic program guide (Grid) –– see the EPG section
Page Up	jump back the configured number of minutes (default is 10)
Page Down	jump ahead the configured number of minutes (default is 10)
End or Z	skip to next commercial break marker
Home or Q	skip back to previous commercial break marker
T	toggle close caption support
F	rotate between the various Picture Adjustments (Colour, Hue, etc.) While Picture Adjustment is on–screen, use Left and Right arrows to adjust. These settings adjust the look of the video playback, and are independent of the G–key settings used at record–time.
[or F10	decrease volume
] or F11	increase volume
| or F9	toggle mute
/	jump to the next "favorite" channel
?	mark/unmark the current channel as a "favorite"
U	increase the play speed
J	decrease the play speed
A	Adjust time stretch (speed up or slow down normal play of audio and video
W	cycle through 4:3 aspect ratio, 16:9, 4:3 Zoom (like Pan and Scan), 16:9 Zoom, and 16:9 Stretch (eliminates black sidebars in TV signal)
F8	toggle the sleep timer 30m–>1hr–>1hr30m–>2hr–>Off
CTRL–B	Jump to the beginning of the recording / ringbuffer
+	Switch between audio streams
Left	(if a jump amount is entered) to jump back that amount
Right	(if a jump amount is entered) to jump ahead that amount

Without the stickykeys option selected	
Left	rewind the configured number of seconds (default is 5)
Right	fast forward the configured number of seconds (default is 30)
<	starts rewind mode as if stickykeys are selected
>	starts fast forward mode as if stickykeys are selected

With Stickykeys option selected	
Right	starts fast forward mode
Left	starts rewind mode

In fast forward or rewind mode:	
Left/Right	increases the ff/rew speed
0	plays at normal speed, but leaves the time indicator on screen
1 or 2	plays back more slowly than normal ff/rew speed (1 is slowest)
3	plays back at normal ff/rew speed
4–9	plays back faster than normal ff/rew speed (9 is fastest)
Space	exits fast forward or rewind mode

While video is paused:	
Left	rewind 1 frame
<	rewind 1 second
Right	advance 1 frame
>	advance 1 second

Watching TV only

G	rotate between the various Picture Adjustments (Colour, Hue, etc.) for recording. These values affect the look of the resulting .nuv file, and are independent of the playback picture settings. While Picture Adjustment is on−screen, use Left and Right arrows to adjust.
H	Channel history. Each repeat steps back through the previous channels.
O	Turns on 'Browse' mode, allowing user to browse channels and program info while watching current show FullScreen.
Y	switch between multiple capture cards. **NOTE**: you will lose your LiveTV buffer on your current card. Useful for different−sourced cards (such as Dish Network on one, HDTV over−the−air on another card.)

LiveTV Browse Mode

Left	browse program prior to current listed program
Right	browse program following current listed program
Up	browse program on channel above current listed channel/program
Down	browse program on channel below current listed channel/program
/	browse program on next favorite channel
0–9	enter a channel number to browse
Space/Enter	change channel to channel of current listed program
R/r	Toggle recording of current program (cycles through types)
ESC/O	Exit Browse mode

Playback Recording Zoom Mode

Left	Move video to Left
Right	Move video to Right
Up	Move video Up
Down	Move video Down
PageUp	Zoom In
PageDown	Zoom Out
Space/Enter	Exit Zoom mode leaving picture at current size and position
ESC	Exit Zoom mode and return to original size

If you have two or more tuner cards

V	toggle Picture–in–picture on or off
B	toggles the window focus (lets you change channels on the PiP window)
N	swaps the two channels by changing channels on both cards

Watching a recording only

Space/Enter	set a bookmark at that point. Next time you start the recording, you will automatically jump forward to this point and clear the bookmark.
X	queues the current recording for transcoding
O	brings up menu to allow toggling settings such as Commercial Auto–Skip, Auto–Expire, etc.
D	exits the current recording and displays the Delete menu
E or M	enters/exits edit mode.

	In edit mode
Left/Right	move forward and backward
Up/Down	alter the amount of time you jump forward and backward. Increments are: nearest cutpoint, nearest video keyframe, 1 frame, 0.5 seconds, 1 second, 20 seconds, 1 minute, 5 minutes, and 10 minutes.
PageUp/PageDown	move forward and backward to the nearest cut point
< or >	move forward or backward by 10 times the normal jump amount

Space/Enter	allows you to set or delete a cut point
Z	loads the commercial skip list (if one exists) into the cutlist
C or Q	clear all cut points in the cutlist
I	Inverts the cutlist

EPG

Arrows	are used to move the highlighted program point around
A, D, S, W	perform the same as left, right, down and up
PageUp/PageDown	move the channel list up or down a page
Home/End	move the highlight left or right by one day
Ctrl+Left or <	move the highlight left by one page
Ctrl+Right or >	move the highlight right by one page
9, 3, 7, 1	(like a numeric keypad) perform the same as PageUp, PageDown, Home and End
I	bring up more information about a show, and allow you to schedule a recording. If you select "Record this showing" while watching Live TV you can "Instant Record" a program.
Space/Enter	allow you resolve conflicts or change overrides. If the program is not already scheduled to record, it will instead act like pressing 'I'.
M	when on a channel will change to that channel
ESC or C	exits without changing the channel
R	change the current item from Recording/Not–Recording. Successive keypresses cycle through the scheduled recording type list.
X	change the channel to the currently selected channel without leaving the EPG (Most useful in the alternate EPG)
?	mark/unmark the current channel as a "favorite"
/ or 4	toggle the guide listing between all channels and filtered "favorites"

Setting Program or Channel Recording Priorities

Right	increases priority value
Left	decreases priority value
1	sorts by title
2	sorts by priority
Home/End	toggle sort priority
I	edit recording options
ESC	commits changes and exits

Viewing Scheduled Recordings/Resolving Conflicts

1	show all recordings
2	show only important recordings
Home/End	toggle show showing all/important
I	edit recording options
Space/Enter	resolve conflict or override

Viewing Search Listings

Home	change to the previous view if applicable
End	change to the next view if applicable
M	select another view if applicable. In the title and description search popup, press M again to edit or delete the selected view.

Recording Profiles Setup Screen

D	on a custom profile group displays a popup to delete the group

Recording Groups

In the Watch Recordings screen, Recording Groups allow you to separate programs into user−defined categories, such as "Kids", "Alice", "Bob", etc. This can be used to reduce clutter, or to segregate content if you use the PIN function.

M	change the view or to set a group password
I	move a program from one Recording Group to another

Watch Recordings Screen

1 or F1	Meaning of the icons
/	Tags a recording. Tagged recordings can be played either in order or shuffled and deleted as a group. You can also change the recording group for several recordings at once by tagging them and using the popup INFO menu to change the recording group.
?	Clear the tagged list.

Remote Controls

If you are using MythTV with just a remote control then it is suggested that you map the remote control keys as described below. Your remote control may not have the same set of keys as those named below, the names are only a suggestion that roughly correspond to the function.

If you are adding new key bindings to the program then consideration of this suggested list will help users with remote controls.

This list assumes a minimal remote control that only has 20 keys, nearly all features can be used with this configuration. If you have more keys then you can access all of the features. With only 16 keys most features are usable.

REMOTE CONTROL	LIRC KEYSTROKE	FUNCTION
0 – 9	0 – 9	channel selection, EPG navigation, ff/rew speed setting (with stickykeys)
Left Arrow	Left	scroll left, rewind
Right Arrow	Right	scroll right, fast forward
Up Arrow	Up	scroll up, channel change up
Down Arrow	Down	scroll down, channel change down
Select / OK / Play	Space	Select item, play (with stickykeys) set bookmark
Cancel	Escape	Cancel, quit playback
Menu	m	EPG (from watching TV) edit (from playback).
Pause	p	Pause
Other key 1	i	Information
Other key 2	c	Change tuner card input

11.2 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. Download the MythThemes tarball from the

website and untar it:

```
$ tar -xjf myththemes-0.18.1.tar.bz2
$ cd ~/myththemes-0.18.1
$ qmake myththemes.pro
$ su -
# make install
# exit
$
```

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

11.3 Adding DirecTV information to the database

A script used to exist for adding DirecTV pay−per−view information into the database. As of 2005−04−01 this script no longer appears to work, and this section will be removed from the HOWTO with the release of 0.19.

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.

11.4 Adding DishTV information to the database

A script for adding Pay Per View information into the MythTV database for DishTV subscribers is available at http://www.mythppv.com/.

11.5 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.

11.6 Using Shutdown/Wakeup

What does the MythTV Shutdown/Wakeup function do? The scheduler on the Master backend (MBE) keeps track of the idle status of the entire MythTV system, including the Slave backends (SBE). If it considers the system to be idle, and thus ready to shutdown, it sets the wakeuptime to the time of the next recording and then proceeds to shut down all Slave backends and then itself. Once it is time to begin recording, the Master backend and the Slave Backends are automatically woken up. This system allows MythTV to record like a

normal VCR, thereby conserving power when not in active use.

In order to use the Shutdown/Wakeup function there must be some method of waking up the Master backend. There are any number of solutions, but we will discuss in detail two possibilities:

- Use another server that runs 24/7 and have it send a WakeOnLAN (WOL) packet to wake the Master backend. This assumes that you have the WOL tools installed, and that your Master backend motherboard supports WOL.
- Use your motherboard's BIOS wakeup capability. You'll need a motherboard that supports BIOS wakeup, and some tools. Two that work are: http://sourceforge.net/projects/nvram−wakeup and http://www.malloc.de/tools/wakeup_clock.html

A deeper look into the operation

The scheduler keeps track of the idle status of the MythTV system. To determine whether or not the MythTV system is idle, the following conditions must be met for a period of time defined in the "Idle timeout (secs)" parameter.

- no client is connected to the server
- no recording (neither LiveTV nor a regular recording) is currently taking place
- no recording starts within a definable amount of time ("Max. wait for recording (min)")
- the "pre Shutdown check−command" returns 0

If we get to this idle state the Master backend will set the wakeuptime using the "Set wakeuptime command", which is the same for WOL and BIOS wakeup. The Master backend will then shut down the Slave backends and itself using the "Server halt command".

One caveat is that the scheduler tries to guess if the Master backend was started by a wakeup call or by the user. If it thinks it was woken up by a user, it blocks shutdown until a client connects to the Master backend, after which it will behave as described above. To disable this feature, unset "Block shutdown before client connected" in the mythfrontend Setup−>Setup−>General screen.

Once it is time to startup the system, the Master backend is woken up first and will wakeup the Slave backends using the "Wake command for slaves". At this time, there is no support for starting only the required Slave backend, so all Slave backends will startup.

Setting up the MythTV side of this extension.

There are a number of options that are used to control the Shutdown / Wakeup feature.

Shutdown/Wakeup Options:

- "Idle timeout (secs)" is the time the server waits while idle until a shutdown occurs.
- "Max. wait for recording (min)" is the time the Master backend waits for a recording without shutting down. For example, this would be used to prevent a 10 minute system shutdown if a recording is set to start 15 minutes from now.
- "Startup before rec. (secs)" Sets how long before a programmed recording the MythTV system will be woken up. This should be roughly be the time your systems need to bootup, and if you have Slave backends, you'll need to ensure this value is long enough for all your machines to perform their bootup cycle.

- "Wakeup time format" is the format of the wakeup time that is given in the "Set wakeuptime command" as a parameter "$time". You need to set this according to your wakeup mechanism. If you need seconds since the epoch (1970−01−01) set the "Wakeup time format" to "time_t".
- "Set wakeuptime command" is the command executed to set the new wakeuptime.
- "Server Halt Command" is the command executed to shutdown the Master backend and the Slave backends.
- "pre Shutdown check−command" is an optional command used to give a "Go/NO−GO" decision from a non−MythTV source. This command is executed immediately before the shutdown would occur. The return value is use to make the following choices:
  - If it returns a "0" the shutdown will occur as scheduled.
  - If it returns a "1" the "idle timeout" will be reset and the system waits again for the timeout.
  - If it returns a "2" the entire shutdown sequence is reset. This means that a new client connect is needed before a shutdown occurs, unless you have the "Wait for client connect" setting disabled, in which case this is the same as returning "1". An example of a use for this return value is to prevent the shutdown if a user is currently logged in, or if a specific program (i.e. transcode, automatic updates, etc.) is currently running. If you don't need it, leave the field blank.

The "WakeOnLan settings": These settings have nothing to do with using BIOS or WOL wakeup, they are the same for both.

- "Master backend" This setting defines timings for the frontends to wakeup the Master backend using WOL. Useful if your frontend can emit a WOL packet so you don't need to physically go to the Master backend if you're trying to watch TV.
- "Reconnect wait time (secs)" is the time the frontend waits after executing the "Wake command" before attempting to retry the connection. This should be roughly the amount of time your Master backend needs for bootup. Set to "0" to disable. The frontends will retry to connect for "Count of reconnect tries" times before giving up.
- "Wake command for slaves" is the *one* command executed to wake your Slave backends. This should be a script that contains the calls to wakeup all Slave backend systems.

Using WOL to wake your Master backend.

To use WOL to wake your Master backend you will need a WOL capable Master backend, a machine that runs 24/7 which can execute an at−job and nc (netcat) on the Master backend. I use some little bash scripts to make my DSL router wakeup my mythbox if required.

Replace $SERVER and $PORT with your own settings! On my Master backend I have a script that gets called as 'setwakeuptime command' which looks like the following:

```
#! /bin/sh
echo $@ | nc $SERVER $PORT
```

This simply cats the parameters (that is $time) to my 24/7 server. On my $SERVER I have (x)inetd listening on $PORT starting a little script which cares about setting the at−job. The following additions are necessary on the $SERVER:

If you use **inetd**:

In `/etc/inetd.conf` add:

```
mythwake stream tcp nowait mythtv /usr/sbin/tcpd /usr/local/bin/mythwake
```

If you use **xinetd**, save the following as **mythwake** in your /etc/xinet.d/ directory:

```
service mythwake
  {
        socket_type      = stream
        wait             = no
        user             = mythtv
        protocol         = tcp
        id               = mythwake
        server           = /usr/local/bin/mythwake
  }
```

and add the following to /etc/services:

```
mythwake           $PORT/tcp
```

Finally, /usr/local/bin/mythwake looks like:

```
#! /bin/bash
#this should be a command to wake your server
WAKECMD="#!/bin/sh\n /usr/local/bin/wakeMBE"
#first we need to delete all wake jobs in queue
for JOB in atq | cut -f 1 ; do
    atrm $JOB;
done
#now we read the date from 'nc'
read date;
#now set the atjob
echo -e "$WAKECMD" | at $date ;
```

SECURITY WARNING: Be sure to secure $SERVER:$PORT from untrusted networks, because this allows 3rd parties to run arbitrary code on your server!

Using BIOS wakeup to wake your Master backend.

Since I don't use this, I cannot say much about this. If your motherboard supports any wakeup tool you have to call that tool as "Set wakeuptime command" with the "Wakeup time format" suitable for that tool.

Wakeup the MySQL server using WOL

If your MySQL server and your Master backend are not on the same machine, you can have the Master backend wake your MySQL server using WOL. You will find the settings for this in the second page of the setup program, or at the end of mysql.txt. The meanings are the same as discussed in "The WakeOnLan settings" above.

Tips/Tricks:

If, for example, one of the Slave backends is also your desktop computer, you could simply use a little script as 'server halt command' which first calls /sbin/shutdown -t TIMEOUT where TIMEOUT is a value sufficient for you to react. You could then popup a window using *dialog, asking for permission to shutdown. If you cancel the shutdown, simply call /sbin/shutdown -c.

If you get "nvram−wakeup: /dev/rtc: Device or resource busy" your set−wakeuptime−script should stop the program that uses `/dev/rtc` before setting the wakeuptime.

11.7 Scheduling Recordings

The MythTV master backend is responsible for managing the schedule for all TV tuner cards on the master and any slave. It's job is to search the TV listing for the shows you have requested and assign recordings to the TV tuner cards. If none of the shows that you've chosen overlap, it simply records all of the shows. However, if there are shows where the beginning and end times overlap, the scheduler follows rules that you've specified or makes logical decisions about what would be best if you haven't express your preference. Further, the "Upcoming Recordings" page allows you make specific decisions about what you really do and don't want to record.

Record Types

When you choose a show that you would like to record from the Options Page, there are eight different types of rules to help the scheduler find which showings you would like to record.

- Single Record −− record only this title at this specific time. This is the best way to be sure that a certain showing is recorded. However, if the TV listings change and the show is not broadcast at that time, the show will not be recorded.
- Find One −− this will record a title once from any of the times that appear in the TV listings. This is useful for recording a movie or special that has multiple showings because it allows the scheduler to choose one that doesn't conflict. It is not a good choice for recording a single episode of a series because it records the first available showing of the title without regard to the episode information.
- Record Weekly −− this records a show whenever the title is listed on the same channel, weekday and time. Note that if the TV station changes the schedule for a special episode, it would not be recorded. However, you can add a Single record for the special episode.
- Find Weekly −− this will record a title once per week from any of the times that appear in the TV listings beginning from the time of the showing that was selected when the rule was set. This is useful for news, current events or other programs where the same episode is shown several times each week but the listings may not include descriptive information. This is not a good choice if there are different episodes shown during the week.
- Record Daily −− this records a show whenever the title is listed for the time and channel on any day of the week. Here again, a show will not be recorded if the time was altered by the station.
- Find Daily −− this will record a title once per day from any of the times that appear in the TV listings beginning from the time of the showing that was selected when the rule was set. This is useful for news, current events or other programs where the same episode is shown several times each day but the listings may not include descriptive information. This is not a good choice if there are different episodes shown during the day.
- Channel Record −− records a show anytime the title is listed on this channel. This is perhaps the most common rule to use for most shows.
- Record All −− record this title on any channel at any time. This can be useful if a station has sister stations where shows are rebroadcast allowing the scheduler to record rebroadcasts on the other station when the original airing cannot be recorded.

Duplicates

Singles will record without regard to duplicate matching.

The standard recurring methods of All, Channel, Weekly and Daily use the descriptive information in the TV listings to try to record only one showing of each unique episode. However, This goal is sometimes complicated by the fact that the stations may not include a description for a specific episode but use a generic description for the series instead. When there is a generic description, the default behavior is to assume that it may be an episode that you have not seen and to record it for you. One of the duplicate matching options is "Record new episodes only". If this is selected, listing that have an original air date of more than 14 days earlier are considered repeats and are not eligible to record. Generally, generic episodes will be marked as repeats also.

Because of generic episodes and other situations, MythTV offers an alternative approach where shows may be recorded by choosing from multiple showings even when the descriptive information is not reliable. All of the "Find" record types look for matching titles in the listings. If there is a showing with specific episode information and that episode has recorded before, that showing is marked as previously or currently recorded. The scheduler will then choose to record the earliest non−conflicting showing from any of other remaining showings regardless of the descriptive information. Generally, Find One is most useful for movies or specials and the Find Daily and Find Weekly rules are best for news or current events shows that are repeated. However, these may be useful in other situations where the standard recording rules may not work correctly.

Conflicts

As you add more shows that you would like to record, the scheduler will eventually encounter conflicts. If there are two shows at the same time and you have two or more TV tuner cards, both shows will record. However, if there are more shows than cards, the scheduler will have to decide what it thinks it should not record based on the information you have given. If you see an unexpected situation you are not "stuck" with the scheduler's choice. You can still tell the scheduler exactly which shows you do want to record and/or don't want to record in any situation.

Priority

By default, all shows you select are have equal value to the scheduler. There are a set of rules to make good choices when two or more shows are in conflict. However, priority values let the scheduler know what you want so that it can set the schedule based on your preferences.

Initially all priority values are set to zero. You may choose to leave everything at "0" and let the scheduler follow rules to guess what you might prefer when there are conflicts. However, if you have one or two favorite shows, you may want to increase the priority value so the scheduler will know that you would prefer recording these over other shows. You might use certain values to rate shows so that all favorites are 2. good shows are 1 and extra 'filler' shows are all −1 for example. You could sort each title on the "Set Priorities" to have a unique value so the scheduler can always know which show you'd prefer verses any other show. The choice and style are entirely up to you. However, the more information you give to the scheduler, the more likely it will make the choices you would like in the first place.

The scheduler choices are based on the total priority for a showing by adding up four different priority factors. By default, all of these factors are "0" but you may use any combination to express your likes and needs.

Per record rule −− this is the "priority" field in the "Scheduling Options" section of the options page and this values is included for any showings that match the recording rule. You may choose to only use these values and not use the other factors for the sake of simplicity and clarity.

Per record type –– Setup–>TV Settings–>Recording Priorities–>General allows you to add to the priority based on the type. It may make sense to increase the value for "Single" so that by default they have an extra advantage over other shows. You may want to decrease the value for Find rules so that they will be less likely to interfere with regular shows and will be more likely to record in a non–conflicting time instead.

Per channel –– Setup–>TV Settings–>Recording Priorities–>Channel Priorities can be useful if you believe that you prefer any of the shows on certain channels. This would give all shows on a channel an advantage by default.

Input preference –– in the MythTV "setup" program, the "Input Connections" section allows you to add additional priority in the "Input preference". This is simply another priority factor but has an interesting effect. If a card input has a higher value than the other cards, the scheduler will see that you would rather record showings of episodes on this card rather than a showing on another card. If you have multiple cards of different quality, you may want to set input preference to encourage the scheduler to record shows on your best card(s) whenever possible. This can also be useful if you have multiple video sources which include the same stations. For example, with digital and analog cable you could increase the digital cable input preference by 1 to tell the scheduler that you want to record from the digital station whenever possible but the station on the analog input could still be used when the digital input is busy.

For any single showing of any show you've chosen to record, these factors are added together to find the "total priority". This is the priority that the scheduler uses to decide which shows are given the first choice when filling in the schedule.

Scheduling decisions

Here are the actual decisions made by the scheduler as it fills in the schedule.

- Currently recording beats not currently recording –– A recording in progress can not be moved so it "wins" it's current timeslot.
- Higher total priority beats lower total priority –– This is the core of the scheduling process. Episodes of the highest priority show are placed on the first available card then the next highest priority and so on.
- Future start time beats passed start time –– If there is an episode in progress and also a later showing of the same episode, it is better to record the complete episode. If there isn't another showing, start recording now to record the remaining portion. This should only happen if you add a new rule while the show is in progress or if the master backend is started after the start time of a scheduled show.
- More specific record type beats less specific record type –– If two shows are on at the same time and have the same total priority but different types they will be sorted by Single then Find One, Record Weekly, Find Weekly, Record Daily, Find Daily, Channel and finally All. This only applies if the priorities are the same.
- If both start times have passed, later start time beats earlier start time –– This attempts to miss the least amount of time.
- If neither start time has passed, earlier start time beats later start time –– This rule help assure that the earliest showing of an episode has the advantage.
- Lower input id beats higher input id –– The scheduler fills in open time slots on the first available card. The next cards is used when there is another show already placed for the first card.
- Older record rule beats newer record rule –– If two shows are still equal after all of these other checks, the show whose record rule was added first is preferred over a more recent addition.

Reschedule Higher Priorities

Setup−>TV Settings−>Recording Priorities−>General has a checkbox for "Reschedule Higher Priorities" which tells the scheduler to try to be a little smarter in certain situations. If this is checked, the scheduler will look for situations where a show cannot record because all inputs for the channel are used for higher priority shows. It will check to see if any of the other shows could be recorded at another time so that the conflicting show can be recorded in it's place.

Generally, this is a good strategy but there are tradeoffs. If a higher priority show is postponed, you will not get to watch it until it is recorded in the later timeslot. There is also a risk that the TV listings may change and the later showing may go away. In this rare case the higher priority show may never record. On the other hand, if you do not use this option you will miss recording some lower priority shows unnecessarily unless you make similar these changes manually.

By using Reschedule Higher Priorities, the scheduler will do a better job of recording as many of your shows as possible when left unattended. It will also be easy to see that shows have been marked to record at a later time. You can then decide for yourself when you would prefer to record the first showing.

Controlling Your Schedule

The Manage Recordings−>Upcoming Recordings page is your control center for the MythTV scheduler. Unlike other DVR systems, this one page gives you all of the information and tools you need to see all of your alternatives and make whatever adjustments you desire.

The upper half of the screen has a scrollable box listing items that match your record rules sorted by time. The lower half shows the details for the highlighted item. There are two 'views' available. Press "1" to include all of the items that match record rules even if they do not need to be recorded. Press "2" to focus on just the things that will record and items that may need your attention. The message in the upper right−hand corner will remind you when there are conflicts that would prevent one or more shows from being recorded.

The items in the list are colored white for things that will record, gray for those that do not need to record and yellow when there is a time conflict. Items at the top of the list may also be highlighted indicating that the recording is in progress.

Along with the channels, start times and titles, the right−hand column has a status code. Numbers indicate which card number has been assigned to record the show. Letters are used to indicate the reason that something will not be recorded. Just below the box is a short status message for the highlighted item that indicates the type of record rule that was matched and a one or two word explanation of the status code. If you press SELECT, you will see more information about the status.

There are a few status codes that may require your attention. "C" indicates that there are more overlapping shows to record than there are TV tuners to record them. "L" indicates that the scheduler found that it may be better to record a later showing of this episode. These states happen as a result of your choices and should normally reflect your preferences. However, you may notice situations where you would like to modify the scheduler's initial choices.

The first thing you can do is to highlight an item and press INFO once or twice to see the Recording Options page. From this page you can change the record rule type, the duplicate matching rules, or raise or lower the priority to resolve whatever problem you noticed.

Additionally, you can treat any individual showing as an exception that you do want to record or don't want to record. To use these "override" features, highlight the item and press SELECT. You will see a message explaining the current status and at least an "OK" button to exit without making changes.

For items scheduled to record, there will be a button for "Don't record" which will prevent recording this showing but will still allow the same episode to record in the future. If there is episode description information, you may also see a button for "Never record". This prevents recording this showing and tells MythTV to remember that this is an episode that you've seen or don't need to see if it is ever in the TV listings again.

For items that are not scheduled to record, the message will describe the reason and in the case of "C" or "L" it will include a list of the shows that are scheduled to record instead. For any item that could potentially be recorded there will be buttons for "Edit Options" and "Add Override". "Edit Options" will allow you to change the options for the existing record rule such as raising the priority so that the show will record. These changes would apply to this and all future showings that match this record rule. "Add Override" will allow you to set options that apply to the specific showing without affecting the existing record rule.

If you return to an override page after an override has already been set, you will also see a "Clear override" to undo your changes. This option makes it very easy to try out some "what if" attempts when deciding on your best strategy in a difficult situation.

12. MythPlugins.

MythTV has a rich set of plugins available. Beginning with MythTV v0.18, a single tarball contains all of the plugins. Once you have downloaded the tarball, untar it and run the **configure** script:

```
$ tar -xjf mythplugins-0.18.1.tar.bz2
$ cd mythplugins-0.18.1
$ ./configure --help

Usage:   configure [options]
Options: [defaults in brackets after descriptions]
NB:      all --enable-* options can be reversed with --disable-*

Generic options:
  --help                  print this message
  --enable-all            Enable all options
  --enable-opengl         enable OpenGL (Music and Gallery) [default=no]

MythBrowser related options:
  --enable-mythbrowser    build the mythbrowser plugin [default=yes]

MythDVD related options:
  --enable-mythdvd        build the mythdvd plugin [default=yes]
  --enable-transcode      enable DVD ripping and transcoding [default=no]
  --enable-vcd            enable VCD playing [default=no]

MythGallery related options:
  --enable-mythgallery    build the mythgallery plugin [default=yes]
  --enable-exif           enable reading of EXIF headers [default=no]

MythGame related options:
  --enable-mythgame       build the mythgame plugin [default=yes]

MythMusic related options:
  --enable-mythmusic      build the mythmusic plugin [default=yes]
```

```
       --enable-fftw              enable fftw visualizers [default=no]
       --enable-sdl               use SDL for the synaesthesia output [default=no]
       --enable-aac               enable AAC/MP4 audio file decompression [default=no]

   MythNews related options:
       --enable-mythnews          build the mythnews plugin [default=yes]

   MythPhone related options:
       --enable-mythphone         build the mythphone plugin [default=yes]
       --enable-festival          enable festival TTS Engine [default=no]

   MythVideo related options:
       --enable-mythvideo         build the mythvideo plugin [default=yes]

   MythWeather related options:
       --enable-mythweather       build the mythweather plugin [default=yes]
```

The sections detailing configuration of the plugins assume that you are already in the
~/mythplugins-0.18.1/ directory.

Note that a single **configure** script now does configuration for all modules. By default, it will compile all modules. If you do not wish to compile a module, either because it is of no use to you, or you do not have the prerequisites, the simplest course of action is to go through the various sections below, satisfying the prerequisites, and then compile all modules at once.

For example, you would only like to compile and install MythGallery and MythMusic. For MythGallery, you would like to use the EXIF data in the .JPG file to present additional information onscreen. For MythMusic, you would like to enable support for the opengl and FFT visualizations. The command line would look like:

```
$ cd ~/mythplugins-0.18.1
$ ./configure --disable-all --enable-opengl --enable-mythgallery --enable-exif --enable-my
$ qmake mythplugins.pro
$ make
$ su -
# make install
# exit
$
```

For simplicity, the examples below will disable compilation of all other modules other than the one being discussed.

13. MythWeb.

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.

13.1 Installation and prerequisites

Mythweb is a part of the **mythplugins** package. See MythPlugins for instructions on downloading the tarball. The next step depends on whether your distribution has a web server and if you have PHP support.

Mandrake

Mandrake 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
```

13.2 Completing the installation

```
$ cd ~/mythplugins-0.18.1/mythweb
$ 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.

13.3 Resetting the key binding table

MythWeb allows you to configure which keys are bound to which actions within MythTV. If you'd like to reset this back to the default, execute the following command:

```
$ echo "delete from keybindings ;" | mysql -u mythtv -pmythtv mythconverg
```

13.4 Resetting the theme.

If you find yourself wedged into a theme that isn't working, open your web browser and go to site:
```
http://mythweb_name_or_ip_address/mythweb/settings.php?RESET_THEME=yes
```

14. MythGallery.

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

14.1 Installation and prerequisites

MythGallery is a part of the **mythplugins** package. See MythPlugins for instructions on downloading the tarball.

There are a number of transitions available, some requiring OpenGL support. You will also need to install a TIFF library. Under Mandrake, you would perform the following command:

```
# urpmi libtiff3-devel
```

Once you have satisfied the prerequisites for your distribution, install the application:

```
$ cd ~/mythplugins-0.18.1
$ ./configure --disable-all --enable-opengl --enable-mythgallery
$ qmake mythplugins.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.

The controls for MythGallery can be found in the README that comes with the application.

14.2 Using MythGallery

When you first start MythGallery, you will see a thumbnail view of any folders and pictures in the Gallery Directory you specified in setup. If this is the first time you have accessed this directory, the thumbnails will be generated on the fly. If the Gallery Dir is writable, these thumbnails will be cached thus speeding up future access. On the left is a greyed−out menu of options.

Use the arrow keys to select a folder or picture to open/view with the Select key, or use the Menu key to toggle access the menu on the left. The menu options are as follows:

- Slideshow – Will cycle through all the pictures in the current folder. The currently selected item must be a picture (not a folder) for this to work. It does not currently traverse subfolders.
- Rotate CW – Rotate the current image 90 degrees in the clockwise direction. This change persists if the current directory is writable.

- Rotate CCW – As above except the direction of rotation is counter(anti) clockwise.
- Import – Import pictures into your Gallery Dir. This option is described in the next section.
- Settings – Access the MythGallery settings screen.

14.3 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:

```csh
#!/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
```

15. MythGame.

16. 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

16.1 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
$
```

16.2 Mandrake

Mandrake 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://addmedia.linuxfornewbies.org/ 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 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 it may be installed by typing:

```
# urpmi libMesaGLU1-devel
```

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

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

16.3 Red Hat Linux 9

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.

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

16.4 Compiling MythMusic

Once all the prerequisites have been installed, you can proceed with compiling MythMusic.

```
$ cd ~/mythplugins-0.18.1
$ ./configure --disable-all --enable-mythmusic --enable-fftw --enable-sdl --enable-aac
$ qmake mythplugins.pro
$ make
$ su
# make install
# exit
```

16.5 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 configuration.

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.

16.6 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.

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 (i.e. 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.

16.7 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.

17. MythWeather.

MythWeather is a part of the **mythplugins** package. See MythPlugins for instructions on downloading the tarball.

```
$ cd ~/mythplugins-0.18.1
$ ./configure --disable-all --enable-mythweather
$ qmake mythplugins.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.

18. MythVideo.

MythVideo is a part of the **mythplugins** package. See MythPlugins for instructions on downloading the tarball.

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

```
$ cd ~/mythplugins-0.18.1
$ ./configure --disable-all --enable-mythvideo
$ qmake mythplugins.pro
$ make
$ su
# make install
# exit
```

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

19. MythDVD.

MythDVD is a part of the **mythplugins** package. See MythPlugins for instructions on downloading the tarball. 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.

19.1 Manual Compilation of Prerequisites

19.2 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 prerequisites using the following command:

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

In the example below, we have enabled support for transcoding and for VCD playing. You may remove these options if you don't need them.

```
$ cd ~/mythplugins-0.18.1
$ ./configure --disable-all --enable-mythdvd --enable-transcode --enable-vcd
$ qmake mythplugins.pro
$ make
$ su
# make install
```

19.3 Running the Myth Transcoding Daemon

Transcoding ("ripping") a DVD requires you to run the Myth Transcoding Daemon (mtd). To ensure that mtd is configured correctly, you should first test it at the command line.

```
$ mtd -n
```

The last line of text should show something like:

```
mtd is listening on port 2342
```

This indicates that mtd is ready for use. Once you've successfully tested mtd in the foreground, type **CTRL−C** to stop mtd. You may then start it as a background (daemon) process.

```
$ mtd -d
```

Running mtd as a daemon will allow you to automatically start it during the boot process. For example, you may add mtd -d to your rc.local file, or you can adjust the script/steps outlined in the section called Automatically starting mythbackend at system boot time to start mtd instead of mythbackend.

20. MythNews.

MythNews is a part of the **mythplugins** package. See MythPlugins for instructions on downloading the tarball. MythNews is a RSS reader.

```
$ cd ~/mythplugins-0.18.1
$ ./configure --disable-all --enable-mythnews
$ qmake mythplugins.pro
$ make
$ su
```

```
# make install
```

21. Troubleshooting.

21.1 Compiling

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.

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.

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.

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.

Internal Segmentation Fault.

This is most likely to be caused by an overheating processor rather than an actual programming fault within gcc.

21.2 Debugging

MythTV segfaults

MythTV isn't doing anything

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.

Re–run the **configure** script and add `--compile-type=debug` to any previous configuration options you may have used. Check the `config.log` file if you have forgotten.

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 distclean
$ ./configure --compile-type=debug
$ 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 ./mythtv-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
```

All of the output from gdb 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.

Illegal Instruction

This error used to occur when users compiled MythTV with incorrect CPU options. This should no longer occur with the new **configure** script.

MythTV is crashing your system

When run as a non−privileged user, MythTV *can not* crash your system. 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.

If you are running as root, which is **strongly discouraged**, it is possible that your system may crash due to the real−time thread using all available CPU. You will not be able to interrupt the process, so for all intents and purposes your computer will have crashed.

21.3 Installing

When trying to run setup, you 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.

21.4 Using

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. Also verify that `bind-address` is set to your IP

address instead of `127.0.0.1`. If you change either of these items, restart **MySQL**.

MySQL database is corrupt

If you have reason to believe that your MySQL database is corrupt, execute the following commands to attempt to repair it.

NOTE: Ensure that there are no programs accessing the database while you attempt to repair it. Make sure that all backend and frontend programs have exited.

```
mysqlcheck -r -umythtv -p<password> mythconverg
```

Using a MPEG−2 encoder card and the video appears "jittery"

Using a MPEG−2 encoder card and the video is jumping up and down

This is a different problem than the one discussed in the previous section. Currently, the ivtv driver or firmware appear to have some issues if the vertical capture resolution is not the full screen height. If you are having a jitter problem then ensure that you are capturing either 480 lines (for NTSC) or 576 lines (for PAL). The default capture profiles may need to be edited for your setup. Go to Settings−>TV Settings−>Recording Profiles and adjust the **Default** and **Live TV** options to 480 or 576 from their defaults.

Screen goes blank but returns when mouse is moved or keyboard is used

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 due to a screensaver that has defaulted to a blank screen. MythTV now has DPMS support built−in, and should intelligently handle the screen. Continue reading if you wish to override DPMS and force it off.

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` or `/etc/X11/xorg.conf` file, and look for:

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

    Option "blank time" "0"
    Option "standby time" "0"
    Option "suspend time" "0"
    Option "off time" "0"
    Option "NoPM" "1"
EndSection
```

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.

21.5 Miscellaneous

mythfilldatabase failing

If mythfilldatabase suddenly appears to be failing, there are at least two things to check.

First, if you are in North America, ensure that your DataDirect subscription is still valid, otherwise, check to

see what version of XMLTV you're running and that it's the latest version.

STOP **NOTE**: It is highly recommended that you run the latest version of XMLTV available. Your listings provider may have made changes which negatively impact XMLTV.

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.

Frontend appears to be slow at jumping / seeking.

On–screen Display shows incorrect program length.

This may occur when MythTV doesn't have an accurate seek table. Run **mythcommflag −−rebuild**

Troubleshooting audio

STOP **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 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 real–time video. You should see messages about "underrun". These can be ignored but they do confirm that the driver is loaded and there is an active device. 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**. If you are using an ALSA version after 1.0.6, use **alsamixer –V all**:

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` , or whatever device file is being used, hasn't already been grabbed by another process, like **esd** or **artsd**. If the device file isn't available, then MythTV won't work. You may wish to run **configure** and enable support for these.

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.

Mythbackend reports that your card is not reporting full duplex capabilities

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 hardware encoder card such as the Hauppauge PVR–250, DVB cards or HDTV capture cards as your recording source and you're only using the playback function of your sound card, 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 sound card to go full–duplex and need it to, then check your distribution for updated sound drivers. If your sound card 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.

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")

NOTE: You do not need to use **irxevent** if you are using MythTV's native LIRC support, so the window manager focus issue does not apply in that case.

PVR–250 card doesn't show inputs in the setup screen.

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

Where's "canada–cable"?

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, i.e. 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.

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.

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.

What does "strange error flushing buffer" mean on the 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.

Can't change the channel when watching Live TV.

Something's wrong with your program database. Did mythfilldatabase run with no major errors? Or, MythTV may not have permissions to the appropriate video4linux devices. See the section titled Device Permissions for an example.

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.

Screen goes black when you 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, or, if you're running from a startup script, check the log file.

Poor performance with NVidia cards and XvMC

XvMC is a NVidia driver feature which is supposed to help with decoding video. Users have reported that rather than speeding up their video it appears to be doing the opposite. You may want to check that your color depth is set for 24bpp.

Computer is loading a media player application when you insert a CD or DVD

You need to disable any sort of auto–running media player in your environment, otherwise MythDVD or MythMusic will not be able to work properly.

In KDE, you may want to perform the following:

```
$ rm ~/.kde/Autostart/Autorun.desktop
```

22. Miscellaneous.

22.1 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
```

STOP **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
```

22.2 Using a different 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.

22.3 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 horizontal lines stacked vertically.

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

Horizontally, the maximum value allowed for a Bt8X8 chip is 720. However, due to limitations in the chip and other limitations of broadcast television, there may not be a noticeable improvement in image quality beyond 400 or 500 pixels.

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

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

- 720x480 ("DVD" resolution. Also commonly, 704x480)
- 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.

22.4 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 `/etc/X11/XF86Config-4` for XFree, or `/etc/X11/xorg.conf` for Xorg, so that 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.

STOP **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.

22.5 Saving or restoring the 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.

22.6 Deleting 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
```

22.7 Moving your data to new hardware

This assumes that you will be moving your data to newer / bigger hardware and don't want to lose your programs.

The first step is to create a database backup as demonstrated in an earlier section.

Next, you will extract only the data that is relevant to the programs:

```
$ grep "INSERT INTO record "         mythtv_backup.sql > restore.sql
$ grep "INSERT INTO recorded "       mythtv_backup.sql >> restore.sql
$ grep "INSERT INTO oldrecorded "    mythtv_backup.sql >> restore.sql
$ grep "INSERT INTO recordedmarkup " mythtv_backup.sql >> restore.sql
```

Note the space after the table name and the ">>" to append to the file for all but the first grep. "recordedmarkup" is huge and may be hundreds of thousands of lines if you had lots of hours of recordings.

After you have moved the data files to the new hardware, configure MythTV using the **mythtv−setup** program as you normally would with a standard MythTV installation.

At this point we will restore the information about your programs back into the database:

```
$ mysql −u mythtv −pmythtv mythconverg < restore.sql
```

After successful insertion of the data you may delete the `restore.sql` file.

22.8 btaudio

btaudio allows you to obtain the audio data from your tuner card directly over the PCI bus without using a sound card. This is useful if you would like to use multiple tuner cards in a system without adding a sound card for each one, or if your existing sound card 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)

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

22.9 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 grabber configuration 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.

If you are using the DataDirect service, login to your account at http://labs.zap2it.com/ to modify your lineup. Uncheck the boxes for any unwanted channels, and they will no longer be included in your download.

If you are using a grabber from XMLTV, comment out the channel from the `~/.mythtv/<sourcename>.xmltv` file by inserting the word "not " (including the space) in front of the unwanted entry. This will prevent **xmltv** 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. To delete the data from the database we need to perform some steps. First, assuming that HBO is channel 15, we need to find out the internal `chanid` used by MySQL:

```
$ mysql -u root mythconverg
mysql> select chanid from channel where channum=15;
+--------+
| chanid |
+--------+
|   1015 |
+--------+
1 row in set (0.00 sec)
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;
```

22.10 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,hard,intr,nfsvers=3
```

Then run # `mount -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; "hard,intr" is there because that's the recommendation of the NFS−HOWTO, and the nfsvers is required for filesizes over 2GB.

22.11 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, BorderWidt
^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.

22.12 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

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
```

22.13 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 Master backend 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 unavailable

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.

22.14 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.

22.15 Changing your hostname

If you need to change the name of the computers used with MythTV you'll need to perform a sequence of steps. There are a number of pieces of information that MythTV keeps track of which are tied to the hostname of the box, so changing the hostname involves altering the name in the operating system and in the MySQL database. In the examples below, the old name of the system was "frontend1" and we're going to change it to "kidsroom".

STOP **NOTE**: Changing the hostname using direct SQL update commands will break things. You *MUST* use this indirect method.

1. Stop all backends. If you run **mythbackend** from a terminal session, press control−c. If your backends are started with an init script, you would do something like the following:

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

2. Change the hostname.

For Red Hat and derived distributions, edit the `/etc/sysconfig/network` file. Look for `HOSTNAME=frontend1` and change this to `HOSTNAME=kidsroom` or whatever you'll be using. For other distributions, refer to the documentation, such as the

`hostname(1)`

man page.

To alter the hostname in the current session, run:

```
# hostname kidsroom
```

3. Dump the database.

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

4. Rename the host in the database. First, ensure that the new hostname you'll be using isn't already in the database.

```
$ grep kidsroom mythtv_backup.sql
```

Now we're actually going to change the name. The following should all be typed on the same line:

```
$ cat mythtv_backup.sql | sed s/\'frontend1\'/\'kidsroom\'/ >> mythtv_restore.sql
```

If you don't feel comfortable using **sed**, you can open the `mythtv_backup.sql` file in a text editor and perform a global search and replace. When saving the file, make sure you use the new name, `mythtv_restore.sql` or the rest of the steps below will fail.

5. Drop and recreate the database.

```
$ mysql −u root
mysql>drop database mythconverg;
```

```
mysql>create database mythconverg;
mysql>exit
```

6. Restore the database using your edited version.

```
$ mysql -u mythtv -pmythtv mythconverg < mythtv_restore.sql
```

If you are running slave backends or frontends, don't forget to re−enable access as detailed in Modifying access to the MySQL database for multiple systems.

7. Start the backends. If you use init scripts, do the following, otherwise start them from terminal consoles.

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

8. Quit and restart all frontends. Delete the `mythtv_backup.sql` and `mythtv_restore.sql` files.

22.16 Can I run MythTV on my TiVo?

22.17 Can I run MythTV on my ReplayTV?

No.

While it is true that the TiVo runs the Linux kernel, and TiVo has released their changes to the kernel under the GPL, the TiVo is *not* a general−purpose computer, and there is no programming information available for the custom hardware contained within a TiVo. TiVo is under no obligation to release the source code to their *application*.

The ReplayTV runs VxWorks, a Real Time Operating System from Wind River Systems.

22.18 Can a wireless connection be used between the frontend and the backend?

Yes, assuming that your wireless connection has sufficient bandwidth to maintain the datarate between the frontend and the backend. 802.11b should be sufficient if the encoded bitrate of the content is less than the datarate of your wireless connection, which in the case of 802.11b would be approximately 4 Mbps. (The advertised rate of 11Mbps gives an actual throughput of 4 Mbps.) 802.11a and 802.11g, if operating in their high−speed modes, or proprietary 802.11b "Turbo" schemes should be adequate. Multiple wireless frontends, poor signal strength or other factors can severely impact the viewing experience on the frontend.

22.19 How can I burn shows that I have recorded to a DVD?

There are two answers to this question. The easiest thing to do is to simply burn the MythTV files onto the DVD and treat the DVD as if it were a hard drive. This will allow you to save the programs, but it's likely that they will *not* be playable in a DVD player because the disc will simply contain files and not any DVD structure. If you'd like to create a DVD that will play in a standard DVD player and your files have been recorded as MPEG−2 from a device supported by the IVTV driver, the first step you'll need to perform is to remux the file to make it DVD compliant. Next, you'll use **dvdauthor**, **mkisofs** to create the ISO image, and finally **dvdrecord** to burn that to a blank DVD−R.

The requirements for this procedure are **mjpegtools** (provides **mplex**) and **dvdauthor** (provides **mpeg2desc** and **dvdauthor**).

This script will perform the necessary steps. You'll need to pass it the .nuv file and a name for the new DVD project. You should end up with a directory which you can **mkisofs** into a ISO file then burn it.

```
#!/bin/sh

mkfifo aud0
mkfifo vid0
mkfifo dvdmpg

mpeg2desc -a0 < $1 > aud0 &
mpeg2desc -v0 < $1 > vid0 &
mplex -f 8 -V -o dvdmpg aud0 vid0 &
dvdauthor -o $2 -f dvdmpg
dvdauthor -o $2 -T

rm aud0
rm vid0
rm dvdmpg
```

It's also possible to pipe the output of **mkisofs** directly to **cdrecord** without any intermediate files being written to your hard drive. For example:

```
#!/bin/sh
TSIZE=`mkisofs -dvd-video -udf -q -print-size .`
mkisofs -dvd-video -udf -V "$1" . | nice -10 cdrecord speed=1 dev=1,1,0 \
driveropts=burnfree -dao -v tsize="$TSIZE"s -
```

22.20 Burning a MythTV recording to a DVD (alternate version)

This is specific to MPEG–2 .nuv files created by Hauppauge PVR–X50 video capture cards.

The software packages you will want are:

- dvd+rw–tools
- dvdauthor
- mjpegtools
- avidemux2
- transcode – not necessary for this, but useful for working with video
- k3b – a GUI for burning DVD's and CD's. Not necessary, but useful.

1. Find the file you want.

2. Run **avidemux2** and open the .nuv file which corresponds with the program you wish to burn to DVD.

3. Remove the segments you don't need. Click twice on "A" when you are at the beginning of the segment you want to save, find the end of the segment, and click "B".

4. Save the raw video stream and audio: `File-> Save -> Save Raw Video Stream`. Save the file as `1.m2v`. When that's done, go to `Audio -> Save Audio`. Save the file as `1.mp2`

5. Repeat steps 2 & 3 until you have saved all the segments you wish to keep.

6. Mplex the video and audio for each segment together in DVD−MPEG2 format. Do that by running

```
mplex -f 8 -o 1.mpg 1.m2v 1.mp2
```

7. Create a DVD file system by running

```
dvdauthor -o DVD 1.mpg 2.mpg ... n.mpg
```

```
dvdauthor -T -o DVD
```

8. Put a blank DVD in your drive and write the video by running

```
growisofs -Z /dev/scd0 -dvd-video DVD
```

9. Be sure and check the DVD in a player and then delete all the files you created in this process.

Notes:

You can reduce the size of your files and fit more video on a DVD by using **tcrequant**, which is part of **transcode**.

The steps above create a DVD that starts playing the video automatically. If you want to burn a DVD with menus and multiple video titles, see the next section.

22.21 Creating a DVD with Menus

This assumes that you have already formatted the videos in a DVD−compliant format that works with **dvdauthor**.

CREATING A MENU

This is potentially the most difficult step in creating a DVD with a menu. You must capture or create a background image with the appropriate resolution and **mplex** it together with an audio file (typically silence, although though you may use sound) and then create overlay images that show

1. the menu text
2. the highlighted buttons
3. the selected buttons

You would then use **spumux** to combine the background video with the overlay menus.

The background image must be saved as a JPG and the overlay images must be PNG with a maximum of three colors. Four colors are allowed, but the transparent background counts as one of the available colors. In these examples, the background file is called `menu.jpg`, the menu text is called `menu1.png` and the highlighted and selected buttons are the same and called `menu2.png`.

Create a Background

To get a background image, use **avidemux2** and find the frame you wish to use as a background image. Save the frame as `menu.jpg` by executing `File -> Save -> Save JPG Image`.

You may also create your own background image in the GIMP. Use the following settings:

For NTSC Video:

```
width: 720
height: 480
x-axis: 81dpi
y-axis: 72dpi
```

Convert the background JPG file into a M2V video file

```
jpeg2yuv -n 50 -I p -f 29.97 -j menu.jpg | mpeg2enc -n n -f 8 -o menu.m2v
```

For PAL Video

```
width: 720
height: 576
x-axis: 75dpi
y-axis: 80dpi
```

Convert the background JPG file into a M2V video file

```
jpeg2yuv -n 50 -I p -f 25 -j menu.jpg | mpeg2enc -n p -f 8 -o menu.m2v
```

Merge the background you just created with audio

You will need some sort of audio to **mplex** with the M2V file you just created, so either capture some audio using **avidemux2** or another program, or to create empty audio to **mplex**, use this command

```
dd if=/dev/zero bs=4 count=2000 | toolame -b 128 -s 48 /dev/stdin menu.mp2
```

Merge the M2V video with the MP2 audio

```
mplex -f 8 -o menu.mpg menu.m2v menu.mp2
```

Create the menu buttons

Open the background `menu.jpg` file in the GIMP. Add a transparent layer over the background and mark where you will want the different buttons to be located. This layer will later be deleted, so it doesn't have to be clean. When creating this layer, use all the colors you want to use in your buttons. Remember, no more than three colors.

Delete the Background layer.

Change the image to an indexed color mode. Do this by right–clicking on the image and selecting `Image ->
Mode -> Indexed`. Enter "4" in the number of colors.

Create two new layers.

In the top layer, you will enter the text for your buttons. Select the color you want for your text either in the
color palette or by using the Color Picker Tool. To use the Color Picker, you must make sure you have that
bottom layer selected. Enter the text for all the buttons on the top layer according to whatever layout you
decided on.

Now you will create the actual buttons by drawing boxes around the text and filling that area with the Color
Fill Tool. Be sure and select the color you want (contrasting to the text) and to create your boxes in the second
from the top layer.

Delete that bottom layer which we stated earlier would be deleted.

Save the current image as `menu2.png`. Do that by right–clicking and selecting `File -> Save as`, and
make the name `menu.png`. Click "Export" and make sure that "Save Background Color" and "Save
Resolution" are selected.

Select the bottom layer and change the Opacity to 0%. Now save the image as `menu1.png`.

Create the final menu video

To create the DVD menu, an XML file is used. An example `menu.xml` file is shown below. The autoorder
line is used to decide what takes precedent for ordering. If your buttons are ordered primarily by rows, then
select "rows". If your ordering is by column, then choose "columns".

```
###begin menu.xml###
<subpictures>
  <stream>
    <spu
     force="yes"
     start="00:00:00.00"
     image="menu1.png"
     select="menu2.png"
     highlight="menu2.png"
     autooutline="infer"
     outlinewidth="6"
     autoorder="rows"
    >
    </spu>
  </stream>
</subpictures>
###end menu.xml###
```

Then create the final menu with

`spumux menu.xml < menu.mpg > menu_final.mpg`

You should see some lines in the output to the console that show

INFO: Autodetect ...

The number of these lines that you see should be the same number of buttons you created. If there are, then good job!

Authoring the DVD

To create the DVD structure, an XML file is used. The below file is an example `dvd.xml` file. This example file allows for the 4 buttons in the menu created earlier and has three video segments making up each of the 4 video titles. Each of the separate video segments will be treated as separate chapters of one consistent video title.

```
### begin dvd.xml ###
<dvdauthor dest="DVD">
  <vmgm>
    <menus>
      <pgc>
        <button>jump title 1;</button>
        <button>jump title 2;</button>
        <button>jump title 3;</button>
        <button>jump title 4;</button>
        <vob file="menu_final.mpg" pause="inf" />
      </pgc>
    </menus>
  </vmgm>
  <titleset>
    <titles>
      <pgc>
        <post>call vmgm menu 1;</post>
        <vob file="1a.mpg" />
        <vob file="1b.mpg" />
        <vob file="1c.mpg" />
      </pgc>
    </titles>
  </titleset>
  <titleset>
    <titles>
      <pgc>
        <post>call vmgm menu 1;</post>
        <vob file="2a.mpg" />
        <vob file="2b.mpg" />
        <vob file="2c.mpg" />
      </pgc>
    </titles>
  </titleset>
  <titleset>
    <titles>
      <pgc>
        <post>call vmgm menu 1;</post>
        <vob file="3a.mpg" />
        <vob file="3b.mpg" />
        <vob file="3c.mpg" />
      </pgc>
    </titles>
  </titleset>
  <titleset>
    <titles>
      <pgc>
        <post>call vmgm menu 1;</post>
        <vob file="4a.mpg" />
        <vob file="4b.mpg" />
        <vob file="4c.mpg" />
```

```
            </pgc>
          </titles>
        </titleset>
      </dvdauthor>
      ### end dvd.xml ###
```

Create the DVD file structure with

```
dvdauthor -x dvd.xml
```

Test the created DVD file structure with

```
xine dvd:/full/path/to/DVD/VIDEO_TS/
```

Then burn the DVD file structure to a DVD with

```
growisofs -Z /dev/scd0 -dvd-video DVD
```

22.22 Migrating from XMLTV to DataDirect

DataDirect will use different information for your video sources and channels than was used by XMLTV. The following will clear your old information in a way that will allow your EPG and existing recording rules to continue to work as you'd expect.

The first step is to create a backup of your database using the instructions found in Saving or restoring the database

NOTE: Examine the contents of the backup file to be sure that the backup succeeded before modifying your database.

NOTE: The MythTV DataDirect grabber requires **wget** version 1.9.1 or later. Verify that this is installed before modifying your database.

Next, start **mysql** and perform the following commands:

```
$ mysql -u root mythconverg
mysql>CREATE TABLE channel_copy SELECT * FROM channel;
mysql>exit
$
```

Run the MythTV setup program. When the setup program prompts "Would you like to clear all program/channel settings before starting configuration [no]" type "yes".

Complete the configuration as detailed in the section called Configuring the Zap2It.com DataDirect service and in Video Sources.

You *must* associate your DataDirect information with the inputs on your capture device. If you created "Cable−DD", then don't forget to bind /dev/video0 to Cable−DD in the Video Sources portion of setup.

Once you're done running setup, run **mythfilldatabase** to populate the database with new data.

To restore the channel icons, you will need to perform one of two steps. Both are dependant on the channel_copy table you created above.

If you're running MySQL v3, execute the following. Note that the select command is very long. You can press ENTER at the end of each line; the command won't be evaluated until you end the statement with a semicolon.

```
$ mysql -u root mythconverg
mysql>create table channel_temp
select c.chanid, c.channum, c.sourceid, c.callsign, c.name, cb.icon,
c.finetune, c.videofilters, c.xmltvid, c.contrast, c.brightness,
c.colour, c.freqid, c.recpriority, c.hue, c.tvformat,
c.commfree, c.visible, c.outputfilters
from channel c
left join channel_copy cb on (c.callsign = cb.callsign);
mysql>truncate channel;
mysql>insert channel select * from channel_temp;
mysql>drop table channel_temp;
mysql>drop table channel_copy;
mysql>exit
$
```

If you are running MySQL v4, it's a little simpler. The next command is one long string:

```
$ mysql -u root mythconverg
mysql>UPDATE channel,channel_copy SET channel.icon = channel_copy.icon
WHERE channel.callsign = channel_copy.callsign;
mysql>drop table channel_copy;
mysql>exit
$
```

22.23 What do the icons on the Watch Recordings screen mean?

Press "1" or F1 to get a popup.

22.24 What is the difference between a PVR−250 and a PVR−350

This is covered in the hardware section, and extensively covered on the Hauppauge website. Briefly recapping:

A PVR−250 (model 980) is primarily a MPEG−2 encoder. It has a tuner, composite and SVideo connection for video, and an 1/8" audio jack for stereo analog audio. It also has an IR receiver and comes with a remote control.

The PVR−250 MCE (Media Center Edition, aka model 975)) contains a FM radio tuner. The tuner is currently not support by MythTV. The PVR−250 MCE does not contain a IR receiver or a remote.

The PVR−250 Rev 1 contained an MPEG−2 decoder. However, this function was not connected to any output jacks, and there doesn't appear to be any way to pull decoded video from the card, so it's a fairly useless feature.

The PVR−350 (model 990) has the features of the PVR−250 as well as being able to decode MPEG−2. The encode and decode functions may be used simultaneously. The MPEG−2 decoder function gives superior video quality. However, the decoder function is only available once Linux has started, so you will not see any boot−time messages.

Also, the card does not support any 3D functions, making it a poor gamer card.

Non−MPEG−2 content will be displayed on the PVR−350 output using framebuffer features, placing a large load on the host CPU.

Note that for the PVR−350 there are some considerations regarding the way audio is handled.

22.25 Changing channels on an external Set Top Box

If you need to use an external Set Top Box (STB), such as for satellite TV or for digital cable you will need some way for MythTV to tell the STB to switch to a new channel. There are several methods:

1. Use an IR blaster. An IR blaster is an infrared transmitter connected to your computer. When MythTV needs to change channels it will send IR pulses, thereby emulating a remote control.
2. Use a direct serial connection. Some STB's have a serial port on the back, although it may not look like a serial port. It may look like a phone jack, or a strange VGA connector. It may be labeled "Low Speed Data". A direct serial connection is more reliable than an IR blaster. Not all STB's that have a Low Speed Data port have it enabled; you may need to convince your service provider to turn it on. Stating that you have a Tivo may help; the Tivo has a direct−connect capability.
3. Use a firewire connection. There is a `6200ch.c` in the MythTV contrib directory which may work for you.

22.26 Configuring one machine to flag all commercials

Commercial flagging can be CPU intensive. By default, the backend that created a recording is the one which will flag commercials. You may wish to use a different machine to run commercial flagging.

On the slower machine:

Start the backend setup program. Advance through the pages until you get to the Job Queue page. Turn off the setting that says "Allow Commercial Detection jobs", thereby preventing any commercial flagging jobs from running on this machine.

Next, make sure that "Run Jobs only on original recording host" is turned OFF so that new jobs are allowed to run anywhere.

Restart **mythbackend** since it only reads this setting when it starts up.

On the faster machine:

Start the backend setup program. Advance through the pages until you get to the Job Queue page. Ensure that "Allow Commercial Detection jobs" is turned ON for this machine.

Run **mythjobqueue**. **mythjobqueue** will examine the JobQueue and run any jobs it finds. **mythjobqueue** should be left running so that it will pick up any new commercial flagging jobs that are added to the queue,

otherwise new jobs will be added to the queue and your programs won't be flagged until you run manually run **mythjobqueue**.

23. Example Configurations.

This section contains configurations which you may find useful.

23.1 Hauppauge PVR−250/350 hardware MPEG−2 encoder

STOP **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/ 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 (i.e. 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
```

STOP **NOTE**: You do not need to recompile your kernel. Download the latest IVTV driver from Sourceforge. Check http://sourceforge.net/projects/ivtv/ for the current version. Right now (2004−01−21), it's 0.1.9. Untar the file.

```
$ tar −xzf ivtv−0.1.9.tar.gz
```

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

```
$ cd ivtv/utils
$ wget http://hauppauge.lightpath.net/software/pvr250/pvr250_17_21288.exe
$ su
# ./ivtvfwextract.pl pvr250_17_21288.exe
# exit
$ cd ../driver
```

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.

ivtv also needs to know how your kernel was compiled. The default Makefile is not configured for the way Mandrake "does things". At the top of the Makefile in the `driver` directory is a line like this:

```
KERNELDIR= /lib/modules/$(KERNVER)/build
```

edit it so that it looks like:

```
KERNELDIR= /usr/src/linux
```

STOP **NOTE**: If you are running a distribution / kernel which is using the new version of i2c (version 2.8.x, found in the latest Mandrake and Cooker) you will need to make the following edit. Look for

```
# uncomment if you use i2c 2.8.0+
#CFLAGS += -DNEW_I2C
```

and remove the "#" from the line containing `CFLAGS`.

Next, compile the driver:

```
$ cd ~/ivtv/driver
$ make
```

Compile the **test_ioctl** program:

```
$ cd ../utils
$ make
$ su
# cp test_ioctl /usr/local/bin
# exit
```

STOP **NOTE**: You may get warnings about i2c during the compile. Ignore them if they look something like this: `#warning Using temporary hack for missing I2C driver-ID for saa7114`

However, if you get an actual compile error where the compile doesn't finish and the error looks like: `saa7115.c:1117: error: unknown field 'inc_use' specified in initializer` then you didn't uncomment CFLAGS as specified above.

Add the following to `/etc/modules.conf`:

STOP **NOTE**: You do not need to add the last line "`add above...`" if you will not be using the IR remote that came with the card.

```
alias char-major-81      videodev
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
```

STOP **NOTE**: You must specify the tuner type manually. See `~/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. At this point we need to ensure that there are no other versions of `msp3400.o` that would be loaded instead of the version that is created by ivtv. Note that in the command below we are using the backtick, usually located on the same key as the tilde " " and not the single quote.

```
$ su
# cd /lib/modules/`uname -r`
# pwd
/lib/modules/2.4.21-0.16mdk
```

```
# find . -name "msp3400*" | xargs rm -f
```

Next, copy the files that were compiled in the driver directory into the appropriate modules directory:

```
$ su
# make install
# depmod -ae
```

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`.

You can now add this card available to MythTV. Go into your mythtv setup directory on the machine hosting the hardware MPEG−2 card. When you add a new capture card, ensure that you are using the `video0` device.

STOP **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
```

23. Example Configurations. 111

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
```

23.2 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.

```
$ 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
```

23.2 Hauppauge PVR−250 remote and MythTV's native LIRC support   113

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.nerim.net/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
```

STOP NOTE: If you receive a compile error stating: `lirc_i2c.c:296: unknown field 'owner'`
`specified in initializer`, you will need to edit the file and comment out a line in the source code. See message http://www.gossamer−threads.com/lists/mythtv/users/40136 for detailed instructions. 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
...
# 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:

Recompile and install:

```
$ make distclean
$ ./configure --enable-lirc
$ 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.

23.3 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.

If you don't understand how to partition a drive, or how to change the partition type you should stop and look at documentation on how to perform these steps.

🛑 **BIG FAT WARNING**: Using an incorrect parameter can make your files inaccessible, prevent your computer from booting, etc. Be careful!

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...)` If you don't see any messages during boot, you may need to install a LVM init script or confirm that you have all of the LVM packages installed from your distribution.

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.

In the following example, you have a 15GB disk. The first 6GB are set as your boot partition. `/dev/hda2` was added as an extended partition, and within that partition you created the `/dev/hda5` linux (ext2) partition.

```
# fdisk /dev/hda

The number of cylinders for this disk is set to 1823.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)
```

```
Command (m for help): p

Disk /dev/hda: 15.0 GB, 15000330240 bytes
255 heads, 63 sectors/track, 1823 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot    Start        End      Blocks   Id  System
/dev/hda1    *        1        764    6136798+   83  Linux
/dev/hda2             765      1823   8506417+    5  Extended
/dev/hda5             765      1823   8506417    83  Linux

Command (m for help): t
Partition number (1-6): 5
Hex code (type L to list codes): 8e

Command (m for help): p

Disk /dev/hda: 15.0 GB, 15000330240 bytes
255 heads, 63 sectors/track, 1823 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot    Start        End      Blocks   Id  System
/dev/hda1    *        1        764    6136798+   83  Linux
/dev/hda2             765      1823   8506417+    5  Extended
/dev/hda5             765      1823   8506417    8e  Linux LVM

Command (m for help): w

#
```

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 VGforMyth -s 64m /dev/hda5
```

Create a logical volume of 5GB called "video" and then create the reiserfs filesystem and mount it:

```
# lvcreate --name video --size 5G VGforMyth
# mkreiserfs /dev/VGforMyth/video
# mount /dev/VGforMyth/video /var/video
```

Now create a 3GB volume for mythmusic files if you like:

```
# lvcreate --name music --size 3G 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
```

STOP **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
```

23.4 Advanced Partition Formatting

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.

Unlike a typical filesystem, a MythTV video partition is usually a very large filesystem filled with a fairly small number of large files. Filesystem I/O is usually not an issue, even in multi−tuner and/or multi−frontend setups.

There is however, one aspect of filesystem performance that can have a bearing on the performance of

MythTV. In Linux, deleting a file will utilize I/O bandwidth until the deletion has been completed. If deleting the file takes long enough, the video capture buffer may overrun, thereby resulting in dropped frames. Some filesystems are faster at deleting files than others and, for multi−gigabyte MythTV video files, these differences can be significant.

Fortunately, there are published tests (http://aurora.zemris.fer.hr/filesystems/big.html) that provide insight into filesystem performance under conditions relevant to MythTV usage. In addition, some limited testing (archived at http://www.gossamer−threads.com/lists/mythtv/users/52672) with very large files (10 gigabytes) was reported in the MythTV Users mailing list.

Ext2

Ext2 was the defacto standard Linux filesystem for many years. It is stable, provides good I/O performance and can quickly delete large files. The primary disadvantage of Ext2 is that it is not a journaling filesystem, so a file system consistency check (fsck, which is normally only performed after a system crash) can take many hours on a filesystem the size of a typical MythTV partition.

Ext3

Ext3 is Ext2 with a journal, so your biggest gain is that in case of a crash and reboot you won't have to wait very long for your partition to be remounted.

There are options available when formatting an Ext3 partition, as in:

```
# mkfs.ext3 −T largefile4 /dev/hdb1
```

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`.

The "−T largefile4" option creates one inode per 4 megabytes, which can provide a few percent more storage space. However, tests indicate that using the "−T largefile4" option can drastically increase the amount of time required to delete a large file and thus it should only be used with encoder settings that produce small video files (YMMV).

You can check on your filesystem using the **dumpe2fs** program. See the man page for details.

ReiserFS

The Reiser filesystem is another journaling filesystem commonly distributed with Linux. It is known to be an extremely efficient filesystem and it especially excels at managing partitions containing a large number of small files. However, tests indicate it is not the fastest at deleting very large files. For that reason, it may not be the best choice when using encoder bitrates that produce very large files.

JFS

JFS (Journaling File System) is a journaling filesystem originally developed by IBM for AIX which was later released as open source. While not as common as Ext3 or ReiserFS, it is distributed with RedHat 9 (RH9) and Fedora Core 1 (RHFC1) and Mandrake as well as other distros. According to tests, JFS is the file deletion speed king, deleting virtually any file in under one second, even files as large as 10 gigabytes.

XFS

XFS is a journaling file system originally developed by SGI for Irix, and later released as open source. While not a part of the default RedHat Linux 9 or Fedora Core installation (although it is a part of Mandrake and Fedora Core 2), it can be easily installed via ATrpms. XFS provides deletion speeds for large files only slightly slower than JFS. According to the test results shown at (http://aurora.zemris.fer.hr/filesystems/big.html), XFS provide higher I/O rates than JFS, albeit at a higher CPU loading. This may cause issues if you do not have the spare CPU capacity to handle XFS, potentially leading to dropped frames.

23.5 HDTV with the Air2PC−ATSC−PCI card

The Air2PC card has the capability to capture over−the−air HDTV signals (8VSB). The Air2PC card is installed as a DVB device and is supported within MythTV.

First, you must compile a version of the Linux kernel to support the new tuner device.

Download the source to 2.6.11.

```
$ su
# cd /usr/src
# wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.11.tar.bz2
# tar -xjf linux-2.6.11.tar.bz2
# pushd .
```

If you already have a 2.6 kernel running, copy the `.config` file and use it as your starting point. If not, then go straight to the `make menuconfig` portion below.

On Debian with a 2.6.10 kernel:

```
# cp /lib/modules/`uname -r`/build/.config .
# make oldconfig
```

Read the prompts that come up; for the most part you should be able to answer "No".

```
# make menuconfig
```

For Gentoo:

```
# emerge genkernel udev hotplug coldplug
# genkernel --menuconfig --udev --save-config all
```

Go to the Device Drivers−>Multimedia−>Digital Video Broadcasting Devices

Enable DVB for Linux and select B2C2/Technisat Air/Sky/CableStar 2 PCI.

Exit menuconfig, then compile modules and the kernel. (Gentoo will do this automatically as a part of the **genkernel** step.

```
# make modules
# make
# make install modules_install
```

Make any modifications necessary to your boot configuration files (lilo or grub) as required to boot your new kernel.

Download and compile the dvb applications you'll need to configure the card. At the password prompt, press ENTER.

```
$ cvs -d :pserver:anonymous@linuxtv.org:/cvs/linuxtv login
$ cvs -d :pserver:anonymous@linuxtv.org:/cvs/linuxtv co dvb-apps dvb-kernel
$ cd dvb-apps
$ make
```

NOTE: As of 2005–02–16, the compile aborts with an error, however not before the **azap** is compiled, so you should be able to continue.

```
$ su
# cp utils/szap/azap /usr/local/bin
```

On Gentoo, execute the following:

```
# mkdir /lib/firmware
# cp ~/dvb-kernel/firmware/dvb-fe-nxt2002.fw /lib/firmware
# echo skystar2 >> /etc/modules.autoload.d/kernel-2.6
# exit
```

On Debian:

```
# cp ~/dvb-kernel/firmware/dvb-fe-nxt2002.fw /usr/lib/hotplug/firmware
# exit
```

Shutdown your machine, install the card and restart. Ensure that you're using your newly compiled kernel.

Once you've rebooted, confirm that the card is detected. If you type **dmesg**, you should see something like this:

```
drivers/media/dvb/b2c2/skystar2.c: FlexCopIIB(rev.195) chip found
drivers/media/dvb/b2c2/skystar2.c: the chip has 38 hardware filters
DVB: registering new adapter (SkyStar2).
DVB: registering frontend 0 (Nextwave nxt2002 VSB/QAM frontend)...
```

If you are using 8VSB (Over–the–Air HDTV), run the **azap** program. Replace ZIPCODE with your zip code. The perl script will connect to http://www.antennaweb.org to obtain your HDTV channels.

```
$ cd /usr/src/dvb-apps/util/szap/channels-conf/atsc
$ test ~/.azap || mkdir ~/.azap
$ perl ./make_atsc_chanconf.pl ZIPCODE >~/.azap/channels.conf

$ /usr/local/bin/azap wsvn
using '/dev/dvb/adapter0/frontend0' and '/dev/dvb/adapter0/demux0'
tuning to 183000000 Hz
video pid 0x0000, audio pid 0x0000
status 00 | signal fff0 | snr ea60 | ber 00000000 | unc 00000000 |
status 1f | signal f840 | snr ce6e | ber 00000068 | unc 00000000 | FE_HAS_LOCK
status 1f | signal f600 | snr d2d8 | ber 00000018 | unc 00000000 | FE_HAS_LOCK
status 1f | signal f290 | snr cd54 | ber 00000008 | unc 00000000 | FE_HAS_LOCK
status 1f | signal f340 | snr c658 | ber 00000030 | unc 00000000 | FE_HAS_LOCK
CTRL-C
```

```
$
```

NOTE: if you have anything in the unc column ("uncorrectable errors") you will probably have problems with HDTV. Check your antenna, cables, etc.

Once you have confirmed that your card is able to lock onto a HDTV signal it is time to configure MythTV. Start the setup program, and in the Capture Cards section change the card type to "Digital Video Broadcast".

For HDTV you do not need to use the Advanced configuration.