

[Expand everything](#) |
 [Expand non-PVR-350 Output bits](#) |
 [Expand PVR-350 Output bits](#) |
 [Close everything](#)

*** Current setting: *Everything expanded* ***

Step-by-step guide to building a MythTV System on Fedora Core 6 w/ATrpms

Compiled by Jarod C. Wilson <jarod@wilsonet.com>
 Made possible by information gathered from all over the Internet (Google is your friend)...
 \$Revision: 121 \$
 \$Date: 2007-03-26 00:27:51 -0400 (Mon, 26 Mar 2007) \$

TABLE OF CONTENTS:

1. [Introduction](#)
2. [Hardware](#)
3. [Initial System Setup](#)
4. [Firstboot setup -- create MythTV user and enable ntp](#)
5. [Get your system up-to-date](#)
6. [Configure 3rd-party package repositories](#)
7. [Get and install video card drivers](#)
8. [Audio setup](#)
9. [Get and install MythTV](#)
10. [Get and install capture card driver\(s\)](#)
11. [Get and install lirc](#)
12. [Set up MySQL](#)
13. [Set up MythTV](#)
14. [Configure automatic startup](#)
15. [Configuring MythTV add-ons](#)
16. [Upgrading your system](#)
17. [Miscellanea](#)
18. [Trouble-shooting](#)
19. [Changelog](#)

1. Introduction

MythTV is an open-source project, with its primary functionality being similar to that of a TiVo on steroids, with far more features, much greater flexibility and the ability to handle an extremely wide variety of content. The project's founder and leader is Isaac Richards. The official documentation can all be found on MythTV's official web site, at <http://www.mythtv.org/>. The project is supported by the contributed time and effort of the open-source community, with the primary lines of technical support being the documentation on the project web site, and the project's mailing lists. It is highly recommended that if you are intending to install MythTV, join at least the mythtv-users mailing list, which you can subscribe to here: <http://www.mythtv.org/mailman/listinfo/mythtv-users/>, and read through the official docs.

While the official documentation attempts to be as distribution-agnostic as possible, this document is geared specifically toward building a MythTV system on the Fedora Core 6 Linux distribution put out by Red Hat, with all components installed from binary packages using automatic dependency resolution tools (i.e., you won't need to compile anything or manually deal with any twisted dependency issues). Quite a few people following this particular body of documentation have successfully created their own MythTV systems, and many of them frequent

the MythTV Users mailing list. Someone on the list can generally help you if you have a problem. I also frequent the mailing lists, and will help out in any way I can, though my spare time is in very short supply these days. However, please search the mailing list archive, found here: <http://www.gossamer-threads.com/lists/mythtv/>, before you mail the list, because there is a good chance you'll find your answer there. If you are unable to find the answer there, then read this document: <http://catb.org/~esr/faqs/smart-questions.html>, after which you may address your question to the mailing list. ;-)

Versions of this document for older Fedora releases are NOT maintained in any way, but are still available for historical purposes. Just use the latest version, unless you have some very compelling reason to do otherwise. The old versions can be found here:

[Fedora Core 5 version](#)

[Fedora Core 4 version](#)

[Fedora Core 3 version](#)

[Fedora Core 2 version](#)

[Fedora Core 1 version](#)

It was suggested to me by a number of folks that I provide PayPal account information for those wishing to contribute. Donations help persuade my wife that I'm doing more than just wasting my time! Certainly don't feel obligated, but go ahead and click on the little PayPal button if you feel inclined. =)



Anyhow, I try to answer any email I get, but of late, there's no way I can keep up (I'm at least a month and a half behind on replies as I write this). I often get some of them the same questions/problems over and over, so chances are, someone on the list can answer your question, even if you don't find what you needed in the mailing list archive, and you'll likely have a solution faster than I can respond. I used to be very active on the mailing list myself, but just plain haven't had the time lately (work and family has been taking precedent). However, if you think you've found a problem with something in this document, or have a suggestion for an improvement, please don't hesitate to contact me directly via email, and I'll try to get to it when I have the time. I try to keep this document as up to date and accurate as my time permits -- time just isn't permitting much these days... :(

Thank you for reading, and enjoy!

--

Jarod Wilson

jarod@wilsonet.com

2. Hardware

I suggest you start here for information on hardware:

<http://www.mythtv.org/docs/mythtv-HOWTO-3.html#ss3.1>

An excellent place many folks get their Myth box hardware from is [NewEgg.com](http://www.newegg.com).

And those looking at building the quietest Myth box possible should probably check out [SilentPCReview.com](http://www.silentpcreview.com) for advice.

Just about any computer that runs Linux and has enough processing power can be used as a MythTV box. For example, I had zero problems playing back full-resolution PVR-250 recordings on an Athlon 800 with a GeForce 4 MX. I've recently shuffled around a ton of hardware at home, so the systems listed below are merely examples of setups that I'm familiar with and know to be functional.

. **[Dedicated Master Backend System \(not used at all for viewing\):](#)**

- o Case: Antec Full Tower
- o PSU: Antec TruePower 480W
- o Motherboard: Asus P2B-DS
- o CPUs: Dual Pentium III 600MHz
- o RAM: 4x 256MB PC133 DIMMs (1GB of RAM is excessive, but use it if ya got it)
- o Hard Drives: 4x 7200rpm ATA/100 200GB Western Digital Caviar (in software RAID5 for video)
- o Optical Drive: Generic CD-ROM
- o Video Card: nVidia TNT2
- o Sound Card: none
- o NIC: 3Com GbE
- o Video Capture Card: [Hauppauge WinTV PVR-250](#)

High-Definition TV Capture and Playback System:

- o Case: [Antec Sonata](#)
- o PSU: Included [Antec TruePower 380W](#)
- o Motherboard: [Chaintech Summit 7NJL6](#) (full ATX nForce2 Ultra board)
- o CPU: AMD Athlon XP 3200+ w/400FSB (overkill for most systems, but not for HDTV)
- o RAM: 2x 256MB DDR-400 (PC-3200) for 512MB of RAM running in dual-channel DDR mode
- o Hard Drives: 2x 7200rpm SATA 160GB Samsung SpinPoint (highly recommended by Silent PC Review)
- o Optical: Black bezel 16x Sony DVD-ROM
- o Video Card: 128MB 8x AGP Chaintech GeForce FX 5200 (fanless low-profile card)
- o Sound Card: SoundBlaster Audigy OEM, feeding digital coax to a Yamaha amp
- o NIC: Onboard Realtek
- o Video Capture Card: [pcHDTV HD-2000](#)
- o Display: 47" Panasonic HDTV, connected via an [Audio Authority 9A60](#) VGA to component video transcoder

Standard Definition Slave Backend and Playback System:

- o Case: [CoolerMaster ATC-620C-BX1](#) (black desktop micro-ATX)
- o PSU: [Enermax Whisper Series EG365P-VE 350W](#) (adjustable fan speed for low noise operation)
- o Motherboard: [Asus A8N-VM CSM](#) (nForce4-based micro-ATX)
- o CPU: AMD Athlon 64 3500+
- o RAM: 2x 512MB DDR-400 (PC-3200)
- o Hard Drive(s): 7200rpm SATA 80GB Seagate
- o Optical Drive: Black bezel Sony CD-RW/DVD-ROM combo
- o Video Card: PCIe GeForce 6200TC (64MB onboard)
- o Sound Card: SoundBlaster Audigy
- o NIC: Onboard nForce4 GbE
- o Display: 37" Westinghouse LCD High-Definition Television (via DVI at 1920x1080p)

Standard-Definition Remote Playback-only System:

- o Case: [Travla C137](#) (the black one)
- o PSU: external supply included w/case
- o Motherboard: [VIA EPIA M10000](#)
- o CPU: VIA C3 @ 1.0GHz
- o RAM: 256MB DDR-266 (PC-2100)
- o Hard Drive(s): 2.5" 4200rpm ATA/33 10GB Toshiba (laptop HD, requires adapter, but very quiet and low-noise)
- o Optical: Samsung slimline (laptop-style) CD-RW/DVD-ROM combo drive
- o Video Card: Onboard Unichrome/CLE266
- o Sound Card: Onboard
- o NIC: Onboard VIA Rhine-II LAN
- o Display: 19" Monitor

All-in-one Standard-Definition System:

- o Case: [ASUS Pundit](#) (SiS chipset variant)

- o PSU: Included with Pundit
- o Motherboard: ASUS P4S8L
- o CPU: Intel Celeron 2.0GHz
- o RAM: 512MB DDR-333 (PC-2700)
- o Hard Drive: 7200rpm ATA/133 200GB Maxtor
- o Optical: [Sony DRU-500AX DVD+RW/DVD-RW/CD-RW/DVD-ROM/CD-ROM](#)
- o Video Card: Onboard SiS 315 Integrated Graphics
- o Sound Card: Onboard AC '97
- o NIC: Onboard Broadcom BCM4401
- o Video Capture Card: [Hauppauge WinTV PVR-250](#)

A good place to look at other examples of what people are using for their MythTV boxes, and how they rate their systems, is Mark Cooper's PVR Hardware database, browseable here:

<http://pvrhw.goldfish.org/tiki-pvrhwdb.php>

Pundit users might also find Matt Marsh's "Pundit MythTV Device" site of use:

<http://www.mattmarsh.net/computing/pundit/>

Additionally, if you're like me, the SPDIF output being on the front of the Pundit case is a big turn-off. One enterprising reader has put together some notes on how they added one to the back of their case:

<http://www.smashedhat.com/pattjin/pundit/>

Also note:

the WinTV PVR series and AVerMedia M179 cards may have issues running correctly on some motherboards based on VIA chipsets. Check the ivtv FAQ to see if your chipset is on the list of known offenders:

http://ivtv.writeme.ch/tiki-view_faq.php?faqId=1#q33

3. Initial System Setup

This guide is geared toward using Red Hat's Fedora Core 6 Linux distribution. You can download the CD images for Fedora Core from one of the many Red Hat mirrors out there, the official list of which can be found here:

<http://fedora.redhat.com/download/mirrors.html>

As with most any Linux distribution, several file systems are available for you to use, some of which are a bit better suited for use with large files -- like Myth recordings -- than Red Hat's default file system, ext3. However, Red Hat doesn't let you use anything else at system install time, unless you type in "linux <otherfs>" at the boot: prompt when firing up the installer CD, where <otherfs> is something along the lines of reiserfs, jfs or xfs. I'd *highly* recommend a custom partitioning scheme rather than auto-partitioning, with a dedicated /storage (or similar) partition for storage of all your recordings, movies, music, photos, etc. Previously, XFS was my personal recommendation for the file system to use for such a purpose, but some additional stability issues with using XFS (and JFS) on Fedora Core's stock kernels have been brought to my attention. In the case where you're stacking software RAID, LVM and XFS all together, stack overflows tend to crop up. I'm simply running ext3 myself, and have been for ages without a problem. While not the speediest kid on the block, its still fast enough for MythTV use, and its generally a more robust (and better supported by Red Hat) file system than the other options. But if you want the best in performance, you might try your hand with XFS (which I believe has since had most issues remedied). Think of ext3 as your reliable family car, XFS as a sports car... :) If you want to go XFS, boot the installer with "linux xfs", then choose XFS for your /storage partition. An example partitioning setup can be found below.

On the Disk Partitioning Setup screen, choose to Manually partition with Disk Druid. A suitable custom partitioning setup is as follows (assuming a single SATA hard drive):

Partition	Mount Point	Size	Format
/dev/sda1	/boot	100MB	ext3
/dev/sda2	/	8-12GB	ext3
/dev/sda3	swap	same as RAM (ex: 512MB)	swap
/dev/sda5	/storage	Everything else	ext3

If you wanted to use something other than ext3 for your /storage partition, be sure to choose XFS (or JFS or ReiserFS) as the partition format type, as the selection will default to ext3. Note that there's really no point in using anything but ext3 on / and /boot. Red Hat tests ext3 heavily, its the only one that is pretty well guaranteed to be problem-free for / and /boot. Those expecting to add additional hard drives to their system for more video storage might want to set the partition type for hda5 to LVM, then create a logical volume over the top of it, formatted with the file system of your choice. That'll allow you to increase the size of your /storage partition transparently across multiple drives. Details on how to do that can be found in the official MythTV docs, at <http://mythtv.org/docs/mythtv-HOWTO.html#toc24.1>. Note that neither XFS or JFS volumes have the ability to be reduced in size, while ext3 and reiserfs do, which could be of concern in an emergency, should you need to remove a disk from an LVM group for some reason...

On the Network configuration screen, I highly recommend setting a static IP address (could either be truly static, or a statically mapped DHCP address). It really isn't a huge deal if you only have one Myth box, though you probably don't want MythWeb to be a moving target, and it could cause major headaches once you have more than one machine, since non-primary systems wouldn't know where the master backend was anymore if the address changed.

On the Package selection screen, uncheck all the check boxes and select the "Customize now" radio button. At the next screen, you'll be able to re-add the components we need/want. Minimally, add (at least) these package groups to those already selected:

- Desktop Environments -> KDE (K Desktop Environment)
- Applications -> Graphical Internet
- Applications -> Sound and Video
- Servers -> MySQL Database
- Servers -> Web Server

Optionally, you might also want to add these groups:

- Servers -> Windows File Server
- Servers -> Network Servers -> vnc-server
- Development -> Development Libraries
- Development -> Development Tools

Because MythTV is heavily tied to the same toolkit as KDE, we're going to use KDE as the desktop environment on top of which we'll run MythTV. However, you're welcome to try your luck with another window manager. Pretty much any desktop environment/window manager will work, but you may have to make minor deviations from this document in KDE-specific parts. Personally, I run ratpoison instead of KDE on my dedicated MythTV frontend, since I don't need a full-fledged desktop, it loads up faster, and reduces the system memory footprint. If you're short on cpu horsepower and/or memory, switching to something lighter-weight than KDE may be worth the little boost in system performance.

Switching from one desktop environment to another is as simple as selecting a different one via the Session menu on the login screen and saying yes, you want to always login with that desktop environment. Alternatively, you can also use the 'switchdesk' command to make the change. For example:

```
$ switchdesk kde
```

For those that are interested, Jim Chandler has already gone down the WindowMaker path, and contributed [these notes](#), and Greg Hawley has [compiled some notes](#) on setting up Myth on a low-end system, which include notes on using twm. Users on the MythTV mailing lists have written assorted information to the list about using Black Box and RatPoison. My own RatPoison setup is dead-simple, so one of these days, I ought to write something about it...

4. Firstboot setup -- create MythTV user and enable ntp

The first time you start up your system following installation, you should be greeted by the Firstboot setup utility, which contains various modules for configuring bits and pieces of your system.

In the Firewall configuration module, its easiest to simply disable the firewall for minimal headache. The firewall being enabled probably won't cause a problem if you only have one Myth box, but if not properly tweaked, would prevent any secondary systems from being able to contact the master backend. If you really want to enable it, I'd suggest permitting SSH, WWW (HTTP), Secure WWW (HTTPS), as well as adding port 6543, protocol tcp and port 6544, protocol tcp, both of which are used by the mythbackend process. Permitting Samba and NFSv4 may be of interest as well.

In the SELinux configuration module, disable SELinux. There are a number of things that just plain won't work right if you have it enabled. ResierFS doesn't work well with SELinux (known bug in Red Hat's bugzilla) and MythWeb (or more specifically, the web server, apache) can't communicate with MySQL without some work (actually relatively trivial, I just don't have the notes handy...). Personally, I don't care enough about security on my Myth box to want SELinux on it, so I just assume make life easier and disable it. Note that this means your system will restart once you're done with Firstboot.

In the Date and Time module, you have the option to enable the Network Time Protocol (ntp). This is highly recommended. You want your shows to start (and stop) recording at the right times, don't you? ;-) Check the box to enable ntp, and either enter an ntp server you know of near you or leave it with the pre-populated values, which are from the ntp.org pool (which aliases to a bunch of time servers around the globe).

In the Create User module, create a user called mythtv, with a password of your choosing. From here on out, most of this document assumes you are logged into the system as the user mythtv.

The Sound Card module should let you test out your sound card, but there's really nothing that should need to be done here. Good to know if it works, I guess, but if necessary, we'll tweak out the audio later on in this document.

5. Get your system up-to-date

Again, this document assumes you're logged in as the user mythtv. Lines with \$ as the prompt are executed as user mythtv, lines with # as the prompt are executed as root. You should NEVER log in to your machine directly as root. Log in with the normal user account, then assume root privs with 'su' wherever a command needs to be run as root.

For the beginners out there, type 'su' (without the quotes) then hit return, and you'll get prompted for the root password. After entering that, you'll have a root prompt. Do what you need to do as root, then type 'exit' (again, without the quotes) to get back to your normal user prompt.

WARNING, DOUBLE-CHECK THIS NOW: there's a [nasty bug](#) in the FC6 installer that will result in an i586

kernel, instead of an i686 kernel, getting installed on some 32-bit x86 systems. Any 32-bit x86 processor newer than a Pentium I, with the exception of some Via C3 processors, should be running an i686 kernel. You can double-check whether you've been bitten via the following command:

```
# rpm -q --qf='%{name}-%{version}-%{release}.%{arch}\n' kernel
```

If you see i586, follow the directions in the Fedora wiki to remedy the problem:

<http://fedoraproject.org/wiki/Bugs/FC6Common>

While at one time the red-headed step-child of automatic dependency resolution and package download/install tools, Fedora's native yum utility has gotten considerably better with each new release, to the point where its pretty much all I use anymore for installing/updating software on my Fedora boxes. With the addition of a few little files and one plugin, it handles kernel upgrades, including upgrading the associated 3rd-party kernel modules needed on some MythTV systems, quite smoothly.

To get your system fully updated, simply run the command:

```
# yum -y upgrade
```

WARNING:

Do NOT attempt any other rpm activity while the upgrade is in progress. Performing multiple exclusive rpm operations at the same time can lead to very bad things happening (rpm database inconsistencies, race conditions, completely hosed database, etc). After the big upgrade, you MAY get an error about being unable to open the rpm database (this used to happen to me w/RHL9, but hasn't ever with FC). If you do encounter this error, try the following:

```
# rm -f /var/lib/rpm/__db*
# rpmdb -vv --rebuilddb
```

The database rebuild will take quite a while to complete. After it completes, you'll see an error message, but you can safely ignore it and move on to see if the rebuilddb cleared the problem. If not, good luck... Note that the presence of those __db* files doesn't indicate a problem, they're supposed to be there, they just get corrupted every once in a while.

Among the packages upgraded just a minute ago via yum should be your kernel, updated to the very latest errata release. Additionally, your boot loader should have been automatically updated to use the new kernel, so all you need to do is reboot to start using the new kernel, which you'll want to do in just a minute. As of this writing, 2.6.20-1.2944.fc6 is the latest released errata kernel. You should be using at least this kernel, if not a newer one. Note that in FC6 and later, only 32-bit PowerPC has a separate smp kernel, all other architectures use a single kernel for single-processor and multi-processor systems.

Generally speaking, you should always install the latest errata kernel, shortly after the release of which Axel typically has all the needed kernel modules available. Kernel modules are NOT maintained for older kernels, because of two things. First, a new kernel from Red Hat usually fixes some flaw in earlier kernels (security hole or bug fix), so its good practice not to use the flawed kernel. Second, building kernel modules for every kernel would take forever and a day and cause assorted other headaches for Axel. To make providing kernel modules feasible, only the very latest one (or maybe two) kernels are supported.

To make life simpler after we've rebooted and are prepared to start installing drivers for some of our hardware not supported natively by the kernel, we're going to set up a custom environment variable, KVER, that will help avoid problems from typos, user error, confusion, etc. Simply drop a file in /etc/profile.d, like so:

```
# echo "export KVER=\`uname -r\`" >> /etc/profile.d/kver.sh
(those are back-ticks, not single-quotes)
```

Now you're ready to reboot into that new kernel and start installing kernel modules for your tuner card(s), remote, etc., as necessary.

```
# reboot
```

On with the show, once your machine comes back up...

6. Configure 3rd-party package repositories

We won't touch any packages from and 3rd-party repositories until a bit later, but you'll need to tell yum about the ATrpms and FreshRPMs package repositories, and since we were on the subject of yum already, go ahead and set it up now. This is a simple matter of adding a configuration file for each repository in `/etc/yum.repos.d/`. Pre-configured files can be simply downloaded to your system like so:

```
# cd /etc/yum.repos.d/
# wget http://wilsonet.com/mythtv/atrpms.repo
# rpm --import http://atrpms.net/RPM-GPG-KEY.atrpms
# wget http://wilsonet.com/mythtv/freshrpms.repo
```

As hinted previously, note that Axel maintains a few different channels of rpms, including many packages besides those needed for a functional MythTV install, with varying stability labels on them. This entire installation can be done using nothing but packages classified as "stable", but even "testing" is generally pretty stable. The only channel to *really* watch out for is the "bleeding" channel. If you enable it, you've been warned; things may break. You're on your own to fix 'em, so be prepared (Axel appreciates bug reports though -- <http://bugzilla.atrpms.net/>). The primary classifications are as follows (from Axel's web site):

- **stable** packages imply minimal upgrading risk. They are considered very well tested. You will usually find slight modifications to Red Hat's own packages here.
- **testing** packages are provided on a "works for me" basis.
- **bleeding** packages means asking for trouble!

Generally, any problems encountered with ATrpms and ATrpms packages should be addressed on the atrpms-users mailing list before being taken to the MythTV list. You can subscribe to the ATrpms lists here:

<http://lists.atrpms.net/mailman/listinfo/>

FreshRPMs is structured a bit different than ATrpms -- it generally only provides add-on packages, while ATrpms provides both add-on packages packages that replace/upgrade/enhance packages originally provided by Red Hat. FreshRPMs only has one channel of packages. You can subscribe to the FreshRPMs mailing list here:

<http://lists.freshrpms.net/mailman/listinfo/freshrpms-list>

As with any other mailing list, please search their respective archives first (you can find links to them in the same locations as above).

7. Get and install video card drivers

While Fedora Core includes video drivers for most video cards, they aren't always the best ones for the needs of a MythTV user. For example, many video cards require experimental 3rd-party or proprietary drivers to enable TV-Out. Without nVidia's own proprietary video drivers, the performance of cards based on their chipsets will

suffer greatly, even if you aren't using TV-Out. Instructions for installing accelerated drivers for nVidia and SiS cards can be found below. You're on your own for other cards. Click to expand the appropriate section for your video output device...

[For users of the PVR-350's TV-Out:](#)

Initially, you'll need another video output device hooked to a display in order to configure the PVR-350's TV-Out, unless you're extremely comfortable at the command line, in which case you can simply ssh into your PVR-350-equipped box (well, okay, you can also ssh in from another machine running an X server and run X-based apps over ssh's X forwarding, but...). I actually did all the setup of mine while the computer was connected to my TV via TV-Out on my GeForce 4 MX, in run-level 5. That way, I had a shell window and a browser window open, so I could easily cut and paste. Further instructions for configuring the PVR-350 can be found a ways down the page...

[For users of nVidia Video Cards:](#)

nVidia's 9755 release works quite nicely for MythTV boxes with reasonably new nVidia graphics cards (GeForce 4 series and later). Older cards may require the legacy nvidia driver, version 7184, but **SERIOUSLY, GET A NEWER CARD** if you need the old driver. My \$35 GeForce 6200TC works extremely well with anything MythTV can throw at it, and can interface with just about any sort of input on any TV (HDTV, projector, analog tube, monitor, etc) I've seen. Anyhow, to make life easier, Axel has prebuilt kernel modules. Install for your running kernel like so:

```
# yum -y install nvidia-graphics9755-kmdl-$KVER
# yum -y install nvidia-graphics9755-libs nvidia-graphics9755
```

Simply replace the 9755 with another version number, and you can install the older version instead. Note that you can actually have multiple versions installed, and switching between them is as easy as:

```
# init 3
# nvidia-graphics-switch 9755 (or 7184, or whatever else)
# init 5
(you can *probably* get away with simply reloading X at this point, but reboot if
you run into problems)
```

Axel's nvidia rpms insert some necessary lines into modprobe.conf and (as previously indicated) modify your xorg.conf file for use with the nvidia driver, based on your initial xorg.conf file, which is saved for safety's sake as xorg.conf.nvidia.bak. While not necessary, I suggest adding `Load "v4l"` in the "Module" section near the top of your xorg.conf file, to enable optimal v4l to X video transfer and v4l picture controls.

If you're in X right now, simply log out to relaunch X with the new configuration. If you're at the command line, either switch to run-level 5 ("init 5") or run "startx". Once you've verified this configuration is working, you'll likely want to look into the additional options below, to enable TV-Out, accelerated rendering, etc.

Here's an example Device setup section for a card using the 1.0-9755 nvidia driver to output to an NTSC television via S-Video:

```
Section "Device"
    Identifier "Videocard0"
    Driver "nvidia"
    VendorName "Chaintech"
    BoardName "nVidia GeForce 4 MX 440"
    #Option "RenderAccel" "1"
```

```
# TV Out Setup
Option      "TVStandard"  "NTSC-M"
Option      "TVOutFormat" "SVIDEO"
Option      "UseEvents"   "True"
EndSection
```

Those with particularly cantankerous displays and/or people who want to pass their signal through an A/V switchbox may have problems with their card/drivers not auto-detecting the attached display type, in which case, you can add an Option line to the above Device section to force the display type, such as this one:

```
Option      "ConnectedMonitor" "TV"
```

As for overscan, nvidia driver versions prior to 6106 used a fixed value option in the above Device section, which required a restart of X when adjusting. The latest driver versions rely on the nvidia-settings utility to set your overscan with a slider, and you can do so without restarting X. You can also adjust gamma, contrast, color balance, etc., with the nvidia-settings utility.

All the settings are saved on a per-user basis, which I like to restore at login time via a method you can find down in the [Configure Automatic Startup section](#).

Another excellent feature you can activate in nvidia-settings is the flicker filter. It helps to greatly reduce the effects of interlace jitter. With the filter enabled, I no longer bother enabling software deinterlacing in Myth for playback. Note that the flicker filter can only be enabled on TV-Out (SVideo or Composite), not over VGA or DVI.

There are a few settings not explicitly specified in the above configuration, as most of them can be auto-detected. However, it is possible you'll have to force some settings if auto-detection doesn't work. All these options are well-documented on nVidia's web site in the documentation files available in their Linux/Unix driver download area (<http://www.nvidia.com/object/unix.html>).

For reference, and because many people have asked, I'm posting full copies of the xorg.conf files from both my HDTV-connected and SDTV-connected systems of old. My old HDTV system used a custom 960x540p mode for the MythTV GUI and standard-definition recordings and a custom 1920x1080i mode for HDTV content to output to a High-Definition TV through an [Audio Authority 9A60 VGA to Component Video converter](#). I used to feed this set via S-Video, and while the picture was good, it can't touch the signals I'm now feeding the HDTV w/the Audio Authority adapter. My secondary system outputs via S-Video to a plain old 27" analog TV. These days, the only set I have is a 1080p LCD HDTV, which is fed via straight DVI from a Mac Mini (yes, running Linux), which requires basically zero alterations to the xorg.conf that the installer lays down.

```
xorg.conf-HD: http://wilsonet.com/mythtv/xorg.conf-HD.txt
xorg.conf-SVid: http://wilsonet.com/mythtv/xorg.conf-SVid.txt
```

[For users of SiS cards \(and Pundit-S\):](#)

NB: this section hasn't been maintained or updated much in quite some time, so caveat emptor... I'm under the impression the sis driver that ships with Fedora's X server is actually good enough that you shouldn't need to pull one in as detailed below -- I've run HDTV on a recent card using the Fedora X sis driver without a problem.

The original Asus Pundit's onboard video controller is made by Silicon Integrated Systems (the later Pundit-R has an onboard ATI Radeon). Accelerated video is not supported by Fedora Core's native video driver. You'll want to obtain the binary SiS driver, or MythTV performance will suffer greatly (and TV-out won't work right).

Use wget to get the binary SiS video driver

```
$ cd ~/sources
$ mkdir sis_drv
$ cd sis_drv
$ wget http://www.winischhofer.net/sis/sis_drv.o_xorg_gcc3_current.tar.gz
$ tar -xvzf sis_drv.o_xorg_gcc3_current.tar.gz
$ su
# mv /usr/lib/xorg/modules/drivers/sis_drv.so
  /usr/lib/xorg/modules/drivers/sis_drv.so.orig
# cp sis_drv.o /usr/lib/xorg/modules/drivers/sis_drv.so
# exit (Replace /usr/lib with /usr/lib64 for 64-bit systems)
```

Note: <http://www.winischhofer.net/linuxsisvga.shtml> also lists a SiS Display Control Panel, which might be of use, especially for folks that need to tweak their display settings a bit and don't want to deal with command-line text-file hacking. I've used it myself, and it certainly is a handy tool. You can adjust overscan and underscan, flicker reduction, display mirroring, etc., all on the fly without restarting X.

Now run the system-config-display utility to change the video driver.

```
# system-config-display
```

Choose the 'Advanced' tab, then press the 'Configure' button for the video card type.

Choose the 'SiS 650' card. This will set the driver to 'sis'. Also click on 'Custom memory size' to the value you have set in the BIOS (8 to 64M).

Press the 'Ok' button, then press the 'Ok' button again.

Log out of your X session (if you had one active) and log back in for the settings to take effect.

Here is an example /etc/X11/xorg.conf 'Device' section for the ASUS Pundit:

```
Section "Device"
  Identifier   "Videocard0"
  Driver      "sis"
  Option      "XvOnCRT2"    "true"
  Option      "ForceCRT2Type" "SVIDEO"
  #Option     "ForceCRT2Type" "TV"
  #Option     "CHTVOverscan" "false"
  Option      "CHTVOverscan" "true"
  VendorName  "Silicon Integrated Systems"
  BoardName   "SiS 315/650"
  VideoRam    65536
EndSection
```

And finally add `Load "v4l"`

in the "Module" section near the top of your xorg.conf file, for optimal v4l to X video transfer and v4l picture control. For reference, here's an xorg.conf file for the Pundit:

xorg.conf-pundit: <http://wilsonet.com/mythtv/xorg.conf-pundit.txt>

[For users of Via EPIA onboard Unichrome graphics:](#)

The Unichrome graphics chipsets on most Via EPIA systems works with stock X drivers as far as basic functionality, but their integrated mpeg2 decoding assistance capabilities require a somewhat experimental driver from the [openChrome project](#). Pre-packaged drivers and details on setup are available from <http://washington.kelkoo.net/epia/>.

For reference, here is my xorg.conf:

xorg.conf-unichrome: <http://wilsonet.com/mythtv/xorg.conf-unichrome.txt>

8. Audio setup

Note:

outside of setting your audio mixer levels, this step is not essential. Just set your volumes with the mixer program of your choice and move on to the next step if you like. ALSA is the default sound system in the 2.6 kernel, so you've already got it, and your sound card should have been auto-configured during firstboot. If you have some flashy new card, there's a chance you might need a newer ALSA version, which ATrpms typically does provide. Those wanting to install the latest ALSA packages from ATrpms should do so with these commands:

```
# yum -y install alsa-kmdl-$KVER
# yum -y install alsa-driver
```

If you have multiple sound cards you want to use, you may have to edit your `/etc/modprobe.conf` file for your specific card. Details for supported cards can be found here:

<http://www.alsa-project.org/alsa-doc/>

I'm including a copy of my `/etc/modprobe.conf` ALSA configuration excerpt, which is for a SoundBlaster Audigy sound card. This should be valid for SB Live! and Audigy2 cards as well. If your card isn't one of those, check out the [ALSA site's sound card matrix](#) to find information for your card. The differences are very minor for most cards, and usually only the driver name has to be changed. For example, to configure ALSA for use with the onboard audio of an nForce2 board, it should be a simple matter of replacing all instances of "emu10k1" in the following `modprobe.conf` and `.asoundrc` with "intel8x0".

```
# Example modprobe.conf entries for my Audigy
alias snd-card-0 snd-emu10k1
install snd-emu10k1 /sbin/modprobe --ignore-install snd-emu10k1 && /usr/sbin/alsactl
restore >/dev/null 2>&1 || :
remove snd-emu10k1 { /usr/sbin/alsactl store >/dev/null 2>&1 || : ; }; /sbin/modprobe -r
--ignore-remove snd-emu10k1
```

For those folks out there unfortunate enough to need dual sound cards, I'm including a theoretical `modprobe.conf` ALSA snippet for a SoundBlaster Audigy and nForce2 onboard (i810) audio.

```
#Example modprobe.conf entries for Audigy and nForce2 onboard (i810) audio
alias snd-card-0 snd-emu10k1
options snd-card-0 index=0
install snd-emu10k1 /sbin/modprobe --ignore-install snd-emu10k1 && /usr/sbin/alsactl
restore >/dev/null 2>&1 || :
remove snd-emu10k1 { /usr/sbin/alsactl store >/dev/null 2>&1 || : ; }; /sbin/modprobe -r
--ignore-remove snd-emu10k1
alias snd-card-1 snd-intel8x0
options snd-card-1 index=1
install snd-intel8x0 /sbin/modprobe --ignore-install snd-intel8x0 && /usr/sbin/alsactl
restore >/dev/null 2>&1 || :
remove snd-intel8x0 { /usr/sbin/alsactl store >/dev/null 2>&1 || : ; }; /sbin/modprobe -r
--ignore-remove snd-intel8x0
```

Next, create a `.asoundrc` file for your mythtv user (full path of `/home/mythtv/.asoundrc`). Technically, this isn't necessary, but it can help out quite a bit in some cases (especially if you're outputting via an S/PDIF connection), and can't hurt. Anyhow, just make a new text file called `.asoundrc` like so, adjusting for your specific card,

substituting for "emu10k1" where necessary:

```
$ cat > ~/.asoundrc
pcm.emu10k1 {
type hw
card 0
}

ctl.emu10k1 {
type hw
card 0
}
(Now hit Control-D to end cat input)
```

This is a *very* basic .asoundrc file, which worked for me just fine for some time, but I'm currently forging forward with native ALSA digital output in Myth, which requires a bit more complex setup. Mike Dean did an excellent writeup on the matter of digital sound, with details on a fully fleshed out .asoundrc, which you can now find in an evolved and fleshed-out form at:

http://mythtv.org/wiki/index.php/Configuring_Digital_Sound

I put together my own tweaked-out .asoundrc for my Audigy based on the long version for the nForce2 sound card found at the above link. It basically amounted to four changes. You can grab my full .asoundrc here:

<http://wilsonet.com/mythtv/asoundrc.txt>

Note that it is set up with output via S/PDIF and hardware decoding (by an external amp) as the default output method, so adjust accordingly. I'm actually not certain everything in here is correct, but audio on my box does everything it should, so I've left well enough alone. Additionally, this page may be of interest if you're on the subject of passing raw digital audio streams (i.e., Dolby Digital Surround-Sound or AC3 audio) to an external amp:

http://mythtv.org/wiki/index.php/Configuring_Digital_Sound_with_AC3_and_SPDIF

Anyhow, now fire up the audio controller of your choice (aumix, alsamixer, kmix or gnome-alsamixer) and adjust the volumes, inputs and outputs to your liking. While enabling S/PDIF out for my Audigy was a matter of a checking a box in gnome-alsamixer, some folks have reported having to set a particular slider (IEC958 something-or-other?) to zero to enable the S/PDIF output on their sound cards. Go figure.

A simple way to test whether or not you've got sound is to use ALSA's aplay utility, like so:

```
$ /usr/bin/aplay /usr/share/sounds/KDE_Startup.wav
```

For whatever reason, one of my boxes doesn't like to restore all my audio levels, despite my having stored them with an 'alsactl store'... To get around this, along with my nvidia-settings not loading, I used a short shell script you can check out in the [Configure automatic startup section](#) below.

9. Get and install MythTV

Now, here's why we REALLY like automatic dependency resolution and installation tools like yum... MythTV has numerous required dependencies to function correctly, which are automatically taken care of with one simple command:

```
# yum -y install mythtv-suite
```

The last time I looked, that one-liner installed 115 different packages, totalling about 102MB. Just a wee bit easier than hunting down all the rpms by hand, let alone compiling everything from source... :)

10. Get and install capture card driver(s)

MythTV supports a myriad of different video capture cards, and as of version 0.17, firewire input from certain cable boxes as well. See the official MythTV docs at <http://mythtv.org/docs/mythtv-HOWTO-3.html#ss3.1> for details on exactly what are and are not supported. Below you'll find details on setting up a number of popular capture devices under Fedora Core.

Click on the links by the + signs to expand and/or collapse the sections you need. If you've got more than one of the same type of card, the setup process for one card should actually get all cards of that type working. If you've got different types of cards (say one dvb and one ivtv), just configure each of them independent of one another, but take note of which driver loads first, as it may impact which card grabs /dev/video0 versus /dev/video1. Enable or disable display of PVR-350 Output information right here.

[Show PVR-350 Output info](#) | [Hide PVR-350 Output info](#)

*** Current setting: *Showing PVR-350 Output info*

[For ivtv-based cards \(PVR-150/250/350/500, M179, MPG600, etc\):](#)

A full listing of cards supported by ivtv can be found at http://ivtvdriver.org/index.php/Supported_hardware. I've used most of the Hauppauge cards myself, though the PVR-500 is the lone mainstay in my production setup (most stuff I record is in HDTV now).

It is highly recommended that you join up with at least the ivtv-users mailing list and possibly the ivtv-devel mailing list, at least until you get your card(s) working to your satisfaction. Details on joining the mailing lists, along with list rules and etiquette, can be found here:

<http://ivtvdriver.org/index.php/Mailinglists>

Go ahead and install the ivtv driver components from ATrpms like so:

```
# yum -y install ivtv-firmware
# yum -y install ivtv-kmdl-$KVER
```

For PVR-350 output on FC6, you'll need to install the ivtv_xorg package, and as of kernel 2.6.19, you'll also need to install a video4linux kernel module add-on from ATrpms, as for some reason, the saa7127 driver is missing from 2.6.19 right now... This is done simply enough though:

```
# yum -y install ivtv_xdriver video4linux-kmdl-$KVER
```

These days, that's all there is to it. The driver should automatically load on boot and autoconfigure all aspects of the card now. If you run into something not working properly, hit up the ivtv wiki, mailing list, etc.

NOTE: Dutch users are encouraged to check out the forum at <http://mythtv.nl/> for more information.

Users of the PVR-350 (may?) need to add a line to /etc/modprobe.conf if they want to enable TV-Out, like this:

```
# load ivtv-fb for PVR-350 output
install ivtv /sbin/modprobe --ignore-install ivtv; /sbin/modprobe ivtv-fb
```

NOTE: Most of the PVR-350 TV-Out information contained here is derived from the [TvOutHowto on the ivtv wiki site](#), with slight modifications to be Fedora/Red Hat specific (and subsequent changes for 2.6

kernels).

We're going to make little modification to the kernel boot line in your grub.conf file that should force the ivtv frame buffer to load on /dev/fb1, as well as allow the ivtv-fb module to be loaded and unloaded. Without doing this, unloading the ivtv-fb module would probably crash your system. To the end of all 'kernel /vmlinuz...' lines in /boot/grub/grub.conf, append 'vga=791', then reboot your system. This tells the kernel to load a frame buffer for your video card at 1024x768, 16-bit color. I use this all the time myself, simply so I can see more when I'm not in X. I'd always done this on my 350-equipped box without even thinking about it, which could explain some of why I've not run into some of the problems other folks have...

Rather than rebooting to get ivtv loaded, load it up by hand right now:

```
# /sbin/depmod -a
# /sbin/modprobe ivtv
```

If you run into problems along the lines of a message saying memory can't be allocated when trying to either modprobe ivtv or use the card (like cat'ing video from /dev/video0), many folks seem to need to use the following command:

```
echo 16384 > /proc/sys/vm/min_free_kbytes
```

Not sure why this is necessary for some folks and not for others (I don't seem to need it). But if that helps, make it permanent with an entry in /etc/sysctl.conf:

```
# echo "# Fix ivtv memory allocation problems" >> /etc/sysctl.conf
# echo "vm.min_free_kbytes=16384" >> /etc/sysctl.conf
```

Again, the version of ivtv from the ATrpms stable repository should include support for all the latest cards, but occasional new variants pop up. If you do indeed come up with, say, an unknown card or tuner type, you should post a message to the ivtv-devel mailing list, with the entire contents of /var/log/messages from between

```
ivtv: ===== START INIT IVTV =====
```

and

```
ivtv: ===== END INIT IVTV =====
```

and the good folks over there can help set you back on course (I'm on that list too). An easy way to get the required info out of your log file is to run this shell script:

```
#!/bin/sh
tac /var/log/messages |
sed -n '/=\ \ END INIT IVTV\ \ =/,/= START INIT IVTV =/p;
/= START INIT IVTV =/q' |
tac
```

With the ivtv-fb line in your modprobe.conf, your system will automatically load up the ivtv-fb module. If you have that line commented out, you can load the ivtv-fb module up manually, with this command:

```
# /sbin/modprobe ivtv-fb
```

Those intending to run X on their PVR-350's output(s) will want to make note of their PCI bus ID, which is the very first part of that lspci output (the 00:07.0 part in the above example), because it'll have to be specified in your xorg.conf file.

Also check that your video device(s) exist:

```
$ ll /dev/video?
l root root 6 Feb 13 18:23 /dev/video -> video0 lrwxrwxrwx 1 root
root 6 Feb 06 2005 /dev/video -> video0
crw----- 1 mythtv root 81, 0 Feb 06 2005 /dev/video0
crw----- 1 mythtv root 81, 24 Feb 06 2005 /dev/video24
crw----- 1 mythtv root 81, 32 Feb 06 2005 /dev/video32
```

Make a note of what your video device has been set up as (device /dev/video0 in this case), as you will need to inform MythTV of this fact later on. You can determine exactly what device itv is configured on by looking in /var/log/dmesg just after loading the module. For example, you should see something like this:

```
# /bin/dmesg |grep Initialized
itv: Initialized WinTV PVR 250, card #0
```

I believe M179 users may see failure messages regarding i2c. I think the primary use of i2c on the PVR-x50 is for the IR interface, which the M179 does not have. I'm not certain on that one though... And if you've got multiple cards, you'll see multiple lines in dmesg, like so:

```
# /bin/dmesg |grep Initialized
itv: Initialized WinTV PVR 250, card #0
itv: Initialized WinTV PVR 350, card #1
itv: Initialized AVerMedia M179, card #2
```

If it isn't already obvious, card #0 should correspond to /dev/video0, card #1 to /dev/video1, etc.

You'll also need to take note of the frame-buffer device that itv-fb grabs. You can find that out with this command (followed by example output):

```
# cat /var/log/messages |grep "iTVC15 TV out"
Feb 7 21:17:39 htpc kernel: fb1: iTVC15 TV out frame buffer device
```

In this case, the PVR-350's TV-Out grabbed /dev/fb1. Most likely, you'll either get fb0 or fb1. You'll need to know that for your xorg.conf file later on.

The next step is to test the functionality of the TV-Out connection. First, remove the saa7127 module, so we can subsequently reload it with a test pattern. If you only have one input to your TV at your disposal, and you're already using it with your video card's TVOut, it is perfectly safe to unplug that connection and substitute the 350 for it. Simply switch the connections back when you've seen the 350 test pattern:

```
# /sbin/rmmod saa7127
# /sbin/modprobe saa7127 test_image=1
```

If everything looks good at this point, unload the saa7127 module and reload it without the test pattern:

```
# /sbin/rmmod saa7127
# /sbin/modprobe saa7127
```

I've heard at least one report that the test_image parameter doesn't seem to be working these days. I haven't actually used the output on my 350 in ages, so I can't easily confirm or deny that with the lack of spare time I've got these days. You might try forging ahead at this point, even if the test_image test doesn't work. Another test you might have more success with is this, which will attempt to play video off the card right back out to your TV:


```
# /usr/bin/ivtvfbctl /dev/fb0 -noglobalalpha -localalpha
# dd if=/dev/video0 of=/dev/video16 bs=64k
(control-c to stop)
# /usr/bin/ivtvfbctl /dev/fb0 -globalalpha -nolocalalpha
```

Now we'll try a little test capture on the tuner input. Typically, just issue the following command to make sure your card is capturing via the tuner:

```
# /usr/bin/v4l2-ctl -i 0
```

But do note that not all cards have their tuner input at position 0, so take a look at the output of the following command to see what your other options might be:

```
# /usr/bin/v4l2-ctl -n
```

Also, note that if you have multiple cards, understand that v4l2-ctl assumes you're operating on /dev/video0 if no device is specified. On any v4l2-ctl command line, tack on a "-d /dev/videoX" to operate on any other card (i.e. add -d /dev/video1 to work on your second card).

Hook up your cable or antenna to your card's tuner input if you haven't already done so (or some other video source to the S-Video or composite input if you've set a different capture input), and we'll try to capture some video... Press Ctrl-C to stop the video capture.

```
# cat /dev/video0 > /tmp/test_capture.mpg
(ctrl-c to stop capture)
```

Use that copy of mplayer you installed a bit ago (as part of the mythtv-suite install) to view the capture:

```
# mplayer -vo xv /tmp/test_capture.mpg
```

If you get something that looks and sounds intelligible, then congrats, your card is good to go! If not, and you're using the tuner input, it may be that you aren't tuned to a channel you actually have a signal on. Try again after changing channels on your card, rinsing and repeating as necessary:

```
# /usr/bin/ivtv-tune -c [channel]
```

In addition to being able to change channels, ivtv-tune can adjust your frequency table and TV standard as well, which may be necessary in some cases. For example, in the US, there are at three different cable frequency tables, us-cable, us-cable-irc and us-cable-hrc, all of which are slightly different, so you may need to try them all. A summary of what ivtv-tune can do is displayed by the following command:

```
# /usr/bin/ivtv-tune --help
```

To hear audio with this video capture when played back using mplayer and the 350's output, audio output is going to be through your sound card, not the PVR-350's audio outputs. Personally, I piped the 350's audio outputs back into the line in port on my sound card, and then the sound card feeds my speakers, because system, MythMusic and mplayer audio (currently) can only be output through a sound card (but work is underway to change that).

Now that you (hopefully) have video capture working, we'll try to play some video through the PVR-350's

output(s). First, we'll tweak some of the cards settings, then read video in and immediately pipe it out:

```
# /usr/bin/ivtvfbctl /dev/fb0 -noglobalalpha -localalpha
# dd if=/dev/video0 of=/dev/video16 bs=64k
(control-c to stop)
```

At this point, you need to connect the PVR-350's video output to something you can actually see its output on. For those with a single TV input, once again, you can type the commands in, then move your cables around. Note that sound for this test will be fed out the 350's audio outputs. If you see and hear what you'd expect from television, you're ready to keep moving on, so reset the alpha settings on the card (which MythTV handles internally, so you shouldn't have to do this again):

```
# /usr/bin/ivtvfbctl /dev/fb0 -globalalpha -nolocalalpha
```

If you got video, but there were like 20 to 50 evenly spaced horizontal lines overlaying it (also described to me as a lot of yellow and green dots), try tweaking the ivtv-fb driver with this command:

```
/usr/bin/ivtvfbctl /dev/fb1 -alpha -on -globalalpha -nolocalalpha
```

Then run the dd test again, and with luck, the picture will look like it should. If not, rinse and repeat from the beginning...

If things still aren't working at this juncture, please take your problem to the [ivtv mailing lists](#).

You can verify all the settings for your card, see possible inputs, video standards, etc., with various flags to the v4l2-ctl command:

```
# /usr/bin/v4l2-ctl -h
(list all available options)
# /usr/bin/v4l2-ctl --all
(display all available information about current settings)
# /usr/bin/v4l2-ctl -n
(list inputs on the card)
# /usr/bin/v4l2-ctl --list-standards
(list standards -- could be particularly useful to set a specific pal variant)
```

NOTE:

MythTV controls the capture resolution and bit rate internally, so MythTV starts controlling the card, you shouldn't have to worry about any of this gunk.

Running X on the PVR-350's TV-Out-

For starters, recall the PCI bus ID of your PVR-350. Mine was 00:07.0 from the example output above. I'm honestly not certain what the "best" way to do this is anymore. What seems to work reliably is to take the output of `lspci`, convert it to decimal and throw a "PCI:" in front of it, so you'll end up having a line in `xorg.conf` that reads 'BusID "PCI:0:7:0"'. If your bus ID was something like 01:0f.0, you'd use 'BusID "PCI:1:15:0"'. See <https://svn.wilsonet.com/projects/mythtvology/ticket/11> for more coverage on the matter.

You can simply download my (ntsc) `xorg.conf` file for the PVR-350, edit the PCI bus ID and frame-buffer device lines to match your system, and you *should* be good to go. PAL users will have to check out the [ivtv wiki TvOutPal HOWTO](#) for PAL-specific info. You can grab my PVR-350 `xorg.conf` file here:

`xorg.conf-PVR350`: <http://wilsonet.com/mythtv/xorg.conf-PVR350.txt>

To be on the safe side, I highly recommend making a backup of your current `xorg.conf` file before putting

your tailored xorg.conf file into place:

```
# cp /etc/X11/xorg.conf /etc/X11/xorg.conf.pre350
# mv /path/to/customized/xorg.conf-PVR350.txt /etc/X11/xorg.conf
```

Assuming you're still in run-level 5 at this time, connected to your display with your video card's TV-Out, move your cabling and/or switch inputs on your TV over to the PVR-350 and hit ctrl-alt-backspace to kill off X and restart it on the PVR-350. You ought to be greeted by a login screen. If you are, you're in business. If not, you probably did something wrong, so retrace your steps from the beginning of this section. =)

NOTE:

there is more X/TV-Out related info at the start of the ivtv-fb.c file in ivtv/driver if you download the ivtv source code.

Once you actually have MythTV up and running, you'll need to make some changes within the frontend specific to using the PVR-350's output, which are covered in [section 13, Set Up MythTV](#).

The final touch is to rebuild your system's initial ram disk to include the drivers for the 350's output, so that you can have the driver load early enough in the boot process that rhgb will work over it. First up, you'll need to make some changes to /sbin/mkinitrd. A patch that adds an extra flag "--ivtvdev" can be found [right here](#). This patch hasn't yet been heavily tested, but should do the job, if I haven't lost my marbles (again) -- feedback welcomed. Download the patch, make a backup copy of mkinitrd and apply the patch:

```
# wget http://wilsonet.com/mythtv/mkinitrd-ivtv.patch
# cp /sbin/mkinitrd /sbin/mkinitrd-ivtv
# patch /sbin/mkinitrd-ivtv < mkinitrd-ivtv.patch
```

Now build your initial ramdisk:

```
# mkinitrd-ivtv -f --ivtvdev /boot/initrd-`uname -r`.img `uname -r`
```

After the ramdisk has been built, subsequent restarts should load the ivtv frame buffer a short bit into the system load process, much earlier than without the initrd changes, and soon enough that you should see the rhgb boot screen.

With the addition of the ivtv X driver, I have no qualms about recommending the use of the PVR-350's output to anybody that 1) doesn't ever want to play games on their system (no OpenGL acceleration), 2) doesn't have an addiction to goom (one of the MythMusic visualizations), or 3) never wants to watch HDTV. The video quality is second to none, for standard-def programming. Of course, it isn't all that easy to set up sometimes...

Oh yeah, and with the ivtv X driver, the measly 1GHz Via C3 in my EPIA (runs like a PIII-600) can decode and play back high-resolution xvid and ffmpeg4 encoded DVD rips without a problem.

[For bttv-based cards \(older WinTV-Go, AVerMedia AVerTV Studio, many others\):](#)

A wide variety of older TV capture cards utilize the Brooktree/Conexant bt8xx chipsets. The particular card I have is an [AVerMedia AVerTV Studio](#), which has a chipset labeled Conexant Fusion 878A (Conexant bought out Brooktree some time ago). Note that there aren't many bttv cards available new anymore, as Conexant has replaced the 8xx chipset with the cx2388x chipset. Many cards branded the same could be either 8xx-based, or could be cx2388x-based. The newer cards use a different driver (cx8800) to function.

Anyhow, the `lspci -v` output for my AVer card looks like so:

```
# /sbin/lspci -v

[snipped to relevant section only]

03:07.0 Multimedia video controller: Brooktree Corporation Bt878 Video Capture (rev 11)
    Subsystem: Avermedia Technologies Inc: Unknown device 0003
    Flags: bus master, medium devsel, latency 64, IRQ 19
    Memory at d0500000 (32-bit, prefetchable) [size=4K]
    Capabilities: [44] Vital Product Data
    Capabilities: [4c] Power Management version 2

03:07.1 Multimedia controller: Brooktree Corporation Bt878 Audio Capture (rev 11)
    Subsystem: Avermedia Technologies Inc: Unknown device 0003
    Flags: bus master, medium devsel, latency 64, IRQ 19
    Memory at d0501000 (32-bit, prefetchable) [size=4K]
    Capabilities: [44] Vital Product Data
    Capabilities: [4c] Power Management version 2
```

The driver for this class of card is automatically loaded at system boot time without having to lift a finger. In my case, everything worked fine in `xawtv` (once it was configured with the right parameters, i.e. `ntsc`, `us-cable`, `tv input`), and I was able to use it in `MythTV` right off the bat.

While my card is correctly auto-detected and configured, some more advanced features may not be enabled and some cards may not be properly detected. A script that appears to be of some use is `ftvco`, or Find TV Card Options, which you can find here: <http://www.mind.lu/~yg/ftvco/>. It'll ask you a few questions, then try a number of combinations, which you can preview in `xawtv`. Chances are, one of those combinations will work for your card, and then a file is written out with the necessary contents to be placed into `/etc/modprobe.conf`.

Unlike the `PVR-x50s`, which mux audio in with the video stream into `mpeg2` files, I needed an audio jumper cable from the sound out on the TV card to the line in on my sound card. This is all well and good for `xawtv`, but you need to make a few quick tweaks for use with `MythTV`. Open up your sound mixer (Fedora/Red Hat menu → `Sound & Video` → `Volume Control`) and mark the check boxes for `Mute` and `Record` for the `Line In` on your sound card. Now when you try `xawtv`, you'll get no sound, but when you fire up `MythTV`, everything will be gravy. Trust me. Note that `ALSA` in the 2.6 kernels has a `snd-bt8x8` modules (or something like that) which works with *some* `bttv` cards to get audio into your system without need for an audio jumper cable, but I haven't tried it.

[North American HDTV/ATSC capture cards:](#)

The latest Fedora kernels support a wide range of HDTV/ATSC capture cards, the full list of which can be found in the DVB wiki, at http://www.linuxtv.org/wiki/index.php/ATSC_devices, complete with whether or not a given card can be used with over the air HDTV broadcasts, cable HDTV signals and analog TV signals. For the most part, once you've installed any necessary firmware files into `/lib/firmware`, these cards should be automagically found and configured by their corresponding drivers.

If you haven't already got a card, the `pCHDTV HD-5500` comes highly recommended, as does the `BBTI/TechniSat AirStar HD-5000`. Neither card requires any firmware to function, and both support cable and over the air HDTV capture. Personally, I prefer to do all my analog standard definition recording via an `ivtv`-driven card, but if you're short on PCI slots (and/or external interfaces) and can spare `cpu` cycles for your standard-def recordings, the `pCHDTV HD-5500` supports analog signals, while the `AirStar HD-5000` does not. Another plus for the `HD-5500` is that `pCHDTV`'s cards are built with Linux support as their primary focus, and they work directly on the drivers for it, versus most other cards that are built primarily with Windows in mind.

The pcHDTV HD-5500 can be had straight from pcHDTV.com, while the AirStar HD-5000 can be found in a variety of places, including [Cyber EStore](#), [Mythic.TV](#) and [Digital Connection](#), all of which carry additional hardware you might want to thumb through as well.

Both of these cards (and most other supported cards) should have corresponding `/dev/dvb/adapterX/` directories created when their drivers are loaded at system boot time. Assuming these exist, your work here should basically be done until you're ready to scan for channels with MythTV.

Note that in the X in `/dev/dvb/adapterX` should be 0 if you have a single card, and increments from there. This number corresponds output in `dmesg`. For example, here's an extract from a system with an AirStar HD-5000 and a pcHDTV HD-3000:

```
flexcop-pci: will use the HW PID filter.
flexcop-pci: card revision 2
DVB: registering new adapter (FlexCop Digital TV device).
b2c2-flexcop: MAC address = 00:d0:d7:0e:a5:02
b2c2-flexcop: found the lgdt3303 at i2c address: 0x59
DVB: registering frontend 0 (LG Electronics LGDT3303 VSB/QAM Frontend)...
b2c2-flexcop: initialization of 'Air2PC/AirStar 2 ATSC 3rd generation (HD5000)' at
the 'PCI' bus controlled by a 'FlexCopIIB' complete
cx2388x_dvb driver version 0.0.6 loaded
cx8802_register_driver() ->registering driver type=dvb access=shared
CORE cx88[0]: subsystem: 7063:3000, board: pcHDTV HD3000 HDTV [card=22]
cx88[0]/2: cx2388x based dvb card
DVB: registering new adapter (cx88[0]).
DVB: registering frontend 1 (Oren OR51132 VSB/QAM Frontend)...
```

Of particular interest are the "DVB: registering frontend X" lines, where X lines up directly with `/dev/dvb/adapterX`. The log output surrounding these lines should give you clues as to which card is which. If it isn't obvious, in this case, card 0 is the AirStar, card 1 is the pcHDTV.

[DVB capture cards \(Nova-T, Fusion DVB-T, other non-North American HDTV\):](#)

I have no first-hand experience with anything but US ATSC DVB cards, but some of the same drivers that push those capture cards also push DVB-T, DVB-S, DVB-C, etc. HDTV capture cards commonly used in Europe, among other places. In fact, their setup should be quite similar -- just put the card in, and the kernel should auto-detect it, load the driver and its ready for action.

Start at the LinuxTV wiki, at http://www.linuxtv.org/wiki/index.php/Main_Page, which includes sections on supported DVB-S, DVB-T and DVB-C devices, as well as all sorts of other DVB-related information and links.

I'd also highly recommend non-US DVB users check out Jani Talikka's pages at <http://www.users.on.net/~jani/dvico-mythtv.html> and Martin Smith's pages at <http://www.ethics-gradient.net/myth/mythdvb.html>.

[FireWire capture from a supported cable box:](#)

MythTV supports capturing video over FireWire with a supported cable box (Motorola DCT-6200/6400 and Scientific Atlanta 3250/4200-series). I've personally recorded HDTV programming off my own DCT-6200 with excellent results. Setup is really quite trivial, and reliability is pretty good. You'll obviously need a FireWire card in the box you want to hook your cable box to, but note that not all FireWire cards are created equal (or at least not all work equally well under Linux). The on-board FireWire on an Asus board I have didn't work worth a damn, but the FireWire port on my SoundBlaster Audigy works great, as does a Western

Digital-branded PCI FireWire card (NEC OrangeLink chipset).

Anyhow, if you're running MythTV builds from ATrpms, you've got almost all the necessary tidbits installed, as we added FireWire support and all the needed libraries to ATrpms some time back. You'll need to manually install the libiec61883-utils package if you want to do some initial testing outside of MythTV:

```
# yum -y install libiec61883-utils
```

Now, connect the cable box to your FireWire input and power it up. You should see some output in /var/log/messages similar to:

```
ieeel394: Host added: ID:BUS[0-00:1023] GUID[0090a94000003380]
ieeel394: Node added: ID:BUS[0-01:1023] GUID[000f9ffffea323c6]
ieeel394: raw1394: /dev/raw1394 device initialized
```

Now run the plugreport utility (part of libiec61883-utils) to see what your system sees where on the FireWire bus:

```
# plugreport
Host Adapter 0
=====

Node 0 GUID 0x0090a94000003380
-----
libiec61883 error: error reading oMPR
libiec61883 error: error reading iMPR

Node 1 GUID 0x000f9ffffea323c6
-----
oMPR n_plugs=1, data_rate=2, bcast_channel=63
oPCR[0] online=1, bcast_connection=0, n_p2p_connections=1
      channel=0, data_rate=2, overhead_id=0, payload=376
iMPR n_plugs=0, data_rate=2
```

Make note of the port and node where plugreport says it sees a device. The port (aka Host Adapter) is almost always 0, the node varies. In the above example, the port is 0, the node is 1. Note that these match up with the information in the log output, they're just easier to decipher in plugreport's output. You'll pass the node number on to the test-mpeg2 utility to do a test capture, as well as entering it in the firewire recording device configuration in mythtv-setup. First, I suggest setting the cable box to a known-working HDTV channel (manually, using the remote or buttons on the box), preferably to one of the major networks, which ought to be free of any encryption. With that done, try a test capture, where the number after the '-r' switch is the node plugreport says it sees your box at:

```
$ test-mpeg2 -r 1 > testcap.ts
libiec61883 warning: Overlaid connection on channel 0.
You may need to manually set the channel on the receiving node.
Starting to receive
(hit ctrl-C)
done.
```

You should be able to play the file back using mplayer:

```
$ mplayer -vo xv testcap.ts
```

At the moment, you'll have to figure out manually what channels do and don't work. I created a custom lineup with Zap2it containing only the channels I found I could get off the FireWire port on my box. You'll also need a way for Myth to change the channels on your cable box. MythTV has internal channel-change support over FireWire for most supported high definition cable boxes, so no external channel-change app is

generally required, which definitely simplifies setup. Once you've figured all that out, you're ready to set up the capture device for Myth to use.

Fire up `mythtvsetup`, and navigate into section 2 to set up a new capture device. Set the card type to "FireWire Input", select your cable box model, set Connection type to "Broadcast", FireWire port to 0 (unless you got something different from `plugreport`), FireWire Node to whatever `plugreport` said, FireWire speed to "100Mbps" (more than sufficient, this option may go away in the future) and leave Default Input as "MPEG2TS" (the only current option). Assign the custom lineup to your FireWire input, and that's about it. Works for me, anyhow. :-)

11. Get and install lirc

There are `lirc` rpms available from ATrpms, installable via `yum` and everything. However, it should be noted that they don't work with every single remote out there. Axel has attempted to roll in as much support for the most popular remotes as possible, and quite a few do now work. It is also possible to obtain the source packages, and potentially build it for your specific remote/receiver, if the prebuilt packages don't work for you.

The earlier "`yum -y install mythtv-suite`" command installed part of LIRC for us already (`lirc-lib`). To finish the install, we need the appropriate kernel module as well. Simply install the remaining bits like so:

```
# yum -y install lirc-kmdl-$KVER
# yum -y install lirc
```

Expand the appropriate section below for your IR receiver interface.

[Using the IR port on a PVR-x50:](#)

There are now three different remotes floating around that come with PVR-x50 cards. There's an old black one, and two different grey and black ones. The older grey and black remote has the four colored buttons around the channel and volume controls, while the newer one has the four colored buttons at the bottom of the remote. A single config file configured to recognize all three remotes is provided in the `lirc` packages. Simply copy that into place, like so:

```
# cp /usr/share/doc/lirc-*/remotes/hauppauge/lircd.conf.hauppauge /etc/lircd.conf
```

Alternatively, you can also use certain universal remotes with you Hauppauge IR receiver. For a while, I was using a RadioShack 15-2116 remote to control two of my MythTV boxes. I've done a little write-up on what was required to make it work, complete with config files, which you can find here:

<http://wilsonet.com/mythtv/remotes.php>

Now we add a few lines to `/etc/modprobe.conf` to set up the `lirc` module. Add these line to the bottom of `modprobe.conf`, which should ensure `ivtv` is loaded before `lirc_i2c`, so we can access the `i2c` bus on the `ivtv` card:

```
alias char-major-61 lirc_i2c
install lirc_i2c /sbin/modprobe ivtv; /sbin/modprobe --ignore-install lirc_i2c
```

Now try loading up the `lirc` module...

```
# /sbin/depmod -a
# /sbin/modprobe lirc_i2c
```

If everything is still going like it should, let's set up lircd (the lirc receiver daemon) to start at boot time:

```
# /sbin/chkconfig lircd on
```

Then go ahead and fire it up right now...

```
# /sbin/service lircd start
```

If you get an [OK], then proceed to testing the remote. If not, check /var/log/messages to see if you can figure out what went wrong. To begin testing, fire up the irw tool:

```
$ /usr/bin/irw
```

You should get some output corresponding to the buttons you pressed. Hit ctrl-C to stop irw.

The next step is to set up a file to map those remote button presses onto key combinations that MythTV understands, so your remote actually does something useful once we have MythTV working. I've got one already prepared and ready for download for the older grey and black remote. You can view it here:

<http://wilsonet.com/mythtv/lircrc-haupgrey.txt>

Getting it onto your MythTV box is a simple matter of:

```
$ wget http://wilsonet.com/mythtv/lircrc-haupgrey.txt
$ mkdir ~/.mythtv
$ mv lircrc-haupgrey.txt ~/.mythtv/lircrc
```

Please be aware that this lircrc file is somewhat customized toward my own needs (though I actually use a universal remote now). For example, the VOL+/- buttons don't alter volume at all, because I use an external amp. However, the MUTE button does mute. Other notables are the YELLOW button, which seeks to the previous commercial cut point, the BLUE button, which seeks to the next commercial cut point, and the FULL button, which toggles the aspect ratio (very useful with my HDTV). Adjusting the buttons to suit your own needs is a simple matter of editing the lircrc file.

More likely than not, you'll also want to use your remote with whatever video player gets launched to view your video library via MythVideo. The above lircrc file includes settings in it for both xine and mplayer, but they both look for an lirc settings file in ~/.lircrc, which we can remedy simply by symlinking to ~/.mythtv/lircrc, like so:

```
$ ln -s ~/.mythtv/lircrc ~/.lircrc
```

Courtesy of Brad Templeton, here's an lircrc file for the newer grey and black Hauppauge remote:

<http://wilsonet.com/mythtv/lircrc-haupgrey-g3.txt>

Follow the same steps as detailed for the other grey remote to get everything playing nice.

[Using the IR port on your bttv card:](#)

There are quite a few different possible remote control interfaces used across the wide array of bttv cards. Many of them SHOULD work with the latest lirc kernel module available from ATrpms, but maybe not all. You'll have to do some investigation of your own to figure out which lirc interface type your card uses, but the types that should work are (from the lirc driver types in the rpm spec file):

```
lpt1, sir_com3, it87, flyvideo, avermedia, tekram_bt829, udp, avermedia98, com1,
hauppauge
```

My AVerTV Studio card uses the avermedia98 lirc driver, and works quite nicely with this rpm. The actual kernel module used is lirc_gpio, so I add this line to my /etc/modprobe.conf file:

```
alias char-major-61 lirc_gpio
```

Next, I dropped the appropriate lircd.conf file for my remote in place. Config files can be found in /usr/share/doc/lirc-*/remotes/<your type>. I simply did this:

```
# cp /usr/share/doc/lirc-*/remotes/avermedia/lircd.conf.avermedia98 /etc/lircd.conf
```

[Using a serial IR receiver:](#)

Most serial IR receivers should work with the lirc_serial module, but will need some extra parameters, such as port, irq, and possibly receiver type passed to them. All of them will require the following in /etc/modprobe.conf:

```
alias char-major-61 lirc_serial
```

If your receiver is hooked to the first serial port, aka COM1, you'll need these extra settings in /etc/modprobe.conf:

```
options lirc_serial irq=4 io=0x3f8
install lirc_serial /bin/setserial /dev/ttyS0 uart none ;\
/sbin/modprobe --ignore-install lirc_serial
```

If your receiver is hooked to the second serial port, aka COM2, you'll need these settings:

```
options lirc_serial irq=3 io=0x2f8
install lirc_serial /bin/setserial /dev/ttyS1 uart none ;\
/sbin/modprobe --ignore-install lirc_serial
```

The one other thing to look out for are serial receivers that are "Igor-style", which use a slightly different chipset, requiring an extra item "type=4" tacked onto the end of the appropriate "options ..." line above.

Note that most lirc drivers don't get automatically loaded, but Fedora has a neat little facility to force the loading of modules early in the boot process, via a file in /etc/sysconfig/modules/. I've got a script you can download which should load up any lirc modules you've configured in /etc/modprobe.conf. Set it up like so:

```
# cd /etc/sysconfig/modules/
# wget http://wilsonet.com/mythtv/lirc.modules
# chmod +x lirc.modules
```

At this point, you should be able to start up lircd:

```
# /sbin/service lircd start
```

The start command should return an OK or two (look in /var/log/messages if you don't get an okay for some possible insight as to what went wrong). Now fire up irw to verify that lircd is receiving signals from your remote:

```
$ /usr/bin/irw
```

You should see some output when you press buttons on your remote that correspond with each button press. Hit ctrl-C to stop irw.

At this time, you're on your own to create a lircrc file for your remote. I have mostly implemented one for my AVerTV Studio remote, mostly just to verify that it work and until I get something better (quite frankly, it is utter junk, especially for use with MythTV).

Hopefully, this section will be of some aid to those using a different remote interface also...

Other routes to consider for an IR receiver if you either don't like the one that comes with your capture card or have a remote frontend without a capture card are serial port and USB receivers. I'm a huge fan of the USB receiver from Windows XP Media Center Ed., because they have a nice long cord to get them optimally placed (versus the very short cord on the PVR-x50's receiver) and have much better range and multi-directionality than any other one I've used. I've also got a serial one on the way shortly.

You can get a USB XP MCE receiver from NewEgg.com, but its a touch spendy, since you don't have the option to buy it without the MCE remote:

<http://www.newegg.com/app/ViewProductDesc.asp?description=80-100-202&depa=0>

It should be noted that there are two different lirc USB MCE remote drivers, one for an early model (which I have and love), which uses the lirc_mceusb driver, and the newer-model ones (which is all NewEgg carries now), which use the lirc_mceusb2 driver. Both are bundled in the ATrpms lirc packages.

Another popular USB combo is the ATI Remote Wonder, which isn't actually IR, but RF. RF doesn't have nearly as many directionality or range issues as IR, but I don't believe you can control anything but your Myth box with a Remote Wonder, and the receiver is only good for the Remote Wonder. These ship with some ATI TV tuner cards, and you can also find them as a stand-alone at NewEgg.com:

<http://www.newegg.com/app/ViewProductDesc.asp?description=80-100-401&depa=0>

Those using an ATI Remote Wonder probably want to check out Tim Litwiller's doc on how he set up his:

<http://www.litwiller.net/mythtv/remote.html>

You can easily build your own serial IR receiver, per the diagrams on the LIRC web site, or you can just pick one up online for a few bucks. One popular source for pre-build serial IR receivers is Zapway.de, which while in Germany, does ship globally.

12. Set up MySQL

We'll need to enable MySQL to load at startup, set some passwords, and create the MythTV database, which we'll populate shortly. The population of this database is handled by mythtvsetup in the next step, and all MythTV add-on module database additions must be done AFTER running mythtvsetup at least one time.

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

Set the mysql root password, replacing ROOT_PWD with your chosen administrator password:

```
# mysql -u root mysql
mysql> UPDATE user SET Password=PASSWORD('ROOT_PWD') WHERE user='root';
mysql> FLUSH PRIVILEGES;
mysql> quit
```

Now we create the mythtv database (called mythconverg) to get us started:

```
$ mysql -u root -p < /usr/share/doc/mythtv-0.20/database/mc.sql
(enter the password you just set above when prompted)
```

Again, all subsequent database population for MythTV's add-on modules must now be done AFTER running mythtvsetup at least one time. Anyhow, It's worth customizing some parameters in /etc/my.cnf for optimal performance with Myth. I know next to nothing about the topic, but from experimentation and [listening to people who do know what they're doing \(I think\)](#), I found these adjustments to my.cnf under the [mysqld] section improve performance with both MythTV (esp. in the GUI) and MythWeb:

```
key_buffer = 16M
table_cache = 128
sort_buffer_size = 2M
myisam_sort_buffer_size = 8M
query_cache_size = 16M
```

13. Set up MythTV

There are a number of things you might want to figure out in advance to successfully complete your setup. First, recall what your video device was set up as (likely /dev/video0 if you have a single v4l card, DVB #0 if you have one DVB card). Note that the actual device number is NOT what determines what order MythTV will use your cards in. What matters is the order that you enter them in mythtvsetup, so your best card doesn't have to be /dev/video0. Set up /dev/video8 first, and it'll be the first card used for recordings.

You will also need your zip code (please tell me you know this...), and it would help to know the right listing from <http://www.zap2it.com/> (for US users, other countries use different listings sources) that matches your service provider, so xmltv gets the right program listings (I screwed up the first time, because I was guessing, and most of the programming info was askew). US folks will also have to fill out some paperwork to use zap2it's new xml data feed listings (the only way to go now). See this page in the MythTV docs for details:

<http://mythtv.org/docs/mythtv-HOWTO-5.html#ss5.3>

Now on to launching the MythTV setup utility. For this part, you need to have an X session started up, as the setup utility is an X application. Fire it up like so:

```
$ mythtv-setup
```

I'm told that non-US folks may have issues correctly getting through the `tv_grab_xx/xmltv` part of the setup if "focus follows mouse" is already set (in KDE's Control Center), so keep that in mind. Just set "focus follows mouse" when everything else is already configured.

Recent changes in the `mythtv-setup` and database population methodology broke the default path settings for the MythTV rpms, which should be `/var/lib/mythtv/` for storage of recorded shows, and `/var/cache/mythtv/` for live TV buffer. These directories are automatically created at install time, but you'll have to manually enter them in section 1 of `mythtvsetup`.

Those using a dedicated `/storage` partition, per my example, are recommended to create a `/storage/recordings` directory, and set it as their video store. You can do pretty much whatever you like here, such as recording to an NFS or samba mount, a software RAID array, or even to an [LVM group](#) so you can expand your `/storage` partition sometime down the road. Just make sure your `mythtv` user has permission to read and write to whatever location you choose.

Go through the setup steps in order. Follow the on-screen instruction, with aid from the MythTV website's documentation on this page:

<http://www.mythtv.org/docs/mythtv-HOWTO-9.html>

NOTE: your system may appear to hang at step 3; give it time, it isn't locked up, that part just takes a while!

Once you've gone through the setup, you have to populate the MythTV database with some program info. I spent a good long time tweaking my channel lineup on zap2it's site to remove all the junk channels I didn't really care to have show up. Once you have your listings to your liking, you're ready to fill your database with programming info, so lets get some guide data:

```
$ mythfilldatabase
```

If you're using a guide data source other than zap2it (i.e., anyone outside the US and Canada), I'm told that you may well need to add a `--manual` flag to the end of that command to get it to work. Look at the output of `"mythfilldatabase --help"` for more clues if you have problems. Also, BE PATIENT! This step can take a fairly long time, depending on your internet connection speed and how many channels your service provides...

Of course, you'll need to re-run `mythfilldatabase` regularly, to keep your guide data up to date. Fortunately, MythTV has built-in support for scheduling when `mythfilldatabase` should be run. You can find it in the MythTV frontend under Utilities/Setup -> Setup -> General, on the last page of that section. Go ahead and fire up the backend and the frontend (I recommend starting them in separate shell windows, so you can differentiate backend and frontend output). In one shell window:

```
$ mythbackend
```

And in the other:

```
$ mythfrontend
```

I like my guide data updated daily, so I set the Run Frequency to 1. I recommend either leaving the execution window at the default size (Start at 2, End at 5), or increasing it slightly, as well as leaving the "Run `mythfilldatabase` at time suggested by the grabber" option enabled. If your grabber supports this option, it will help to distribute the load on your guide data provider's servers (zap2it does support it). Really, any good-sized window in which you generally don't have any recordings scheduled will do (note that the hours are in military time, i.e., 13 == 1:00 pm). Personally, I leave my window set to Start at 10:00 am and end at 2:00 pm, since I'm usually not recording anything in that window, and I should still get the latest and greatest guide data in time for each evenings'

recordings.

It is also possible to run mythfilldatabase every night from cron, but this is no longer the preferred method. But if you insist, here's a reasonable crontab entry:

```
$ crontab -e
#----Insert the following text into your mythtv user's crontab----#
### Run mythfilldatabase every night at some random time after 10:01am
01 10 * * * sleep $(expr $RANDOM \% 14400) && mythfilldatabase >
/var/log/mythtv/mythfilldatabase.log 2>&1
#----Cut above here (don't include this line)----#
```

There are various other options within mythfrontend that you can tweak... In the past, I've recommended enabling "De-interlace Playback", even if you're using an interlaced display (like a TV). However, I've found with the latest nvidia driver and its flicker filter, this is no longer necessary for TV output (note that the flicker filter can only be used on SVideo or composite TV-Out, it isn't an option for DVI or VGA). You're on your own for the rest of the settings. Just be sure to go through every section, and get to the point where you click a "Finish" button, otherwise some values seem to not get initialized (most notably, you may get crashes that say something about not being able to open the dsp).

Another setting I'd recommend enabling, especially if you're using an nVidia video card (not sure about others) is OpenGL VSync, which recently became a toggleable option. It isn't stable for everybody, but works very well for me on several systems (and I find it to be a must for optimal HDTV playback). If everything appears to be working, once you've gone through all the configuration steps, you can set your system up to automatically load everything at startup. And congrats, you've done the bulk of the work. You can stop here if you want, but if you're like me, you'll want everything to load automatically when the system boots.

PVR-350 output users need to enable use of the 350's decoder. Go into Settings → TV Settings → Playback and go to the "Use PVR-350 Output" screen. Check the box, and make sure it reads /dev/video16. Click through to the "Finish" button of that section, and now you should be good to go for MythTV to take full advantage of the PVR-350's decoder and TV Output.

14. Configure automatic startup

The necessary init script for the MythTV backend to automatically start at system boot is already in place for you, just simply turn it on:

```
# /sbin/chkconfig mythbackend on
```

If the backend isn't already running, save yourself a reboot and issue this command:

```
# /sbin/service mythbackend start
```

Now, because I have a few things that don't seem to want to play nice anymore (i.e., nvidia-settings don't load like they should, alsas volume levels aren't restored), I decided to create a quick little shell script in `~/kde/Autostart/myth-load.sh` to handle loading up all the extra goodies I need/want to auto-start, as well as force stubborn things to work. This script loads my nvidia settings, restores alsas volumes, launches irexec for my little power button script (on the [Tips 'n' Tricks](#) page), then launches mythfrontend, all in one fell swoop. Just copy and past all this into `~/kde/Autostart/myth-load.sh` (adjust accordingly for different desktop environments):

```
#!/bin/bash
# Only do this stuff if we're on the main display
```

```
# (i.e., don't do this in a vnc session)
if [ `echo $DISPLAY | grep -c ":0" -ge 1 ` ]
then
  # Load nVidia driver custom settings
  nvidia-settings --load-config-only &
  # Restore audio settings
  /usr/sbin/alsactl restore
  # Launch irexec for myth power button stop/start
  irexec &
  # Launch myth frontend
  mythfrontend &
  # Disable dynamic power management (screen blanking)
  /usr/bin/xset -dpms
  # Disable screen saver
  /usr/bin/xset s off
fi
exit
```

Don't forget to make it executable:

```
$ chmod +x ~/.kde/Autostart/myth-load.sh
```

While logged in to an X session, you'll also have to run `gdmsetup`, to set autologin for your mythtv user. You'll have to get a root terminal open (if you aren't logged into your X session as root already, which is a no-no), then:

```
# gdmsetup
```

In GDM Setup, on the first tab ('General'), you should see a section "Automatic login". Check the box for "Login a user automatically on first bootup" and select your mythtv user from the popup menu. Alternatively, if you are not very comfortable with Linux just yet, and you suspect there may be occasions where you've mucked something up to the point that an auto-login will lock up the computer, you might not want to use Automatic Login. Instead, you might opt to use the "Timed Login" option, to log the mythtv user in a few seconds after the login screen first appears. This way, you can circumvent the mythtv user logging in, and log in as root to hopefully correct whatever you've broken.

NOTE:

if you installed the packages from `kde-redhat`, they probably supplanted `gdm` as your display manager and put `kdm` in its place. I like the look of Red Hat's `gdm` much better than `kdm`, so I changed it back by editing `/etc/sysconfig/desktop` and setting "DISPLAYMANAGER=GNOME". Not that you should be seeing the login screen often, my box is always sitting there logged in with the Myth UI up. Maybe I'll add some `kdm` info in a bit...

Anyhow, I recommend setting the Kicker (the panel at the bottom of the screen) to auto-hide, so it doesn't end up on-screen during video playback, disabling the screen saver, and disabling any monitor-related dynamic power management, or you may not be able to wake your system up if you don't have a mouse and keyboard connected. You can use the KDE Control Center to disable your screen saver, and edit your `xorg.conf` file to disable `dpms` (or at least the screen blanking parts). Note that the above `myth-load.sh` script actually issues a pair of commands at login time that should also disable `dpms` and the screen saver, but you can turn it off everywhere you can to be sure...

And finally, to turn on auto-hiding of the Kicker, right-click on the Kicker, choose 'Configure Panel...', then select the 'Hiding' tab, and select the radio button for 'Hide automatically'.

15. Configuring MythTV add-ons

When you installed MythTV using "yum -y install mythtv-suite", you not only installed MythTV, but also all of the add-on modules currently available. You've likely still got a bit of work to do to make some of them functional. The MythTV wiki and official docs cover the plugins much better than I ever did, so the following is mostly just a

list of references.

MythArchive-

/usr/share/doc/mytharchive-0.20/
<http://mythtv.org/wiki/index.php/Mytharchive>

MythBrowser-

/usr/share/doc/mythbrowser-0.20/
<http://mythtv.org/wiki/index.php/MythBrowser>

MythDVD-

/usr/share/doc/mythdvd-0.20/
<http://mythtv.org/wiki/index.php/MythDVD>
<http://mythtv.org/docs/mythtv-HOWTO-20.html>

MythGallery-

/usr/share/doc/mythgallery-0.20/
<http://mythtv.org/wiki/index.php/MythGallery>
<http://mythtv.org/docs/mythtv-HOWTO-15.html>

MythGame-

/usr/share/doc/mythgame-0.20/
<http://mythtv.org/wiki/index.php/MythGame>
<http://mythtv.org/docs/mythtv-HOWTO-16.html>

Most of the emulators can be installed using yum, like so:

```
# yum -y install xmame fceultra snes9x
```

MythMusic-

/usr/share/doc/mythmusic--0.20/
<http://mythtv.org/wiki/index.php/MythMusic>
<http://www.mythtv.org/docs/mythtv-HOWTO-17.html>

MythNews-

There isn't much to say about MythNews, its just a simple RSS feed reader. Go into MythTV → Utilities/Setup → Info Center Settings → News Settings and select the news feeds you want, and you're done with the setup.

MythVideo-

/usr/share/doc/mythvideo--0.20/
<http://mythtv.org/wiki/index.php/MythVideo>

Within the Setup section of MythTV, configure your movie storage directory (MythTV → Utilities/Setup → Media Settings → Video Settings → General Settings). You might want to look at the [Tips 'n' Tricks](#) page for some additional info on configuring mplayer and/or xine, especially if you have a wide-screen TV. Edit/insert metadata about your video collection in Utilities/Setup → Video Manager.

MythWeather-

/usr/share/doc/mythweather--0.20/
<http://mythtv.org/wiki/index.php/MythWeather>
<http://www.mythtv.org/docs/mythtv-HOWTO-18.html>

MythWeb-

The mythtv-suite put almost everything where it needed to be. You need to read conf.php (/var/www/html/mythweb/config/conf.php) and configure things in that file accordingly. Once that's done, all you have to do is enable apache:

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

Now just point a web browser to <http://your.mythtv.box.ip.or.dns.name/mythweb/> and go to town. Please note: MythWeb provides listings, lets you schedule recordings, and see what is already recorded. You cannot use this interface to control playback on your MythTV box in any way. There are also some controls for MythMusic and MythVideo.

Personally, I don't plan on using the web server on this box for anything *but* MythWeb, so I opted to move everything in /var/www/html/mythweb/ to /var/www/html/ and remove the mythweb folder, so I get to MythWeb with just <http://htpc/> (htpc is the hostname for MythTV box). Alternatively, Zachary Hamm suggests creating an index.php file in /var/www/html/, containing the following to achieve the same effect:

```
<?php header("Location: /mythweb"); ?>
```

This works better for those who also run other web-based apps on their MythTV box (like phpMyAdmin, for example), and keeps the apache doc root a bit tidier.

If you wind up with a blank page when you try to use mythweb, it may be that you've left SELinux enabled. You can either disable it by setting SELINUX=disabled in /etc/sysconfig/selinux and rebooting, or alter the security context solely for httpd, like so:

```
/usr/sbin/setsebool -P httpd_can_network_connect=1
```

Additional MythWeb documentation:

/usr/share/doc/mythweb--0.20/
<http://mythtv.org/wiki/index.php/MythWeb>
<http://www.mythtv.org/docs/mythtv-HOWTO-14.html>

16. Upgrading your system

Upgrading your MythTV installation to a new release is fairly straight-forward. As of MythTV 0.13, all database changes are handled automagically, so you merely have to upgrade your mythtv rpms when they become available. Occasionally, driver updates are also required (such was the case with the ivtv driver when upgrading to MythTV 0.13, and PVR-350 output in 0.14+ required ivtv 0.1.9+). The basic upgrade process should be as simple as:

```
# yum -y upgrade mythtv-suite
```

Note that this will NOT suffice to upgrade all sub-packages in the event of an updated build from ATrpms of the same major version. In other words, say 0.18.1-115 packages are released with some bugfix not in 0.18.1-114 (-114 and -115 are ATrpms build numbers), but mythtv-suite is already satisfied w/0.18.1-114 (mythtv-suite only needs 0.18.1 packages of any build number), so you'll have to specify all components manually (rpm -qa | grep myth for a list of 'em), or you can try the following:

```
# yum -y upgrade \*myth\*
```

If you experience any sort of problem following an update of this nature, try running a full yum upgrade of all packages. I believe Axel hates the above suggestion, but its always worked well for me... :)

If a driver update is required, you may need to:

```
# yum -y install <somedriver>-kmdl-$KVER <somedriver>
(<somedriver> could be ivtv, lirc, nvidia-graphics, etc.)
```

A general note about upgrading and installing rpm packages: you should NEVER use --nodeps to install or remove a package. If you can't get around using --nodeps, there is likely a packaging problem, which you should report upstream, to have it fixed. A tanked rpm database can do Very Bad Things™ to your system, and recovery is sometimes impossible. Try 'yum check' to verify its integrity, should you happen to have committed the --nodeps sin in the past.

Upgrading your kernel is a little bit more involved than upgrading only MythTV, because all custom kernel modules, both rpm-installed and source-installed, will have to be updated also. Kernel modules carry two different versions, one for the kernel they are for (the 'kernel version') and one for the actual module itself (the 'module version'). Everything but kernels and their matching kernel modules get upgraded automatically by 'yum upgrade'. However, note that cases where the kernel version on a kernel module remains the same, but the module version is incremented, an 'yum upgrade' will upgrade that kernel module. Essentially, you really only need to take extra measures when upgrading your kernel, as everything else gets handled automatically.

When you do come to a point where you need to upgrade your kernel, one of the nice features of installing from packages, rather than source, shows itself. For all the ATrpms kernel modules, you can simply type in the command...

```
# rpm -qa {alsa,ivtv,nvidia-graphics,lirc,video4linux}\* |grep kmdl
```

...to get a full list of all the kernel modules you have installed on your system. With that list in hand, you can now install your new kernel, along with all the corresponding updated kernel modules. For example:

```
# yum -y install kernel-2.6.20-1.2944.fc6
# yum -y install {alsa,ivtv,lirc,nvidia-graphics9755,video4linux}-kmdl-2.6.20-1.2944.fc6
```

You can also uninstall an older kernel and its kernel modules with a single command:

```
# yum -y remove kernel-<oldversion>
```

17. Miscellanea

This section has been moved to its own page, titled [Tips 'n' Tricks](#).

18. Trouble-shooting

First and foremost, READ THE OFFICIAL DOCUMENTATION!!! Specifically, the page on [Troubleshooting, at http://mythtv.org/docs/mythtv-HOWTO-22.html](#). Second, search the MythTV users *and* developers mailing list to see if this isn't already a known problem that might be fixed in CVS or have a work-around. Again, you can search the archives online at [Gossamer Threads](#). Third, check out the wiki over at <http://www.mythtv.info/>, which also contains quite a bit of valuable information. Fourth, take a look at [ATrpms Bugzilla](#) and the [Fedora Myth\(TV\)ology Wiki and Bug-tracker](#).

When trouble-shooting, I suggest exiting the frontend, stopping the backend and opening a pair of terminal windows. From one, start up the backend (as root, just type "mythbackend"). In the other, launch the frontend with verbose output (just type "mythfrontend -v all"). There should be a fair amount of output spit to the terminal windows that should give you (and the developers) a better idea of what is going wrong with your system. Please include that output when asking for help. Additionally, if you're getting a segmentation fault, your best bet for getting help determining the cause of the problem is to download the myth source code and compile it yourself in debug mode, then run it with gdb, the GNU debugger (how to do that is detailed at the above link). Only with a backtrace can the developers really help you if your setup is causing Myth to segfault.

I plan to further populate this section over time, when I can...

19. CHANGELOG

The changelog maintained here was getting too long, and really didn't serve much purpose anymore. It'll continue to exist in my source repo for historical purposes, but from here out, people should go to <https://svn.wilsonet.com/projects/mythtvology/timeline> to see what has changed.