Raspberry Pi Media Center

Transform your Raspberry Pi into a full-blown media center within 24 hours

Sam Nazarko

Raspberry Pi Media Center

Transform your Raspberry Pi into a full-blown media center within 24 hours

Sam Nazarko

Raspberry Pi Media Center

Credits

Author
Sam Nazarko

Reviewer
Eric PTAK
Sam Wright

Acquisition Editor
Erol Staveley

Commissioning Editor
Maria D'souza

Technical Editor
Sharvari Baet

Copy Editors
Alfida Paiva
Insiya Morbiwala
Aditya Nair
Laxmi Subramanian
Brandt D'Mello
Ruta Waghmare

Project Coordinator
Amigya Khurana

Proofreader
Maria Gould

Indexer
Monica Ajmera Mehta

Graphics
Aditi Gajjar

Production Coordinator
Pooja Chiplunkar

Cover Work
Pooja Chiplunkar

About the Author

Sam Nazarko is an 18-year-old student, currently studying Computer Science at King's College London, UK. He has been using computers since the age of four and has developed a huge passion for them. He has a wide range of experience and knowledge about both Windows and Linux platforms at desktop and server level. He enjoys developing tailor-made solutions for clients, which range from C# application development on Windows to C development on Linux, and has expertise in system deployment and administration on a large scale. He has recently been involved in open source development, harnessing his skills in embedded systems development by delivering Linux distributions for Apple TV and Raspberry Pi that deliver XBMC.

Sam is planning to launch a technology startup in 2013, which will focus on embedded system development and custom-made solutions.

> I'd like to thank the friendly and helpful chaps at Packt Publishing, in particular: Erol Staveley, Maria D'souza, and Amigya Khurana.

About the Reviewers

Eric PTAK is a software R&D engineer, specialized in network protocols all over the OSI stack. His career began with telecom services in NGN architectures. He is also working on mobility and the Internet of Things and he is interested in many technological fields. He started using XBMC with the first-generation XBox and he knows media center constrains and requirements well. For his first Raspberry Pi project, Eric used shairport open source software to build an AirPlay music player, that he named AirPi. The whole Raspberry Pi community now uses this name for their AirPlayer project.

www.PacktPub.com

Support files, eBooks, discount offers and more

You might want to visit `www.PacktPub.com` for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at `www.PacktPub.com` and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at `service@packtpub.com` for more details.

At `www.PacktPub.com`, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.

`http://PacktLib.PacktPub.com`

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

Why Subscribe?
- Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

Free Access for Packt account holders

If you have an account with Packt at `www.PacktPub.com`, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

Dedicated to EN, MN

Table of Contents

Preface

Low-cost and high-performing, with a massively diverse range of uses and applications, the Raspberry Pi is set to revolutionize the way we think about computing and programming. Using it as a media center allows everyone to have a low-cost device that is always on and attached to their TV.

Imagine controlling your TV with your smartphone, dynamically streaming content that you have on various devices in your home, as and when you want. Raspberry Pi Media Center will show you how to configure the complete experience using readily available and accessible tools, so that you too will soon be enjoying your media from the comfort of your own sofa.

With this practical guide, you'll discover how you can transform your Raspberry Pi into a powerful media center. Using Raspbmc (based on the XBMC distribution), you'll be shown how you can deploy and configure your own media center, including the coverage of all the essentials, from controlling your device with a TV remote or smartphone to the creation and customization of your very own media library (including the conversion of any existing collection). You'll even be able to stream multimedia from your devices or directly from the Internet!

What this book covers

Chapter 1, *Getting Started*, explores why the Raspberry Pi is a good hardware choice for a media center, lists the equipment we will need to install media center, and gives a brief introduction to Raspbmc and how to install it.

Chapter 2, *Using Raspbmc*, explains how to navigate and control Raspbmc as well as configure important settings and install add-ons.

Chapter 3, *Managing Content*, explains how to add existing content for playback, the content sources and types Raspbmc supports, and how to add these sources and manage them.

Chapter 4, *Streaming From a Computer*, explains what's needed to stream media from your computer, the advantages and disadvantages of streaming, and how to set up streaming from Windows, Linux, and Mac OS X computers.

Chapter 5, *Setting up the Library Mode*, explains how we can store information, such as episode descriptions of our content, in a database, sharing this across multiple machines, and getting Raspbmc to recognize this database.

Chapter 6, *Using PVR with Raspbmc*, explains what we can do with PVR, gives us a look at the different types of PVR setups, and explains how to configure Raspbmc to use PVR.

Chapter 7, *Converting Media*, explains why converting media is necessary, gives us a look at codecs, and explains how to convert media with freely available software so it can be played in Raspbmc.

Appendix A, *Useful Raspbmc Commands*, provides useful commands that can be executed via the command line with an attached keyboard or via an SSH connection.

Appendix B, *Useful MySQL Commands*, provides a short summary of MySQL commands that can be used at the command line on the system that is hosting the MySQL database.

Appendix C, *Supported Peripherals*, provides a summary of supported peripherals in Raspbmc.

What you need for this book

To run the examples in this book, the following software will be required:

- Remote Database – MySQL Server
- Video encoding – Handbrake

A list of software is given in the following table:

Software Name	URL
Raspbmc	http://www.raspbmc.com
MySQL server	http://mysql.com
Handbrake	http://handbrake.fr
XBMC	http://xbmc.org

Who this book is for

This book does not require any prior knowledge of the Raspberry Pi, but it does assume you are computer literate and comfortable with Mac OS X, Linux, or Windows, and concepts such as installing software.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "For UNIX systems, the following command would suffice, assuming that the filename of the compressed image was `installer.img.gz`."

A block of code is set as follows:

```
export set MYSQL_HOME=/usr/local/mysql-5.5.18-osx10.6-x86_64
export set PATH=$PATH:$MYSQL_HOME/bin
```

Any command-line input or output is written as follows:

```
sam@ubuntu:~$ ssh pi@192.168.1.6
The authenticity of host '192.168.1.6 (192.168.1.6)' can't be
established.
ECDSA key fingerprint is 55:36:12:f1:df:de:06:1a:7a:a4:36:b4:e1:09:a0:e2.
Are you sure you want to continue connecting (yes/no)?
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "Check the option entitled "**Share this folder**" and give the share a meaningful name."
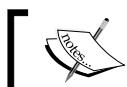
> Warnings or important notes appear in a box like this.

> Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to `feedback@packtpub.com`, and mention the book title through the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on `www.packtpub.com/authors`.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting `http://www.packtpub.com/support`, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website, or added to any list of existing errata, under the Errata section of that title.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at `copyright@packtpub.com` with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at `questions@packtpub.com` if you are having a problem with any aspect of the book, and we will do our best to address it.

1
Getting Started

First off, I'd like to welcome you to this title. By the end of this book, you will be able to set up a comprehensive media center that will serve you well for years to come. Each chapter will guide you through the basics so that by the end of the book we will have gone from a basic system to a comprehensive and complex one that meets your needs.

In this chapter, we will cover the following:

- Why Raspberry Pi is a good hardware choice to be used as a media center
- The equipment needed to build a Raspberry Pi based media center
- What Raspbmc is, and what it can do
- Installing Raspbmc onto an SD card using Windows, Mac OS X, or Linux
- Booting up Raspberry Pi to complete the installation

Why use a Raspberry Pi?

The Raspberry Pi is a small, credit card sized computer launched in 2012 for the purpose of rekindling a dwindling interest in Computer Science. The device therefore had to be affordable, have a low power consumption, and a small form factor. These characteristics are also what make the Raspberry Pi such an appealing solution as a media center. The Raspberry Pi is silent, and a quiet system is essential for an ideal media center experience. The Raspberry Pi may seem like a low performance device for education, however it has HDMI output. Its Ethernet capabilities and powerful GPU capable of decoding full HD videos has attracted much attention. Furthermore, as the operating system runs on an external SD card, a type of memory card, the device itself cannot be bricked, and can always be restored to a working state by reinstalling Raspbmc.

Equipment needed

You will need some equipment to use your Raspberry Pi as a media center; you may or may not have some of these components already. These components can be obtained from most retailers and are relatively inexpensive:

- **Power supply**: The Raspberry Pi requires a micro USB power supply capable of providing 700 mA at 5 volts. As a media center, this supply needs to be reliable, so it is worth picking up a charger capable of at least 1 mA. You can even pickup a charger that is 2 A, but you should ensure that the charger does not exceed 5 volts. It should be noted that not every power adapter will work on Raspberry Pi, mainly because many chargers' output specifications are printed by the manufacturer, rather than regulated independently. A list of good power supplies can be viewed at `http://elinux.org/RPi_VerifiedPeripherals#Power_adapters`.

- **SD card**: This is the card that the media center operating system will run from. Not much space is needed, even in library mode, so an 8 GB card would suffice. It is strongly recommended that you use a Class 6 or Class 10 SD card. This is usually denoted on the card with a number that is encompassed in a circle. Sandisk Ultra cards are reliable, and recommended. The following screenshot shows a Sandisk Ultra card:

- **Card reader**: You may already have this on your computer, but if you do not, you'll need a card reader to install Raspbmc onto your SD card. These can be obtained from most electrical retailers as they are commonly used for digital cameras.

- **Micro USB power cable**: It is used for connecting the power supply to Raspberry Pi. If the supply you are using is a phone charger, you might already have one of these.

- **Network cable**: An Ethernet can be used to connect and install the software on your device. You may also use this later for streaming media. You can also use a supported Wi-Fi adapter with Raspbmc. Refer to *Appendix C*, *Supported Peripherals*, for a list of supported adapters. However, it should be noted that installation is possible without a network connection.

- **HDMI cable**: If you wish to use CEC support (discussed later in this book), which allows you to use your TV or AV receiver's remote to control Raspberry Pi, you will need an HDMI 1.3 or later compliant cable.

- **Composite**: If you are connecting to a display that only allows composite, you will need this cable. You will also need a 3.5 mm stereo cable to get audio playback on your TV as well.

- **Display**: You will need a TV or monitor to use Raspbmc on.

- **Remote**: This is entirely optional, as controlling the Raspberry Pi via other means is possible. For a full list of supported remotes, please visit `http://www.raspbmc.com/wiki/user/configuring-remotes`.

What is Raspbmc?

Raspbmc is a highly optimized and dedicated media center for Raspberry Pi. It brings XBMC to Raspberry Pi in a fast and self updating package in an easy-to-use manner. This is because Raspbmc has been built specifically for the purpose of running XBMC. Raspbmc is flexible and allows expansion via the Debian packaging system. For example, it is possible to utilize **GPIO** (**general purpose input output**) to configure a custom infrared receiver.

XBMC is an open source media center that runs on all major platforms. It began as a project for playing back media on the original Microsoft game console, and it now runs on many platforms. XBMC is not limited to just media playback, and sports a lot of functionality, such as plugins, add-ons, and media scraping. XBMC's existence on the Raspberry Pi can be greatly attributed to the work of Edgar Huceke and Scott Davilla. You can learn more about XBMC at `www.xbmc.org`.

What can Raspbmc do?

Raspbmc is a full-fledged media center and offers the following capabilities:

- Hardware playback of HD (up to 1080p) content in formats H264, MPEG2, and VC1, and MPEG2 and VC1 codecs must be purchased separately

- Playback of multiple music formats (MP3, AAC, FLAC, and more)

- Streaming content from a computer over a variety of streaming protocols

- Support for different looks via skins
- Streaming content from attached hard drives via USB
- AirPlay support (with compatible iOS devices)
- Watch and record TV on your device
- Raspbmc can be controlled via a USB remote, a CEC-compatible device, or via a smartphone
- Using various plugins, you can use online video and music services such as YouTube, Spotify, and iPlayer
- Ability to create a media library that automatically downloads TV shows, film information, and FanArt
- Ability to share this media library across multiple devices, including Windows, Linux, and Mac OSX systems
- Automatic updating means that Raspbmc is always running at optimum performance and has the latest version of XBMC

> Note that this is not a comprehensive list of all the features, and that more are added regularly with Raspbmc's auto-update system.

Installing Raspbmc

Raspbmc can be easily installed from Windows, Linux, or Mac OS X. With the help of a computer, you can install the Raspbmc installer on to the card. You will need an Internet connection on your computer for this. This installer will then configure the Raspberry Pi by grabbing the latest version of Raspbmc. The part where the user installs Raspbmc onto the SD card using a computer card can also be referred to as **imaging**.

The installer for Raspbmc can be obtained from `http://www.raspbmc.com/download`. As instructions for downloading and running the installer are already available on this page, the instructions given in this chapter will be condensed.

> You should double-check you are installing Raspbmc to the right device before imaging. You can reduce the likelihood of imaging the wrong device by temporarily removing devices that you do not want to image, such as USB sticks.

Windows

Installation on Windows can be done by running the application from the Raspbmc website as follows:

1. Installing Raspbmc from Windows is a simple process. Upon running the program, you will see a screen similar to the following one:
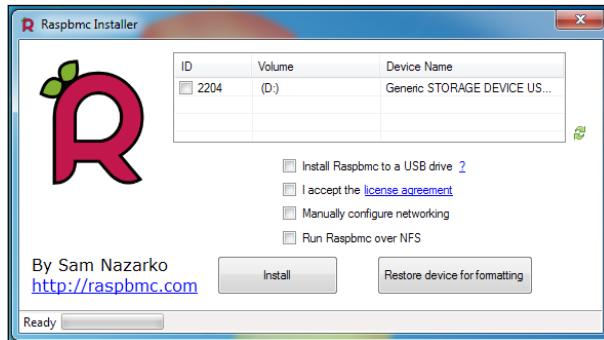
2. If your SD card is not already listed in the grid, verify whether it is inserted and click on the refresh icon. If you wish to configure advanced options, such as installing to a USB drive, or wireless networking settings, tick the appropriate boxes.

3. Simply check the SD card that you would like to image, accept the license agreement, and click on the **Install** button. This process will take up to five minutes, as your computer downloads the latest Raspbmc setup image and writes it to your SD card.

   When this process is finished, you will be given a confirmation similar to the following one:
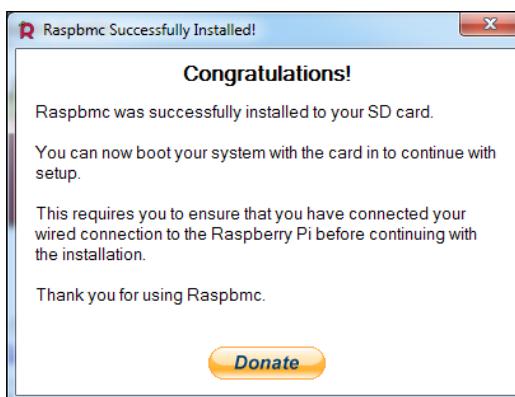
Mac OS X / Linux

An easy-to-use installer is also provided for other platforms. Let's see how to use the installer for Mac OS X:

1. The installer for Mac OS X and Linux is written in Python, and running it as per the instructions on the web page should lead you to a screen similar to the one shown in the following screenshot:
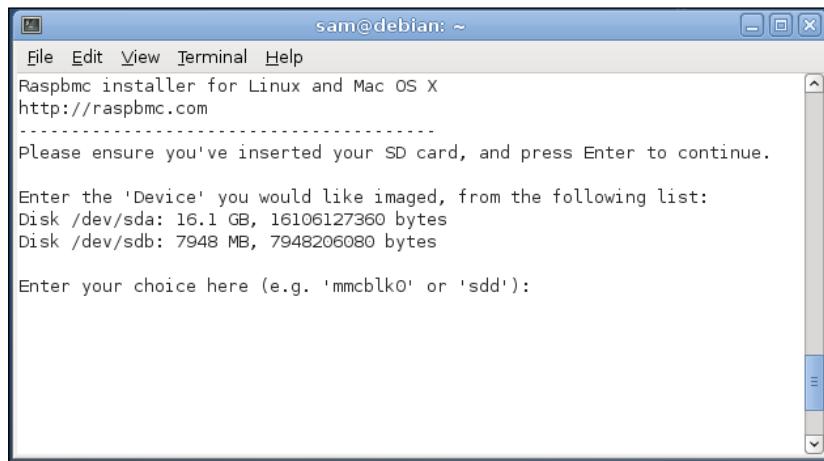
2. Simply follow the instructions to select the correct device to image. The script will then download the setup image and write this to your SD card. When this is done, the device will be safely removed, so it is safe to remove it. Similar to the Windows installer, there will be an option to configure network settings or USB/NFS installs.

Advanced Installation

In the rare instance that the installers do not work on your system, this web page also has an image that can be downloaded if the user wishes to image the card with their own software. Select the **Network Image** option for downloading. If you do not have a network connection on your Raspberry Pi, you should choose the **Standalone Image** option.

For Windows, an alternative to the Raspbmc installer is Win32DiskImager or the USB Image tool. For UNIX systems, the following command would suffice, assuming that the filename of the compressed image was `installer.img.gz`:

```
gunzip -c installer.img.gz | sudo dd of=/dev/sdb
```

Prefixing the `dd` command with `sudo` may not be necessary if you are running it as the root user. Your device may not be `/dev/sdb`; you can verify the device's path with `ls /dev`.

Booting up the Raspberry Pi

Now that we have imaged the card, we need to prepare Pi for its first boot:

1. Insert the freshly imaged SD card into the Raspberry Pi.

2. Attach a network cable or your USB wireless dongle to the Raspberry Pi's Ethernet port and the Ethernet port of a switch or router (note that this is not necessary if you have used the Standalone Image option).

3. Attach the power supply and connect either the composite or HDMI video connection so we can view the installation's progress on a display.

4. The connection to the Raspberry Pi should now look similar to the following screenshot:
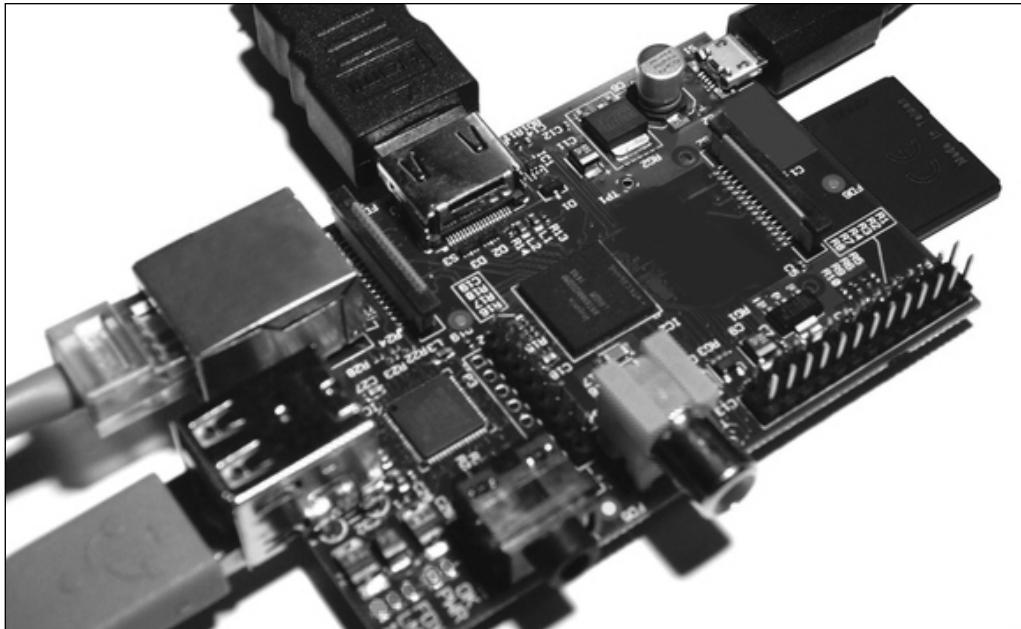
5. With everything connected, plug in the charger. You may need to change the input on your display device if this does not automatically change for you.

6. Unless you installed using the Standalone Image option, Raspbmc will now connect to the update server and install. This usually takes around 20-30 minutes on a home broadband connection, as Raspbmc must download the latest version onto your device. This process is entirely automatic, so feel free to take a break and come back later.

7. When you return, you'll hopefully be greeted by XBMC, the software at the heart of your new media center-to-be. Congratulations, you have successfully installed Raspbmc!
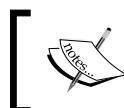
> For a few seconds, you may see a rainbow colored screen. This is to be expected and will disappear a few moments later.
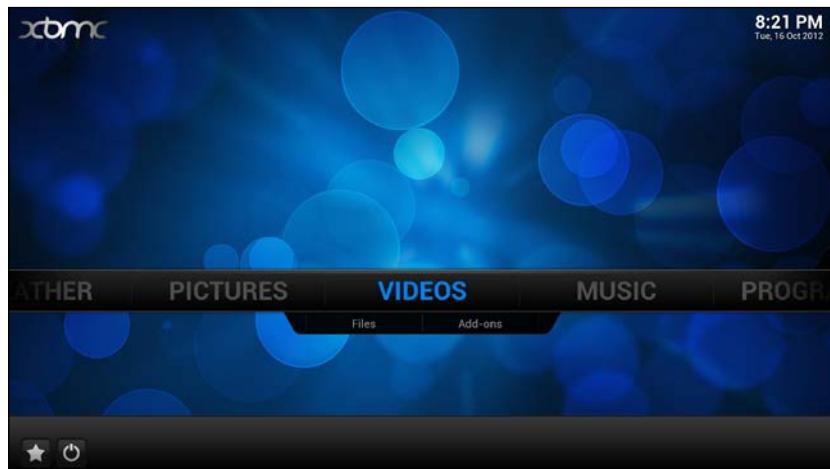
Summary

In this chapter, we covered what makes the Pi appealing as a media center device, what Raspbmc is and what it brings to the tablet, how to connect your Pi, and how to install Raspbmc. This chapter makes an ideal reference to revisit if you need to reinstall or configure another device at a later date.

In the next chapter, we'll look at how to control Raspbmc, how to navigate Raspbmc's menus, how to change important system settings, installing plugins in Raspbmc (such as iPlayer or TVCatchup), and installing additional codec support (such as MPEG2 and VC1).

2
Using Raspbmc

In this chapter, we'll explore:

- How to navigate Raspbmc's menus
- The different ways we can control Raspbmc
- How to set up important settings
- How to install add-ons in Raspbmc

Controlling Raspbmc

Now that we have installed Raspbmc, we need a way of controlling it so that we can navigate the menus. Raspbmc is a flexible system and can be controlled via a variety of devices, from an Xbox DVD remote to a smartphone.

Control via Consumer Electronic Control (CEC)

CEC allows users to use their existing remote control to control devices connected via HDMI. It is supported by devices that adhere to the HDMI 1.3 standard and up. You may have used this in another form already, such as when controlling a Blu-ray player with your TV remote.

Raspbmc has integrated CEC support provided by the Pulse-Eight CEC driver and may work for your device. When Raspbmc is started, if you see a notification stating that a CEC support adapter is connected, then Raspbmc has detected a CEC compatible device, and chances are, you will be able to control it with the remote that is shipped with your TV. Most recent TVs by major brands support CEC.

For CEC support to work, every device that the Raspberry Pi is connected to must be CEC compliant. For example, if your Pi is connected to an AV receiver that is CEC compliant, your TV must also support CEC for this to work. This allows your AV receiver to *pass through* commands sent by your TV remote to the Raspberry Pi:
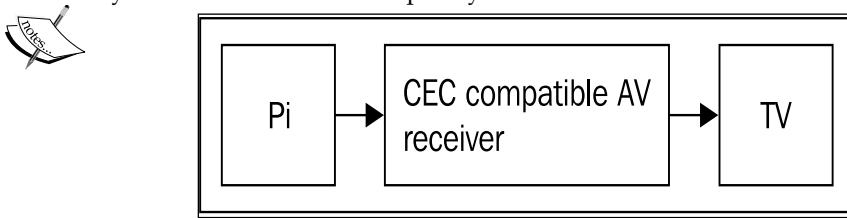
If CEC is working for you, then congratulations, you do not need any additional hardware to control your Raspberry Pi. You can find a comprehensive breakdown of supported devices at the Pulse-Eight website:

`http://libcec.pulse-eight.com/vendor/support`

Control via iOS and Android or any web browser

XBMC can be controlled over a network. This means Raspbmc can be controlled via your smartphone/tablet, or any device with a web browser.

There are a variety of remotes available on each platform; the best way to find them is to enter the Apple App Store or Google Play and search for XBMC. This will return several applications that are able to control Raspbmc. I recommend the Yatse application for Android and Constellation for iOS, but you are encouraged to explore your phone's store yourself.

After installing the remote onto your device, you need to add Raspbmc to the remote's connection list so that it can connect to it. Depending on the capabilities of the remote software you downloaded, Raspbmc may or may not be automatically detected depending on your network's capabilities. If the case is the latter, you will need to know the IP address of the Raspberry Pi. This can be obtained either by checking the router's list of attached devices (look for a client named `raspbmc`), or by checking the availability of another method for controlling the Pi, such as via the keyboard or by going to **System** | **System Info** | **Network**. Raspbmc's **Settings** should also inform you of your IP address based on a successful connection. This is probably the easiest way.
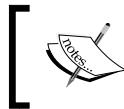
Using one of the web-based remotes always requires an active network connection on both the Raspberry Pi and the iOS or Android device that is controlling it.

Controlling via keyboard and mouse

Raspbmc can be controlled via a keyboard and a mouse, which can be attached to the Raspberry Pi's USB ports. It is not recommended to use a mouse because some skins do not support mouse control, and a keyboard is much more favorable for navigating menus anyhow. Keyboard control may seem cumbersome; however, a wireless keyboard can prove to be very powerful (there are some keyboards that are also designed for media center usage), particularly for searching large libraries (we'll touch on library support in XBMC later on in *Chapter 5*, *Setting up the Library Mode*).

Alternative remotes

Raspbmc constantly gains support for other remotes, and the best way to find out if your remote is supported is to check out the remote support page at `http://www.raspbmc.com/wiki/user/configuring-remotes/`.

Navigating Raspbmc

After installing Raspbmc for the first time, you will be presented with an interface much akin to this:
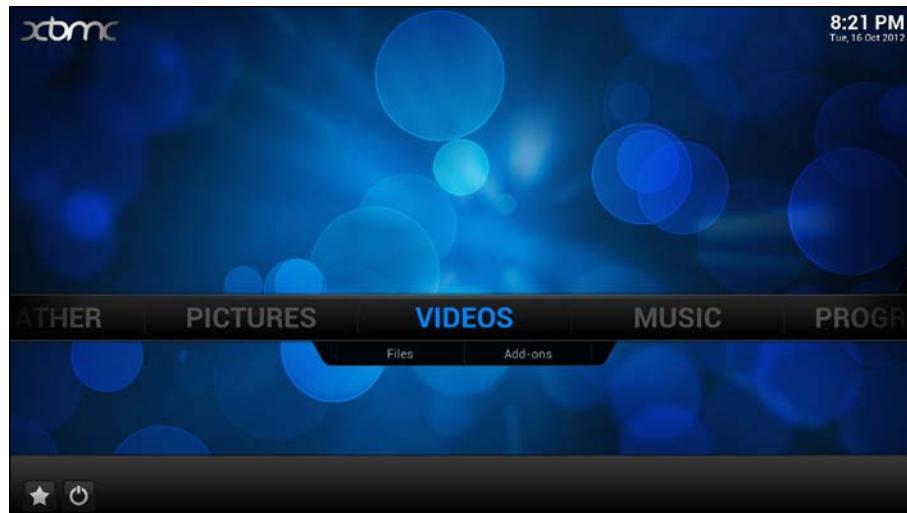
This menu makes available to us the following options:

- **Videos**: For accessing videos that are not in your library
- **Music**: For accessing non-video formats, such as music
- **Movies**: For accessing videos that have been scraped as movies
- **TV Shows**: For accessing videos that have been scraped as TV shows
- **Programs**: For accessing scripts and plugins, such as the Raspbmc Settings add-on
- **Settings**: For changing XBMC settings

We'll touch on adding content to the **Videos** and **Music** sections later. For now, we'll explore how to change settings.

Changing settings

Settings are configurable for both the Raspbmc distribution itself and XBMC.

Changing XBMC Settings

To change the XBMC settings, first go to the **System** option on the menu and then select **Settings**. You will now be greeted with the **Settings** menu, which allows you to change key settings as follows:

- To change locale issues, such as your time zone and keyboard layout, go into the submenu entitled **Appearance** and change it via the **International** tab
- To change playback settings, such as whether to adjust the refresh rate on playback, go into the submenu entitled **Video** and adjust it in the **Playback** tab
- You can also adjust settings such as screensaver settings by going again into the submenu entitled **Appearance** and selecting the **Screensaver** tab

XBMC is highly configurable and the best way of exploring its flexibility is to navigate the menus and learn more about the settings available. A good place to start is `http://wiki.xbmc.org/index.php?title=Settings`.

Changing Raspbmc Settings

The Raspbmc Settings application is for configuring the settings applicable to the Raspbmc OS itself rather than XBMC. This is accessible via **Programs | Raspbmc Settings** from the main menu. The icon looks as follows:

After opening **Raspbmc Settings**, you will be presented with a dialog box that allows you to configure Raspbmc-specific options. Here are some common changes you may wish to make:

- Raspbmc updates itself by default. This can be turned off by unchecking the radio button entitled **Keep Raspbmc updated** under the **System Configuration** tab.

- You can configure wireless networking under the tab entitled **Wireless Network Configuration**. Simply fill out your access point details using a compatible wireless dongle. Refer to *Appendix A*, *Useful Raspbmc commands* for a list of supported adapters.

- You can also change the default password for the **pi** user, which is **raspberry**, under the **System Configuration** tab. This has been shown as asterisks in the add-on for security reasons.

There are many more configuration options, and more of these will be touched on as we expand the capabilities of our media center.

Adding additional codecs to Raspbmc

If some video content you wish to play back on Raspbmc is encoded in either VC1 or MPEG2 support, you will need a license pack to support the playback. This is because these codecs require a license for each device they are used on, due to software patents. You only need to do this if some of your content is in VC1 or MPEG2. Most VC1 content played back by users is from direct Blu-ray rips, and most MPEG2 content are in the form of Live TV streams. Getting a codec license involves a few steps, which are as follows:

1. Firstly, we need to get the serial number of the Raspberry Pi. This is obtainable by going to **System** | **System Info** | **Hardware information**. This serial number is needed to purchase a license. This is tied to the Pi you purchase from, and the license is not transferrable.
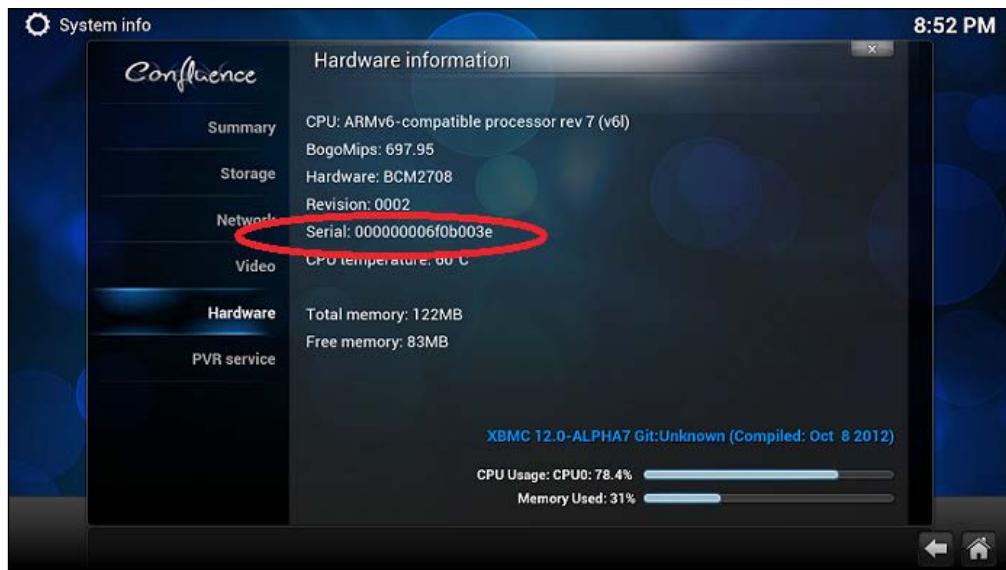
2. Now, purchase the codec pack(s) desired using your serial number from `http://www.raspberrypi.com/license-keys/`.

3. Shortly, you will get a license key via an e-mail. This can now be entered via the Raspbmc Settings' **System Configuration** tab. You should only enter the code after the = sign. After a reboot, the codec(s) will start working.

The following screenshot shows where you would enter the codec keys:
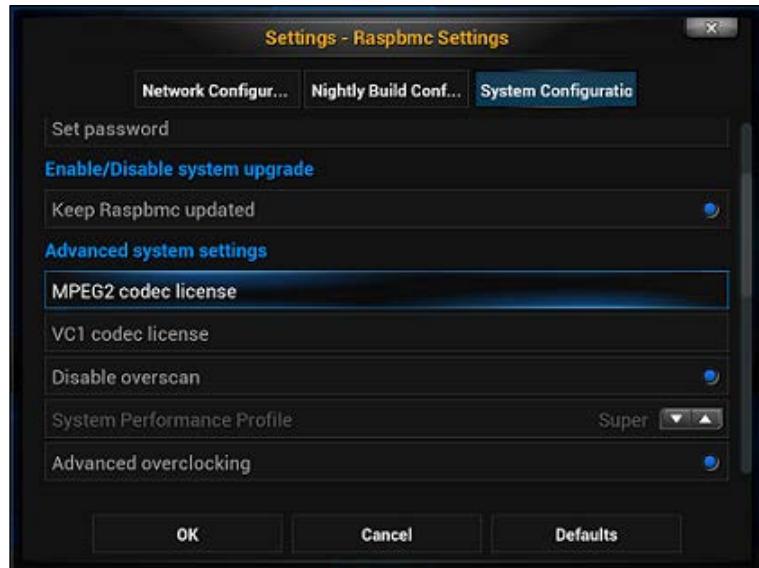
Installing add-ons

XBMC supports plugins, which can enhance the media center experience by
providing access to additional content as well as skins and new features. For
example, the BBC iPlayer plugin allows you to watch on demand BBC content
as well as a selection of live BBC channels.

Installing add-ons via XBMC's standard
repositories

Some add-ons are available for installation from XBMC itself without any manual
downloading. These are considerably easier to install than other add-ons.

The following steps will help you to install an add-on via XBMC:

1. Go to **System** | **Settings** | **Add-ons**.
2. Select **Get Add-ons**.
3. Pick **XBMC.org Add-ons**.
4. There are now many different types of add-ons available for selection. To install one, select the add-on and click on **Install** as shown in the following screenshot:
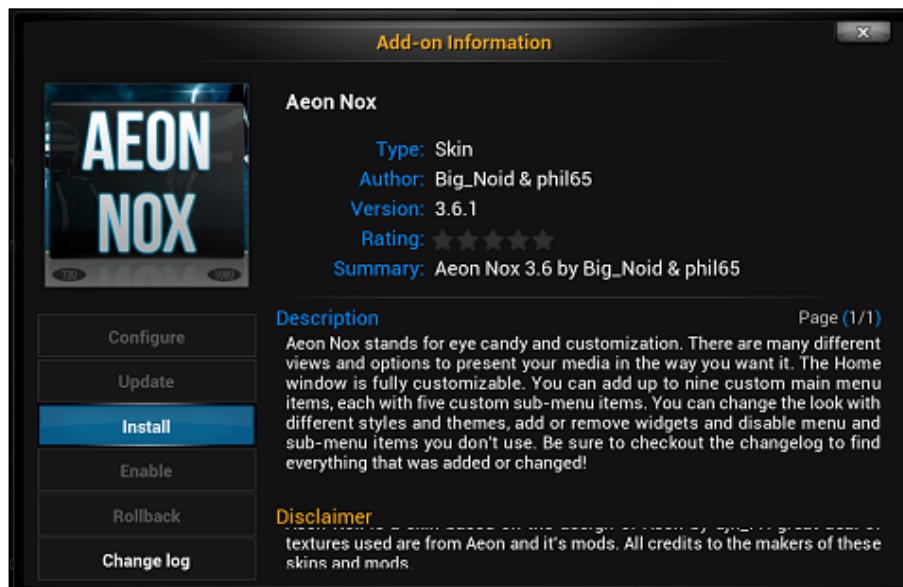
5. The download progress will be shown next to the name of the add-on in the list.
6. The add-on will now be installed on your system. If it was a video plugin, it should be visible within **Videos**. Similarly, music plugins will appear within **Music** if they offer services such as streaming.

   It is best to consult the documentation of the downloaded add-on to find out how to use it in XBMC.

   Add-ons that say **Broken** next to them are known to have issues. These are usually resolved by their maintainers within a few days. It is advised not to install broken add-ons.

Installing add-ons manually

Some add-ons are available for XBMC but cannot be found in the **XBMC.org Add-ons** repository that we used previously.

To install an add-on, we need to add the repository that contains it first or install the add-on directly via a ZIP file. Many add-ons are available, and a list of unofficial repositories is available at `http://wiki.xbmc.org/index.php?title=Unofficial_add-on_repositories`.

In this example, we will install the BBC iPlayer add-on as follows:

1. First, we need to download the ZIP file of the add-on. This should be done on a computer and the file should be saved somewhere memorable. In this case, our ZIP file is at `http://xbmc-iplayerv2.googlecode.com/files/iPlayer-v2.4.17.zip`.

2. An easy way to transfer the file to Raspbmc is using the built-in FTP server. This should be on by default unless you disabled it in **Raspbmc Settings**.

3. To connect to FTP via Windows, enter `ftp://IP` in Explorer, where IP is the IP address of the Raspberry Pi. The IP address can be found in **System | System Information** under the **Network** pane. The following screenshot shows where to type this URL in Windows:
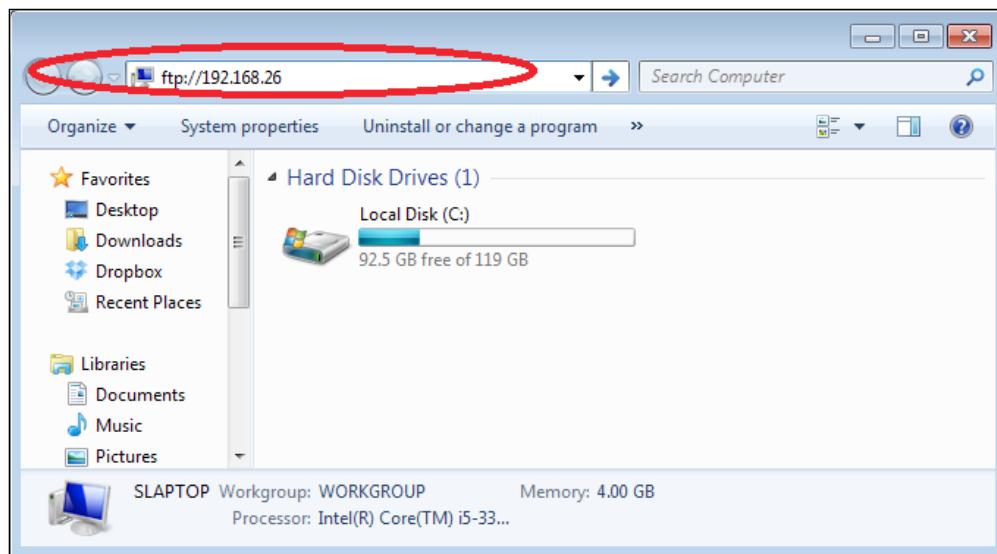
4. You will now be prompted for the login details. The username is `pi`, and provided you have not changed the password in **Raspbmc Settings**, the default password is `raspberry`. This can be entered via a dialog similar to the one shown in the following screenshot:
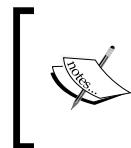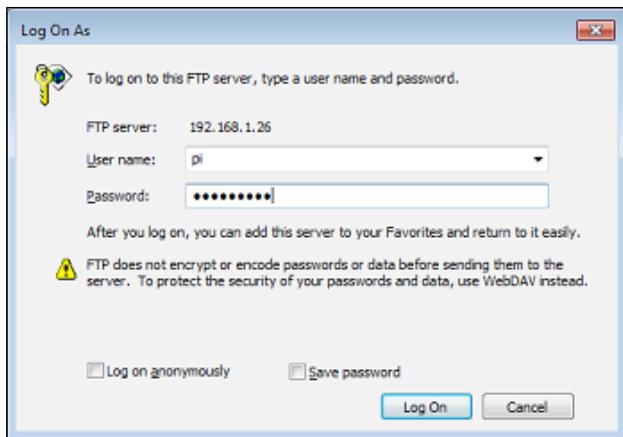
> If you're using Linux, Mac OSX, or simply don't like Windows' built in FTP client, you can use any client. Just remember that the username you'll need is `pi`, and the password by default is `raspberry`, unless you have changed it.

5. Right-click on the ZIP file and click on **Copy**.

6. Right-click on an empty area in the FTP folder and click on **Paste**. The ZIP file will now be uploaded to your device (shown in the following screenshot):
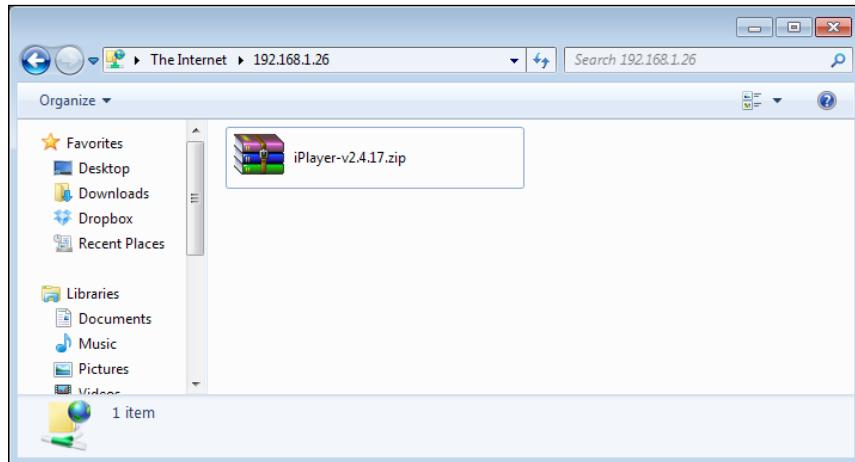
Now, we can install the plugin via the Raspberry Pi.

1. We now go to **System | Settings | Add-ons** as we would if we were installing an add-on from the XBMC Repository.

2. However, instead of choosing **Get Add-ons** we should select **Install from zip file**.

3. Select **Home folder**, as that is the default location of the file we have uploaded.

4. Select the iPlayer ZIP file, and XBMC will now install the add-on. The following screenshot shows how to select a ZIP file for installation:
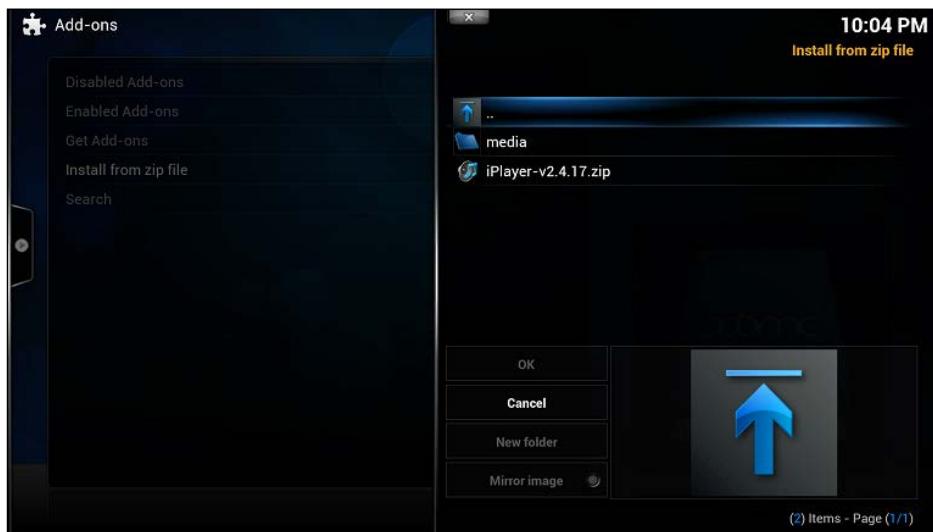
5. You will see a notification informing you that the iPlayer add-on is now enabled.

6. Congratulations! This can be used by selecting **Videos** from the main menu and then going to **Video add-ons**.

> If we had just installed a repository via a ZIP file rather than a plugin, we would then proceed to **System | Settings | Add-ons | Get Add-ons** to then download the add-ons via the newly added repository. It should be noted that manually installed add-ons do not get automatically updated.

Summary

And that's it for this chapter. We've now covered how to control Raspbmc and what remotes are available. Furthermore, we've looked at configuring settings for both XBMC and Raspbmc. Lastly, we touched on enhancing the XBMC functionality by adding additional codec support and installing add-ons.

In the next chapter, we'll look at sources from where we can play content in Raspbmc, adding USB drives and sticks to Raspbmc for playback, and adding, deleting, and pin-protecting sources in Raspbmc.

3
Managing Content

In this chapter, we will take a look at:

- How we can add our existing content to Raspbmc for playback
- The different types of backends that Raspbmc supports
- How to add these as sources and manage them as well as best practices

Supported backends

A backend is a source that XBMC can play content from.

Overview

Raspbmc supports playback of both music and video content from the following source types (this does not cover all of them):

- USB drives
- SD card content
- HTTP: Playing back content over a standard web protocol
- FTP
- SMB: Windows and Linux shares
- NFS: The network filesystem, typically used in Unix
- AFP: **Apple Filing Protocol** (**AFP**) is typically used for sharing content across the network from Mac OS X systems
- UPnP/DLNA: Allows streaming from capable sources, such as UPNP streaming software on laptops
- AirPlay: This entails enabling the option **Allow XBMC to receive AirPlay content** by navigating to **System** | **Settings** | **Network**

> We won't touch on how to configure network shares such as SMB or NFS as the set up is very much dependent on your operating systems and its configuration. You may wish to consult your operating system's documentation for details on how to configure a network share.

USB drives

We'll only look at setting up play back from a USB device for now, as we'll explore configuring network sources for streaming later.

Raspbmc supports USB drive support out of the box, allowing you to plug in a USB drive and be able to play back any supported media files that are on the device. This compatibility is dependent on the hard drive being formatted with one of the following filesystems:

- Ext2, Ext3, and Ext4 filesystems (typically found on Linux systems)
- HFS and HFS+ (typically found on Mac OS X systems)
- NTFS, FAT32, and exFAT (typically found on Windows systems)

> When using a Windows filesystem there are some limitations to consider. When using FAT32, filesystems cannot exceed a file size of 4 GB. It is recommended to use the exFAT filesystem instead of NTFS on USB sticks as there will be no permission issues. Note that Windows XP requires KB955704 for exFAT support.

We will now look at formatting a USB drive as exFAT under Windows:

1. Inarguably, exFAT is the best choice of filesystem if you are planning to connect your USB drive to your Windows PC and your Raspberry Pi regularly.
2. On Windows, we need to locate the drive. This can either be done under **Explorer** or **Computer**.

3. Right-click on the USB drive and click on **Format** as shown in the following screenshot:
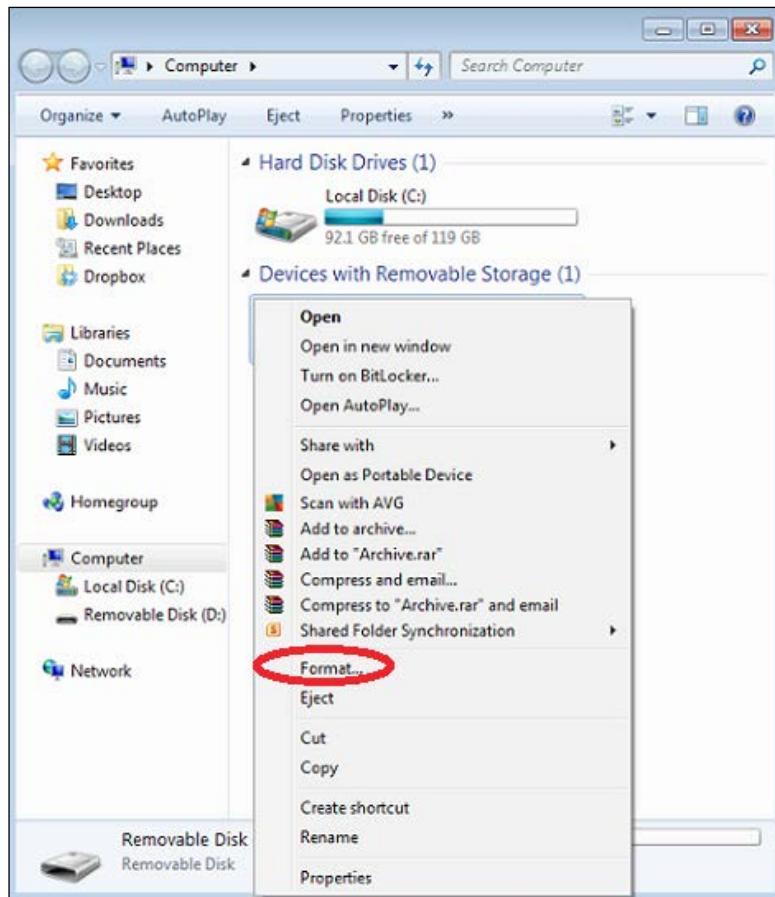
4. Now, ensure that you select the **exFAT** filesystem and **Quick Format** for format speed, as shown in the following screenshot:
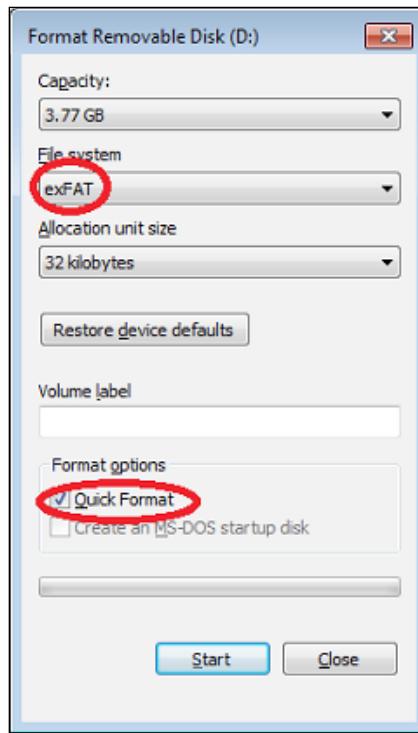
5. Click on **Format**.

6. When the process is complete, a dialog box will appear. You can now close the format window.

7. Your drive is now usable for Raspbmc. You can save files using Windows Explorer just as you would with any other drive.

I have not covered formatting a USB drive under Mac OS X or Linux as the filesystems are less limited for these platforms and do not suffer from caveats to the extent that the ones available to Microsoft Windows users do.

Sources

It's now time to get XBMC to use our content, which can be done from a variety of locations and protocols.

Adding sources in XBMC

Adding sources allows XBMC to access the locations we add and play content from them.

The following steps show how you can add a source in XBMC:

1. From the main menu, open the relevant section based on the type of source we are adding. For example, if we are adding a video source, we should select **Videos**.

2. Now, select the files.

3. If you are planning on using a USB device, you will see that they are already shown; however, the device can be defined as a source as well. The advantages of this are as follows:

   ° When using Library Mode, something that we will explore in *Chapter 4*, *Streaming from a Computer*, you may have different types of content on the drive—such as TV shows and movies—in different folders. Setting the content type can only be done at the drive level if this is treated as a source.

   ° Pin-code locking is unavailable.

   ° Setting content types this way may not always be persistent.

4. We would now click on **Add Videos** if we were adding a video-based source.

5. This presents us with an interface shown in the following screenshot:
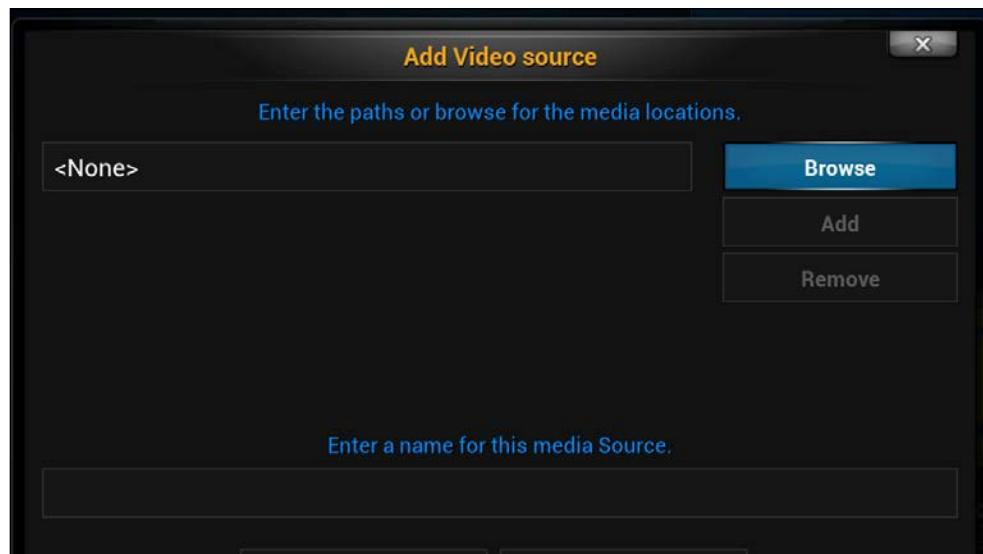
6. We can navigate through this dialog with the arrow keys on a keyboard or our remote.

7. The field that initially says **<None>** is a field that can contain a path to content. As we can see, a source can be made up of several different media locations. This is particularly useful if your content spans two drives.

8. Next, click on **Browse**. Here, we will locate the paths that will make up our sources.

9. To add a USB drive that is connected to Raspberry Pi, click on **Root Filesystem** and then select the **media** directory. Your disk will be there, listed by its volume name. If the device has no volume name, then its UUID will be listed. Click on **OK** to add it as a location.

10. To add a network location, such as an NFS or SMB share, select **Add network location…**. There is another way of adding these sources, that is, by selecting **Windows network (SMB)** or **Network Filesystem (NFS)** from the **Browse** menu; however, this is not as reliable as manually adding the shares, and the first option will allow us to add FTP and HTTP shares too.

11. Select the type of network location we are trying to access, under **Protocol**.

12. Enter the relevant details. If the share does not have a username or password set, you can leave these fields blank. For example, if I were adding an SMB share for my TV Library, I may fill it out as shown in the following screenshot:

13. Click on **OK** to add this as a location. It will now appear in XBMC as shown in the following screenshot:

> We take a further look at network-based streaming in *Chapter 4, Streaming from a Computer*.

14. Select the location so that its contents are visible onscreen and click on **OK**. It will now be added as a location.
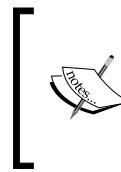
> While the option to **Browse** exists for SMB and NFS shares, this is not always reliable and as such it is better to define the network share manually if possible.
>
> In the example shown, either a hostname or IP address would have been acceptable for the **Server name** field.

15. We can do this repeatedly until we have added all the locations we would like to define as that source. Simply click on **Add** each time you wish to add another source, and then **Browse** as before.

16. Only locations with the same type of content should be grouped together, that is, locations with music, locations with movies, and locations with TV shows should be added as separate sources. This will help us with Library Mode later.

17. By the end of this, we will now see locations defined along the left-hand side of the dialog. In the following screenshot is an example of a source with multiple locations:
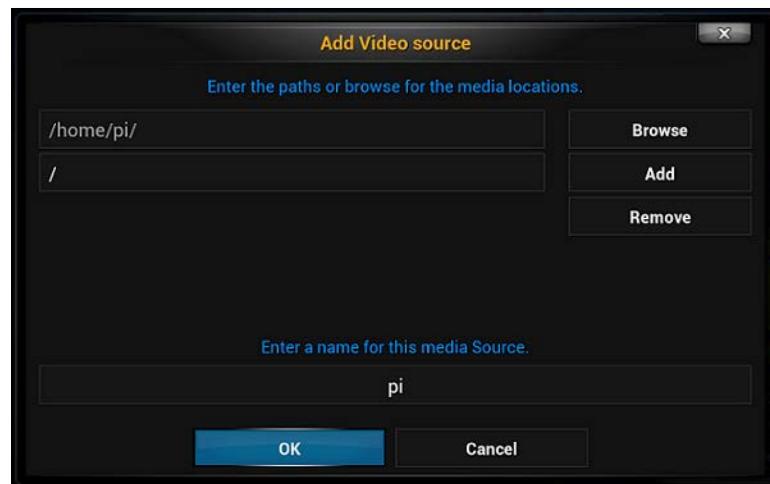
18. You should now proceed to name the source, and then click on **OK**.

19. Ignore any dialog titled **Set Content** for now, we'll look at that in a later chapter. For now, just click on **Cancel**.

The source will now be visible in XBMC.

Managing sources via the context menu

To manage sources effectively, it is vital to learn about the context menu. The context menu provides additional source options such as the ability to edit, delete, and password protect sources.

Invoking the context menu is different depending on your method of controlling Raspbmc. Using a mouse, one can right-click, this is also the button that should be pressed on some remotes. However, when using a keyboard, pressing the *c* key will bring up the menu. On the MCE remote and Xbox DVD remote this is the **guide** key. This menu should appear as shown in the following screenshot:
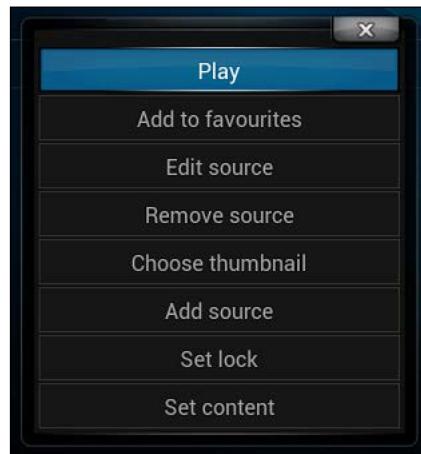
You can edit the locations of the sources you added before at any time by invoking the context menu and selecting **Edit source**.

Deleting a source

Deleting a source can be done as follows:

1. Enter the screen to which the source was added, that is **Videos/Music/Pictures**.
2. Select the files.
3. Highlight the source and then bring up the context menu.
4. Select **Remove Source**.
5. You will be prompted if you wish to remove all references of the source from the library. It is recommended that you select **yes**.

The source will now be removed.

Pin locking a source

Locking a source allows you to restrict who can access it. It should be noted that the implementation is not very strong in XBMC, and this can be trivially circumvented as the pin is stored as text in the XBMC data folder.

Nonetheless, it can still be useful. A prime example would be restricting all sources that are not PG if a young child is using the media center as well.

A pin lock can be set via the context menu; however, **Master Lock** must be enabled in the system settings first:

1. Go back to the main menu if you are not already on it.
2. Go to **System | Settings**, and then select the **System** tab.
3. Go to the **Master lock** pane and select **Master lock code and settings**.
4. You will now be prompted with a dialog allowing you to choose which parts of XBMC you wish to protect with a code.
5. You can choose the kind of master lock implementation from the following choices: **Numeric password (recommended) / Gamepad button combo / Full-text password**.
6. Once you have done that click on **OK**.
7. Find the source you wish to protect and highlight it.
8. Bring up the context menu.
9. Select the **Set Lock** menu option.
10. Enter a code and click on **Done**.

> You do not need to lock any parts of XBMC with a code to lock sources, you just need the **Master lock** feature enabled.

Selecting the source will now prompt you for a code.

Playing/viewing a file

Now that you have defined your sources, you can easily play or view media by going to the source, highlighting it, and selecting a file. Congratulations! Raspbmc is now playing your media content.

Summary

After reading this chapter, you should now know what type of sources Raspbmc supports, the supported filesystems, and the best method of formatting a USB drive in Windows. Lastly, we looked at adding sources in Raspbmc as well as using the context menu to edit and remove them later on. We finally touched on pin locking sources.

In the next chapter, we will look further at streaming our content from a computer, exploring what makes a system a good candidate for a media server, how to configure shares, and how they can be added to Raspbmc.

4

Streaming from a Computer

In this chapter we will cover:

- All that is needed to stream media from your computer to Raspbmc
- The advantages and disadvantages of remote streaming and local playback
- Setting up streaming on Windows, Linux, and Mac OS X based computers

Remote streaming

We will primarily focus on streaming content from a remote system.

Remote streaming versus local playback

While remote streaming has some distinct advantages, it is not often necessary for this kind of setup. I've outlined a few reasons why remote streaming is advantageous and a few reasons why it may not be.

Advantages of network streaming	Disadvantages of network streaming
Content can be streamed from one machine to multiple devices simultaneously.	Remote system must be on whenever you wish to play content from it.
Maintaining your library across all devices can be done by updating a single location. Local playback requires identical content across multiple storage devices for each device to have the same content.	Higher energy consumption because the device your Pi is streaming from must be on as well.
Backups of media can be performed more easily as it is in a single location.	Generally, this works best on a server (a computer dedicated for the role of remote streaming), which can involve extra configuration steps.
A single system allows for larger hard drives, which gives a better £/GB ratio.	Central point of failure if the system is not properly maintained or backups are not made.

As we can see, if we are only planning to play content on one device at a time, it makes more sense to play the media locally. This can be achieved by attaching a USB drive to the Raspberry Pi, as we covered in *Chapter 3*, *Managing Content*.

If you are not interested in configuring remote streaming, you may wish to skip this chapter.

Requirements for remote streaming

At the beginning of this book, we looked at why the Raspberry Pi was an ideal candidate for media center hardware. Likewise, there are suitable criteria that should be taken into consideration when choosing a computer to stream content from. These are ideally as follows:

- **A wired network connection**: A wireless connection can fluctuate in signal strength and throughput, which can cause stuttering or frequent buffering due to dropouts and limited bandwidth. Protocol overhead can also affect this.

- **Low power consumption**: A powerful computer is not particularly necessary for streaming, so we do not need to use components that consume a lot of power. Intel Atom barebones kits are available for around £50 and have very low power requirements. Alternatively, one can purchase a NAS, which provides a simple interface and easy storage configuration.

- **Device dedicated to streaming**: Ideally, for best performance, the computer should only be used for streaming media and should be separate from desktop activities.
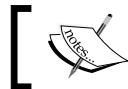
> While these characteristics are ideal for remote streaming, they are not a necessity.

On this system, we will set up network shares, which can then be added in XBMC on the Raspberry Pi to access content remotely.

Configuring network shares

Network shares permit the streaming of content remotely. We'll now look at how to achieve this on multiple platforms.

Configuring network shares on Windows

Windows has a built-in client-server implementation called **Common Internet File System** (**CIFS**). It allows access to files on remote shares that use the CIFS implementation as well as serving files to other systems that have an SMB/CIFS client implementation (such as Raspbmc).

Note that while all versions of Windows have the same CIFS client implementation, the CIFS server implementation in Windows varies depending on the edition you have. For example, *Professional* and *Ultimate* versions of Windows have support for Active Directory domains and thus allow for more fine-grained control over sharing permissions.

To share our content with Raspbmc, we will take advantage of Windows' built-in CIFS server implementation by creating a share as follows:

1. First, we should either create or locate a folder which we will share the contents of.
2. We should preferably ensure that the contents of this folder are all of the same type (that is, TV shows, movies, and music). This will make configuring Library mode much easier later on.

3. Right-click on the folder and click on **Properties**, as shown in the following screenshot:
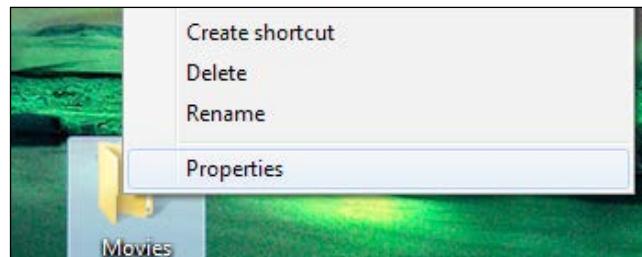

4. Now, click on the **Sharing** tab.
5. Click on **Advanced Sharing**. We will use the **Advanced Sharing** settings so we can have fine control over **Share name** and simplified access to the source.
6. Check the option entitled **Share this folder** and give the share a meaningful name as shown in the following screenshot:
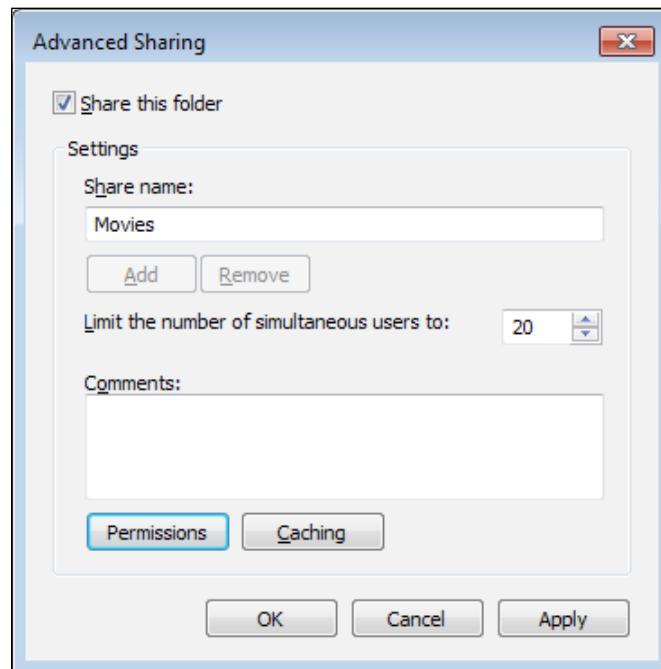
7. By default, the share will grant all users *read* access. If you need to change this, you can do so by clicking on **Permissions**.

8. Click on **Apply** and then on **OK** when you are done configuring the share. You can now close the **Properties** window.

We have now created a share. You can repeat these steps for all the different types of content you wish to serve, such as music, TV shows, and movies.

Your share is now accessible via `\\COMPUTER-NAME\\SHARE-NAME`, where `SHARE-NAME` is the name of the share you created (see the previous screenshot).

> Some home networks do not perform name resolution reliably. Thus, it is best to assign the computer that is to stream media a static IP address and substitute it for the computer name, for example, `\\192.168.1.2\\SAM-MOVIES`.

Configuring on Linux

We'll first look at installing the necessary packages for **Network File System** (**NFS**), and then we'll look at configuring them for our desired distribution.

Installing NFS

File sharing on Linux is typically done using the NFS protocol. This is advantageous to SMB as it has a lower CPU overhead (which is particularly ideal considering the CPU constraints of the Raspberry Pi) and is well implemented across platforms.

NFS server implementation can be done via the kernel or userspace. We will cover kernel-based implementation as userspace implementation is often deprecated in distributions.

The ensuing steps will assume you are running as the root user. In most distributions, you can also elevate your permissions using the `sudo -s` command.

Installation of NFS on Ubuntu/Debian/Linux Mint

Typically, these instructions will apply to any Linux distribution using **Advanced Packaging Tool** (**APT**) as its package manager. We'll need to enter these commands at the command line. In a UI, this can be accessed by opening the Terminal application:

```
apt-get update
apt-get install nfs-common nfs-kernel-server portmap
```

Installation on CentOS/Fedora/Redhat

Typically, these instructions will apply to any Linux distribution that is a derivative of Redhat and uses **Redhat Packaging Manager** (**RPM**) as its package manager. Here, you'll need to enter these commands at the command line, which can be accessed via Terminal if you are using a UI.

```
yum install nfs-utils nfs4-acl-tools portmap
```

Installation on Arch Linux

On Arch Linux, we can install the NFS utilities package via the command line, as follows:

```
pacman –S nfs-utils
```

Distribution independent configuration

We will now configure portmap. Note that, on some systems, this package is deprecated and replaced by `rpcbind`. Your package manager should inform you what package was selected. Follow the appropriate setup instructions given here:

- For `portmap`:

  ```
  perl -pi -e 's/^OPTIONS/#OPTIONS/' /etc/default/portmap
  echo "portmap: 192.168.0." >> /etc/hosts.allow
  service portmap restart
  ```

  You should replace `192.168.0.` in the example shown with your subnet, for example, `10.12.0.0.`. This will allow devices on your LAN to access the NFS shares.

- For `rpcbind`:

  Using your favorite text editor, edit the following line in `/etc/default/rpcbind`:

  ```
  OPTIONS="-w -l -h"
  ```

  Now, we can allow computers on the LAN to connect to our shares using the following command:

  ```
  echo "rpcbind: 192.168.0." >> /etc/hosts.allow
  ```

  You should replace `192.168.0.` in the preceding command with your subnet.

> If you have a firewall installed, you should allow access to the ports `UDP2049`, `TCP2049`, `UDP111`, and `TCP111`.

Now that we have installed the necessary packages and configured the system, we need to create shares. This is called **exporting** as we are exporting a filesystem over NFS.

Shares are defined in `/etc/exports`. Let's open this file for editing with our favorite editor:

```
/home/sam/tvshows *(ro,all_squash,sync,no_subtree_check)
```

This example will export the folder `/home/sam/tvshows` to all systems and grant them read-only access. As this is a read-only filesystem, we could in fact specify the `async` parameter, which will improve performance. More information about the various NFS export options can be found in the Linux manual pages. After we make changes to this file, we need to restart the NFS server so that it will pick up the changes. This can be done as follows:

```
exportfs -a
```

Hopefully, we have now set up NFS file sharing.

> **Check what's shared**
> You can easily check shares you are exporting with the following command:
> **`cat /etc/exports`**

Configuring on Mac OS X

OS X embraces **Apple Filing Protocol** (**AFP**) as its native file sharing protocol. After Lion, the SMB implementation changed, which caused issues with XBMC. Now that XBMC has gained native AFP support, configuring AFP is considered best practice for sharing files from a Mac.

1. Click on the Apple logo and then select **System Preferences**.
2. Click on the **Sharing** icon.
3. Check the **File Sharing** checkbox if it is not checked already.
4. Add any folders you wish to share by clicking on the **+** icon.
5. Click on **Options** and check **Share files and folders using AFP**.
6. Check the **On** checkbox for all users you wish to share these folders with.
7. Click on **Done**.

Adding these shares in Raspbmc

Now that we have created our shares, it is time to add them in Raspbmc. As we covered this in *Chapter 3*, *Managing Content*, it should not be necessary to go over this in great detail. Here is a short summary of how to get things working:

1. From the main menu, select the submenu of the type of content we are trying to add. For example, if we are adding video, select **Videos**.

2. Now, select your files.

3. We would now click on **Add Videos** if we were adding a video-based source.

4. Select **Browse**.

5. Select **Add Network Location**.

6. Select the type of network location we are trying to add:
    ° If you are sharing files from Windows, choose Windows network (SMB)
    ° If you are sharing files from Linux, choose Network File System (NFS)
    ° If you are sharing files from Mac OS X, choose Apple Filing Protocol (AFP)

7. Fill out the share details, such as the name of the shared, the IP, or hostname of the system sharing it, and any username or password that may be needed to access the share. In our examples shown earlier, neither the Windows or the Linux demonstrations would require a password for access.

8. Click on **OK** to add the source to XBMC's directory browser.

9. Click on **OK** again to define this as a location for the new source.

10. Name the source and click on **OK**. The source should now appear in XBMC and be ready for use.

Summary

We've now covered how to set up shares in Windows, OS X, and Linux, what makes a computer ideal for remote streaming, and when remote streaming is ideal. We also clarified how to add shares as sources in XBMC.

In the next chapter, we'll take sources one step further by delving into Library mode, which will let us scrape data such as episode descriptions and fan artwork from the Web. We'll also look at sharing this library between multiple XBMC installations by using a central database.

5
Setting up the Library Mode

In this chapter, we will look at the following:

- Setting up the Library mode that will store information about our content
- Sharing this across multiple machines with a MySQL database
- Setting up Raspbmc to use this database

> If you are planning to use a MySQL database to share the library across multiple devices, you should read the section on installing and configuring a MySQL database first.

Installing MySQL

It's best to install MySQL on the same system we'll be serving files from. This makes sense as we'll need both the file server and the database server to be online to play content, so we may as well confine these services to one system.

Installing on Linux

The MySQL package can be fetched via your distribution's default repositories, as follows:

1. First, we need to fetch the packages using our distribution's package manager as follows:
   - On Debian/Linux Mint or Ubuntu systems, we can do this with the following command:

   ```
   sudo apt-get install mysql-server
   ```

       ° On Redhat, CentOS, and Fedora systems, we can do this with the following command:

```
sudo yum install mysql-server
```

2. Now, we need to allow remote access to the MySQL database service. We can do that with the following command:

```
sed -i /etc/mysql/my.cnf -e 's/bind-address/#bind-address/'
```

3. We now need to restart the MySQL service, which can be done with the following command:

```
sudo service mysql restart
```

Installing on Mac OS X

The MySQL package must be obtained from the MySQL website in OS X. The following steps will help you install MySQL on Mac OS X:

1. Head on over to `http://dev.mysql.com/downloads/mysql/` and select **Mac OS X**. Download the latest version of MySQL in the DMG format.

2. Open the DMG folder and run the `mysql-5.*.*.pkg` file. This will launch the MySQL installer.

3. Perform a standard installation of the MySQL package.

4. We will want our MySQL server to launch on startup. Go back to the contents of the DMG folder and run `MySQLStartupItem.pkg`.

5. Perform a standard installation of the `MySQLStartupItem` package.

6. Return to the contents of the DMG folder again and launch `MySQL.prefPane`. You will be prompted to install a preferences pane for MySQL. Install this for all users of the computer.

7. Open MySQL preferences. Start the MySQL server and check the checkbox entitled **Automatically start MySQL Server on startup**.

8. Open Terminal and enter `touch .bashrc`. Using your favorite text editor, add the following command to this file:

```
export set MYSQL_HOME=/usr/local/mysql-5.5.18-osx10.6-x86_64
export set PATH=$PATH:$MYSQL_HOME/bin
```
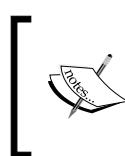
> You should substitute the version number in the preceding example for the actual version you have installed. You can verify this with the following command:
>
> ```
> ls /usr/local| grep mysql
> ```

9. Lastly, we need to allow remote access with the following command:

   ```
   sed -i /etc/mysql/my.cnf -e 's/bind-address/#bind-address/'
   ```

   You should log off and log in again so that the new MySQL variables defined in `.bashrc` are exported.

Installing on Windows

The MySQL package for Windows can be fetched from the MySQL website. The following steps will help you install MySQL on Windows:

1. Grab the MSI installer of MySQL server of the appropriate architecture (32 or 64 bit) from `http://dev.mysql.com/downloads/mysql/`.

2. Run the installer and perform a standard installation.

3. Click on **Next**, accept the license agreement, and click on **Next** again.

4. Choose a **Typical** installation and click on **Install**.

5. When the installation is complete, ensure that **Launch the MySQL Instance Configuration Wizard** is checked and click on **Finish**.

6. Choose the **Detailed Configuration** option and click on **Next**.

7. Select **Developer Machine** and click on **Next**.

8. Select **Multifunctional Database** and click on **Next**.

9. You may wish to choose where to install the database datafiles, however the default location should be acceptable. Click on **Next** when done.

10. Select **Manual Setting**, specify a maximum of 10 connections, and click on **Next**.

11. Ensure that **Enable TCP/IP Networking** is checked, and check **Add firewall exception for this port**. Click on **Next**.

12. Select **Best support for multilingualism** and click on **Next**.

13. Leave the **Install as Windows Service** settings as the default, check **Include Bin Directory in Windows Path**, and click on **Next**.

14. Enter a password for the root user and remember to make a note of it. Enable the **Enable root access from remote machines** option.

15. Click on **Execute**, and the configuration will be deployed and the MySQL service started.

16. You may need to configure third-party firewalls to allow external access to the database (MySQL, by default, runs on `TCP/UDP3306`).

Configuring MySQL

Now that we have installed the MySQL service, we need to add a new user. These credentials will be used by XBMC instances to interact with the database.

Firstly, we need to bring up the MySQL command-line interface. In Linux and OS X, this can be done by typing `mysql` in the terminal/console. On Windows systems, go to **Start Menu** | **Programs** | **MySQL** and select **MySQL Command Line Client**. You will then need to enter the root password you set up during installation.

We can now create a new user called `xbmc` and grant them sufficient privileges, as follows:

```
CREATE USER 'xbmc' IDENTIFIED by 'xbmc';
GRANT ALL PRIVILEGES ON *.* TO 'xbmc'@'localhost' IDENTIFIED BY
'xbmc';
GRANT ALL PRIVILEGES ON *.* TO 'xbmc'@'%' IDENTIFIED BY 'xbmc';
FLUSH PRIVILEGES
```

> **Security Tip**
>
> The preceding commands create a user with full access to MySQL. Unfortunately this is necessary, as XBMC creates the database itself and changes the schema names as the database version matures. We can, however, make this more secure by using a strong password after the `IDENTIFIED BY` portion of our command.

While MySQL can be managed via the command-line interface (what we've used to create the XBMC user), management can be made much easier with MySQL Workbench. This is downloadable from the MySQL website at `http://www.mysql.com/products/workbench`.

Adding our SQL database to XBMC

By default, XBMC will use an internal SQLite-based database implementation. However, we would like to get XBMC to use our MySQL database instead. This is done by defining the database in `advancedsettings.xml`. We'll cover how to set up `advancedsettings.xml` for Raspbmc, but for other platforms, you should consult `http://wiki.xbmc.org/index.php?title=HOW-TO:Sync_multiple_libraries/Setting_up_XBMC`.

The best method for editing the settings file is via **Secure Shell** (**SSH**). SSH allows us to run commands on our Raspbmc installation as if we were typing them via a keyboard. The advantage of this is that we do not need to plug in a USB keyboard (which may not be available in spare) into the unit to edit the file.

1. On Linux and OS X systems, all we need is to open a Terminal and run the following command (substituting the IP address of the Raspberry Pi itself):

   ```
   ssh pi@192.168.1.6
   ```

2. As you are connecting to the database for the first time, you are likely to see a message similar to the following:

   ```
   sam@ubuntu:~$ ssh pi@192.168.1.6
   The authenticity of host '192.168.1.6 (192.168.1.6)' can't be established.
   ECDSA key fingerprint is 55:36:12:f1:df:de:06:1a:7a:a4:36:b4:e1:09:a0:e2.
   Are you sure you want to continue connecting (yes/no)?
   ```

3. You should type `yes` and then enter the password for the `pi` user. This password is `raspberry` by default, unless you have changed it in **Raspbmc Settings**.

4. On Windows systems, there is, by default, no SSH client installed, so we need to download one. PuTTY is a very capable client and can be downloaded from `http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe`.

5. The program is a standalone executable, so it can be run from wherever you saved it to. Enter the IP address of the device in the box entitled **Host Name**. The **Port** number can be left at the default value of **22**, as shown in the following screenshot:
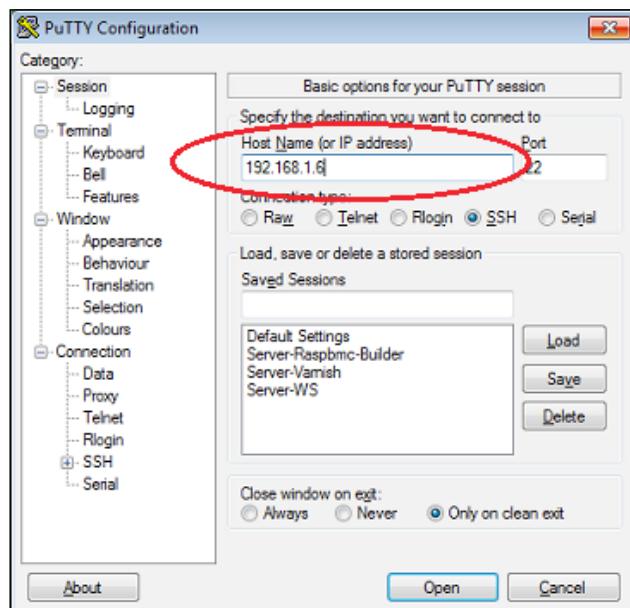
6. Click on **Open** to initialize a connection to the device.

7. You will be prompted for a username and password (the default username is `pi`, and the default password is `raspberry`).

8. As with OS X- and Linux-based connections, you will be prompted to verify the authenticity of the system. You should type `yes` when you see the following message:

```
The authenticity of host '192.168.1.6 (192.168.1.6)' can't be
established.
ECDSA key fingerprint is 55:36:12:f1:df:de:06:1a:7a:a4:36:b4:e1:09
:a0:e2.
Are you sure you want to continue connecting (yes/no)?
```

9. Now, whether we are using OS X, Linux, or Windows, we have successfully connected to our Pi. It's time to now edit the `advancedsettings.xml` file. We'll do this using the **nano** text editor. The following command will open the **nano** text editor:

```
nano .xbmc/userdata/advancedsettings.xml
```

We will now be presented with an interface similar to the one that follows:

We can type the contents of our file into this program. The file we are editing is an **XML** file (**Extensible Markup Language**), which means its layout and structure is very precise.

10. All settings are defined within the `<advancedsettings>` scope, so we will start with the following:

```
<advancedsettings>
</advancedsettings>
```

11. Now, to use a MySQL database for video content, we can add the following segment; it needs to be indented under the `advancedsettings` tags:

```
<videodatabase>
    <type>mysql</type>
    <host>192.168.1.2</host>
    <port>3306</port>
    <user>xbmc</user>
    <pass>xbmc</pass>
</videodatabase>
```

12. You should substitute the relevant details, such as the host (IP address of the MySQL server) and the password for the xbmc user, here, if you specified a custom password.

13. Likewise, if we want to add database support for audio as well, we would use the <audiodatabase> tags.

    Overall, we should end up with something akin to the following:
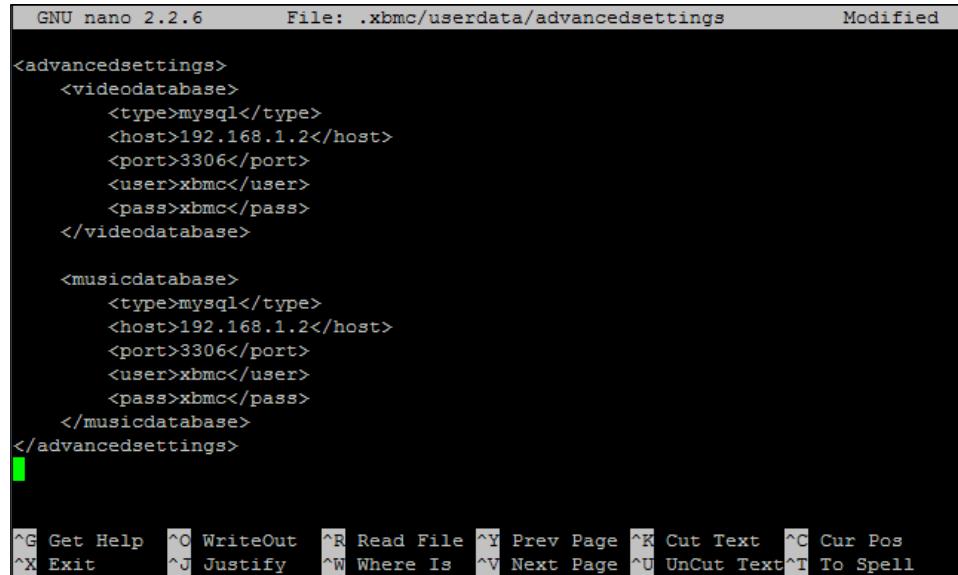
```
  GNU nano 2.2.6          File: .xbmc/userdata/advancedsettings          Modified

<advancedsettings>
    <videodatabase>
        <type>mysql</type>
        <host>192.168.1.2</host>
        <port>3306</port>
        <user>xbmc</user>
        <pass>xbmc</pass>
    </videodatabase>

    <musicdatabase>
        <type>mysql</type>
        <host>192.168.1.2</host>
        <port>3306</port>
        <user>xbmc</user>
        <pass>xbmc</pass>
    </musicdatabase>
</advancedsettings>


^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

14. Now, press *CTRL + O* to save the file. You will be prompted where to save the file. As we specified the file path as an argument when we invoked **nano**, we only need to press *Enter*.

15. You can now press *CTRL + X* to exit **nano**.

16. For these changes to take place, XBMC must be restarted; this can be done via the following command:

    sudo initctl xbmc restart

You can now exit the SSH session by typing exit.

Setting up the Library mode

The Library mode will automatically download FanArt and content information (such as episode descriptions and ratings), which can enhance the XBMC experience greatly. Libraries are *scraped* using a variety of content providers that provide data for your content. Let's learn how to set up the Library mode:

1. Go to **Videos** or **Music** depending on the kind of content you are trying to set.

2. Navigate to your files.

3. Select the source you wish to configure.

4. Bring up the context menu (we explored how to do this in *Chapter 3, Managing Content*).

5. Select **Set Content**. You will now be presented with the following dialog:
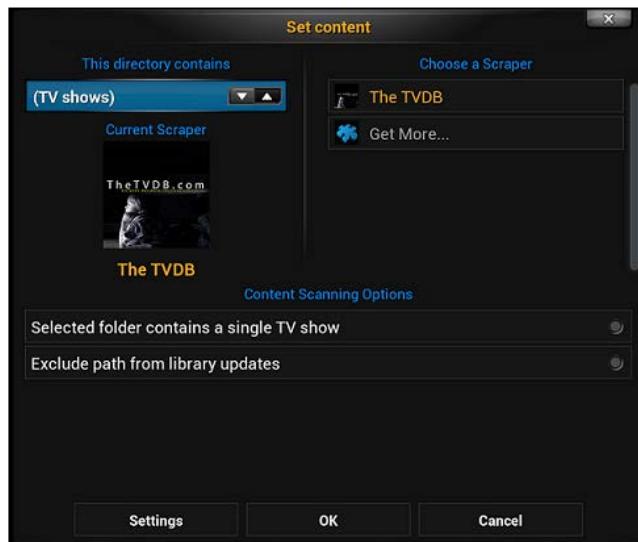
6. The **This directory contains** option allows you to set the type of content that the folder holds. Choose the type of content.

7. The **Choose a Scraper** option allows you to choose the content provider that will bring you detailed information and FanArt. Many more scrapers, such as IMDB, are available via the **Get More...** option.

8. Selecting **Settings** will bring up additional settings pertinent to the scraper you are using, as shown in the following screenshot:
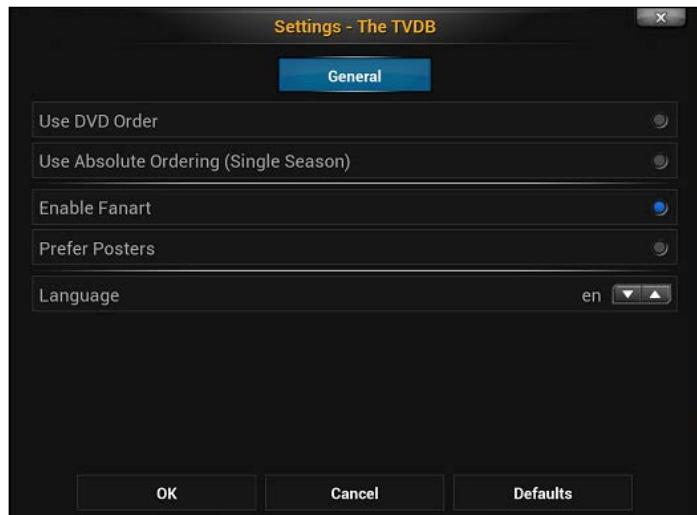
9. In the preceding example, we can select a variety of options, such as whether we prefer posters to banners, using the **The TvDB** scraper.

10. When you have configured the source, click on **OK**.

11. You will be prompted with **Do you want to refresh info for all items in this path?**; select **Yes**.

12. Your content will now be scraped into your Library. You will notice under **Videos** (if you set content on a video source) that a new option called **Library** is available as a submenu. Furthermore, as I added TV Shows to my library, a new option called **TV Shows** is now available, as shown in the following screenshot:

13. The **Library** option will now show the FanArt and descriptions (when available) for your content. These steps can be repeated for all sources you have added; however, scraping must be finished before another source can be configured for scraping.

Setting up the Library mode on a second system

All we need to do on a second or third system is add the same contents of advancedsettings.xml and define the same sources (they must be defined with the exact names and paths).

Summary

We've now covered how to set up a MySQL database, configure the Library mode, SSH into our Pi (see *Appendix A*, *Useful Raspbmc Commands*, for more Raspbmc commands that can be performed via SSH), and edit our advancedsettings.xml file.

In the next chapter, we'll look at PVR support, and how we can watch Live TV and schedule recordings using Raspbmc.

6

Using PVR with Raspbmc

In this chapter, we'll look at the following topics:

- What PVR allows us to do
- The different types of PVR setups that Raspbmc supports
- How to set up Raspbmc to take advantage of PVR

What is PVR?

Personal Video Recording (**PVR**), with a TV tuner, allows you to record as well as watch Live TV. Recordings can be scheduled manually based on a time, or with the help of the TV guide, which can be downloaded from the TV provider (by satellite/aerial or cable), or from a content information provider, such as *Radio Times* via the Internet. Not only does PVR allow you to watch Live TV, but on capable backends (we'll look at what a backend is in a moment), it allows you to rewind and pause live TV.

A single tuner allows you to tune into one channel at once, while two tuners would allow you to tune into two. As such, it's important to note that the capabilities listed earlier are not mutually exclusive, that is, with enough tuners it is possible to record one channel while watching another. This may, depending on the software you use as a backend, be possible on one tuner, if the two channels are on the same multiplexer.

Raspbmc's role in PVR

Raspbmc can function as both a PVR backend and a frontend. For PVR support in XBMC, it is necessary to have both a backend and one or more frontends. Let's see what a backend and frontend is:

- **Backend**: A backend is the part that tunes the channel, records your scheduled programs, and serves those channels and recorded television to the frontends. One backend can serve multiple frontends, if it is sufficiently powerful enough and there are enough tuners available.

- **Frontend**: A frontend is the part that receives content from the backend and plays back live television and recorded programs to the user. In the case of Raspbmc, XBMC serves as the frontend and allows us to play back the content. Multiple frontends can connect to one or more backends. This means that we can have several installations of Raspbmc play broadcast content from even a single tuner.

As we've now learned, Raspbmc has a built-in PVR frontend in the form of XBMC. However, it also has a built-in backend. This backend is TVHeadend, and we'll look at getting that up and running shortly.

Standalone backend versus built-in backend

There are cases when it is more favorable to use an external, or standalone, backend rather than the one that ships with Raspbmc itself. Outlined as follows is a comparison:

Standalone backend	Raspbmc backend (TVHeadend)
A better choice if you do not find TVHeadend feature-rich or prefer another backend.	If you only intend to have one frontend, it makes sense to run everything off the same device, rather than relying on an external system.
If you have a pre-existing backend, it is easier to configure Raspbmc to use that, rather than reconfiguring it completely.	The process may be simplified for you as, generally, one can just connect their device and it will be detected in TVHeadend.
If you are planning on having multiple frontends, it is more sensible to have a standalone backend. This is to ensure that the computer has enough horsepower, and also, you can serve from the same computer you are serving files from, and thus, only need one device on, rather than two (the streaming machine and the Pi).	Raspbmc's auto-update system covers the backend that is included as well. This means you will always have a reliable and stable version of TVHeadend bundled with Raspbmc and you need not worry about having to update it to get new features.

Standalone backend	Raspbmc backend (TVHeadend)
If you need to use a PCI or PCI express-based tuner, you will need to use an external backend due to limitations of the Pi's connectivity.	Better for wireless media centers. If you have low bandwidth throughput, then running the tuner locally on the Raspberry Pi makes more sense as it does not rely on any transfers between the network (unless using HDHomeRun).

Setting up PVR

We will now look at how to set up a PVR. This will include configuring the backend as well as getting it running in XBMC.

An external backend

The purpose of this title is to focus on the Raspberry Pi, and as there is a great variety of PVR software available, it would be implausible to cover the many options. If you are planning on using an external backend, it is recommended that you thoroughly search for information on the Internet. There are even books for popular and comprehensive PVR packages, such as MythTV. TVHeadend was chosen for Raspbmc because it is lightweight and easy to manage.

Raspbmc's XBMC build will support the following backends at the time of writing:

- MythTV
- TVHeadend
- ForTheRecord/Argus TV
- MediaPortal
- Njoy N7
- NextPVR
- VU+/Enigma
- DVBViewer
- VDR

Setting up TVHeadend in Raspbmc

It should be noted that not all TV tuners will work on your device. Due to the fact that the list changes frequently, it is not possible to list here the devices that work on the Raspberry Pi. However, the most popular tuners used with Raspbmc are AF90015 based. HDHomerun tuners by SilliconDust are supported as well (note that these tuners do not connect to your Pi directly, but are accessed through the network). With the right kernel modules, TVHeadend can support DVB-T (digital terrestrial), DVB-S (satellite), and DVB-C (cable) based tuners.

> For a list of supported tuners, see *Appendix C*, *Supported Peripherals*.

By default, the TVHeadend service is disabled in Raspbmc. We'll need to enable it as follows:

1. Go to **Raspbmc Settings**—we did this before by selecting it from the **Programs** menu.

2. Under the **System Configuration** tab, check the TVHeadend server radio button found under the **Service Management** category.

3. Click on **OK** to save your settings.

4. Now that the TVHeadend is running, we can now access its management page by going to `http://192.168.1.5:9981`.

> You should substitute the preceding IP address, `192.168.1.5`, with the actual IP address of the Raspberry Pi. We covered checking the Raspberry Pi's IP address in *Chapter 2*, *Using Raspbmc*.

You will be greeted with an interface much akin to the following screenshot:
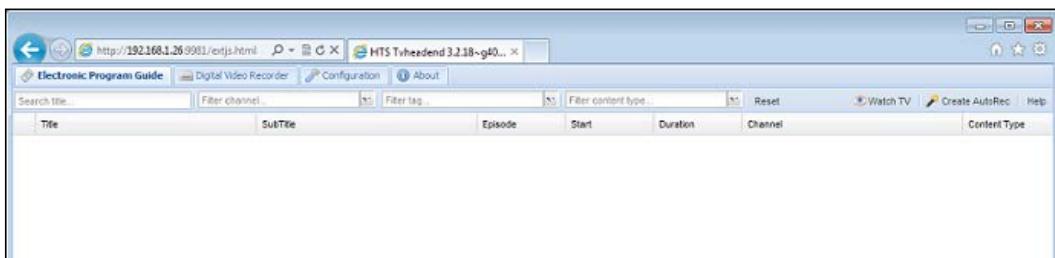
5. In the preceding screenshot we see that there are three main tabs available. They are as follows:

   ° **Electronic Program Guide**: This shows us what is being broadcast on each channel. It is empty in the preceding screenshot because we've not scanned and added any channels.

   ° **Digital Video Recorder**: This will allow you to schedule recordings of TV channels as well as use the *Automatic recorder* functionality, which can allow you to create powerful rules for automatic recording. You can also schedule recordings in XBMC; however, doing so via the web interface is probably more flexible.

   ° **Configuration**: This is where you can configure the EPG source, choose where recordings are saved, manage access to the backend, and manage tuners.

6. The **Electronic Program Guide** and **Digital Video Recorder** tabs are intuitive and simple so we will instead look at the **Configuration** section.

7. Our first step in configuring a tuner is to head over to **TV Adapters**:
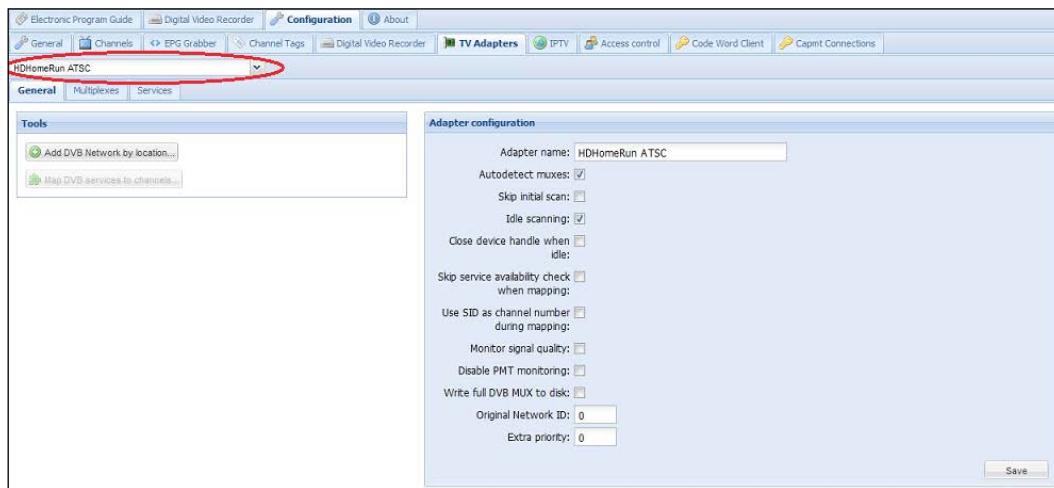
As shown in the preceding screenshot, TV tuners should automatically be detected and selectable in the drop-down menu (highlighted here). On the right, a box entitled **Adapter Configuration** can be used for adjusting the tuner's parameters.

8. Now, we need to select the **Add DVB Network by location** option. The following dialog box will appear:
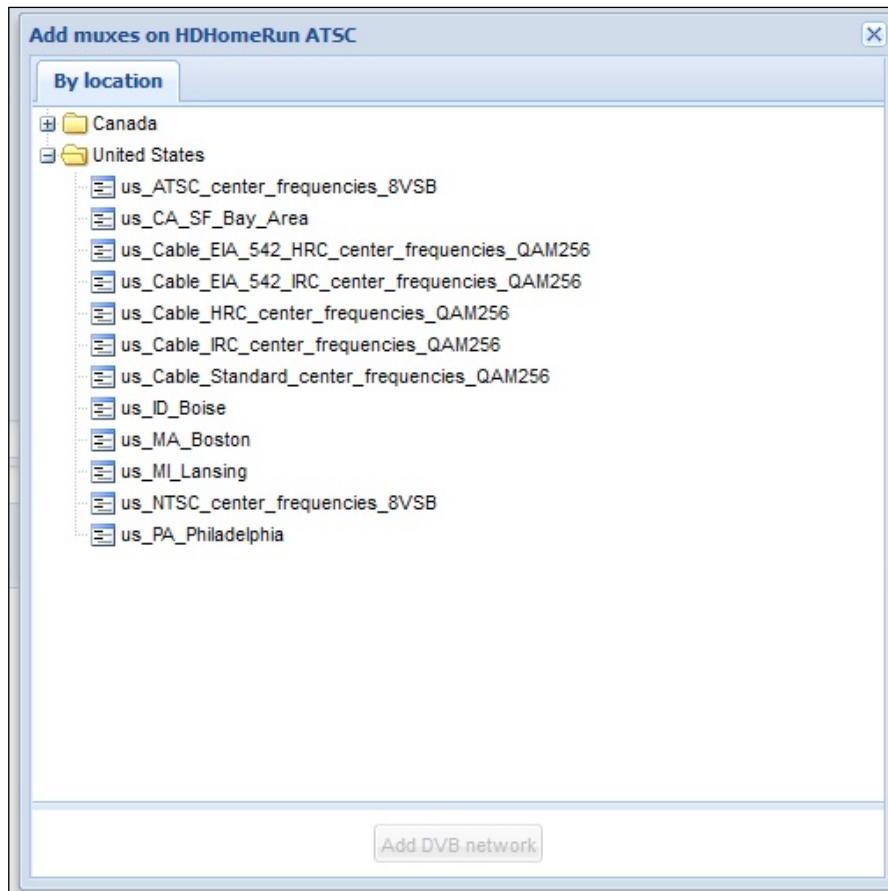
Once we have defined the region we are in, TVHeadend will automatically begin scanning for new *services* on the correct frequencies. These *services* can be mapped to channels by selecting the **Map DVB services to channels** button as shown earlier. We are now ready to connect to the backend in XBMC.

Connecting to our backend in XBMC

Regardless of whether we have used Raspbmc's built-in backend or an external one, the process for connecting to it in XBMC is very much the same.

We need to do the following:

1. In XBMC, go to **System | Settings | Add-ons | Disabled Add-ons | PVR clients**. You will now see the following screenshot:
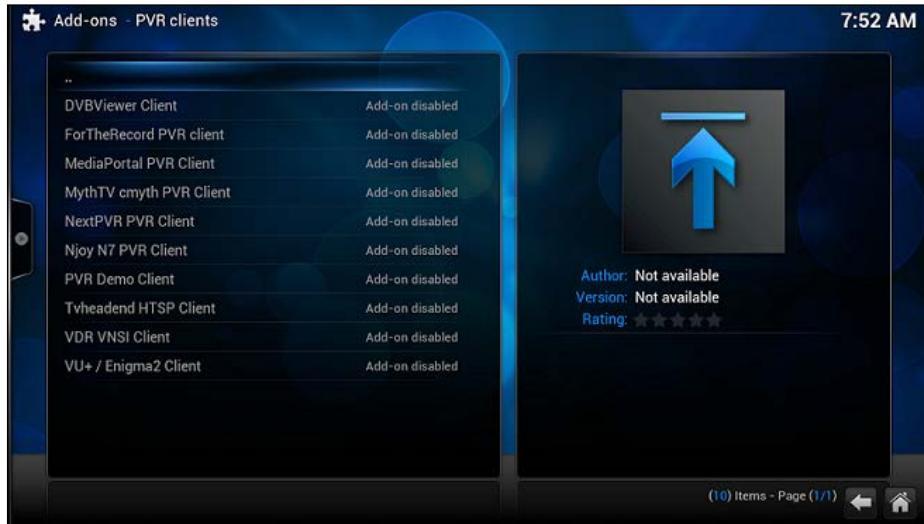
2. Select the type of backend that you would like to connect to. You will then see a dialog allowing you to configure or enable the add-on.

3. Select **Configure** and fill out the necessary connection details. Note that if you are connecting to the Raspbmc built-in backend, select the TVHeadend client to configure. The default settings will suffice:

4. Click on **OK** to save these settings and select **Enable**. Note that the add-on is now located in **System | Settings | Add-ons | Enabled add-ons | PVR clients rather than Disabled add-ons | PVR clients**.

5. Now, we need to go into **Settings | System | Live TV**. This allows you to configure a host of options related to Live TV. The most important one is the **Enable Live TV** option—be sure to check this box!
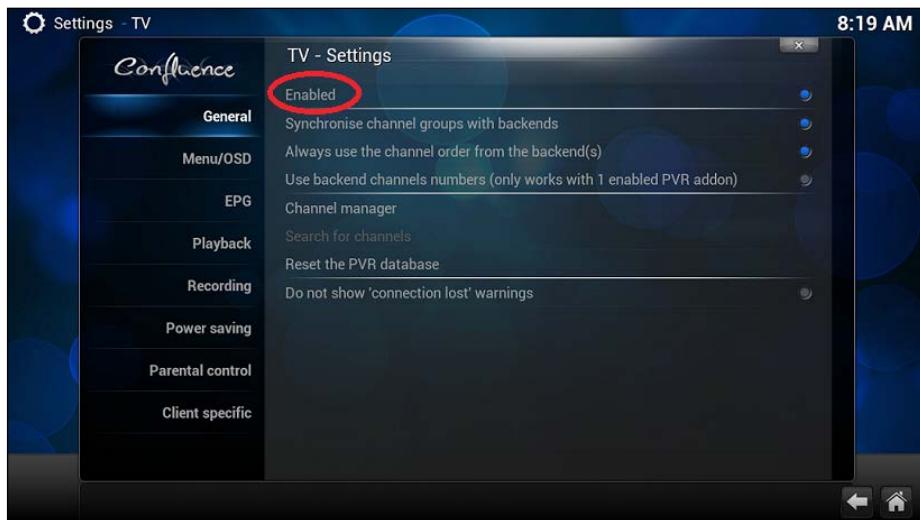
6. Now, if we go back to the main menu, we'll see a **Live TV** option. Your channel information will be there already, although, like the instance shown as follows, it may need a bit of renaming:
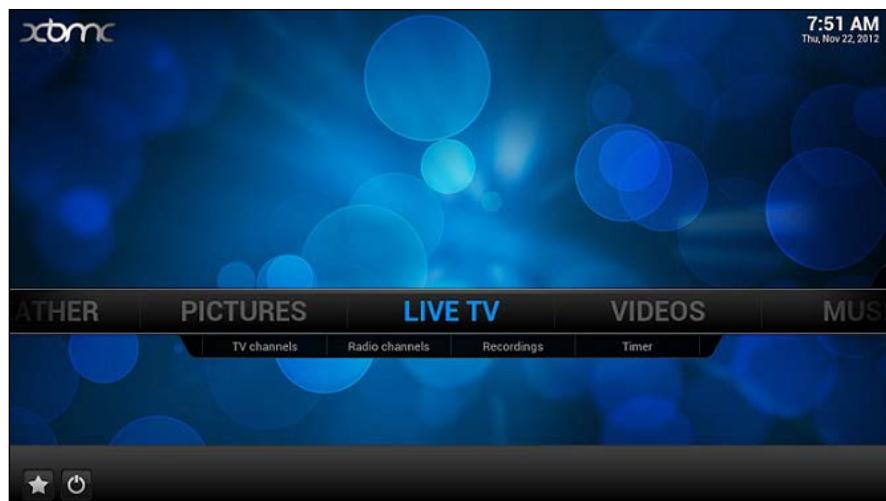
The following screenshot shows us a sample electronic program guide:

Simply select a channel and press **Play**! The functionality that PVR offers is controlled in a similar manner to XBMC, so this won't be covered in this chapter. If you have got this far, you've done the hard part already.

Summary

We've now covered what PVR can do for us, the differences between a frontend and backend, and where a remote backend may be more suitable than the one Raspbmc has built in. We then covered how to connect to that backend in XBMC and play back content from it.

In the next chapter, we'll look at what content Raspberry Pi can and cannot play, why this is the case, and converting video to suitable formats for playback in Raspbmc. We'll cover ripping from DVDs and Blu-ray discs as well.

7
Converting Media

In this chapter, we'll take a look at the following topics:

- Why we need to convert media
- A breakdown of the most popular codec types
- Converting media files on your computer with freely available software so they can be played in Raspbmc
- Ripping content from DVDs and Blu-ray Discs and converting them to a suitable media format for Raspbmc

Reasons for converting

Media may need to be converted for a variety of reasons. It may be that the current codec in use is inefficient. For example, the MPEG2 codec is at the same bitrate as H264, and approximately half as efficient. That is, an H264 video can be half the size of an MPEG2 video (saving storage) and still look visually equivalent.

Another reason for conversion is if the device does not support the codec that the content is currently encoded in. By converting the video to a format that the device does support, we can get playback support.

Codec breakdown

At present, there are several popular codecs that are in use. A comparison is outlined as follows:

Codec Name	Commonly found on	Efficiency (space requirements)	Supported by Raspbmc
H264	Blu-ray discs, HD-DVD discs, and licensed TV downloads	Best (joint with VC1)	Yes; supported out of the box till Profile L4.1 High
VC1	Blu-ray discs, HD-DVD discs, and licensed TV downloads	Best (joint with H264)	Yes; with an additional codec pack explained in *Chapter 2, Using Raspbmc*
MPEG2	DVDs and SD TV broadcasts	Average	Yes; with an additional codec pack explained in *Chapter 2, Using Raspbmc*

Converting media using Handbrake

Converting media is also sometimes known as **transcoding**. This is different from encoding, as encoding is considered to be the first conversion from the source format to the reproduced format (for example, from a film studio to Blu-ray Disc). All future conversions after this will thus be transcoded.

We'll use the popular and freely available Handbrake. This is because it works on Mac OS X, Windows, and Linux, and is capable of delivering some impressive results with minimal effort.

Handbrake can be downloaded from `http://handbrake.fr/downloads.php`. I will not cover its installation here as it is not only fairly simple but also platform-specific.

After launching Handbrake, we are presented with an interface similar to the one that follows:
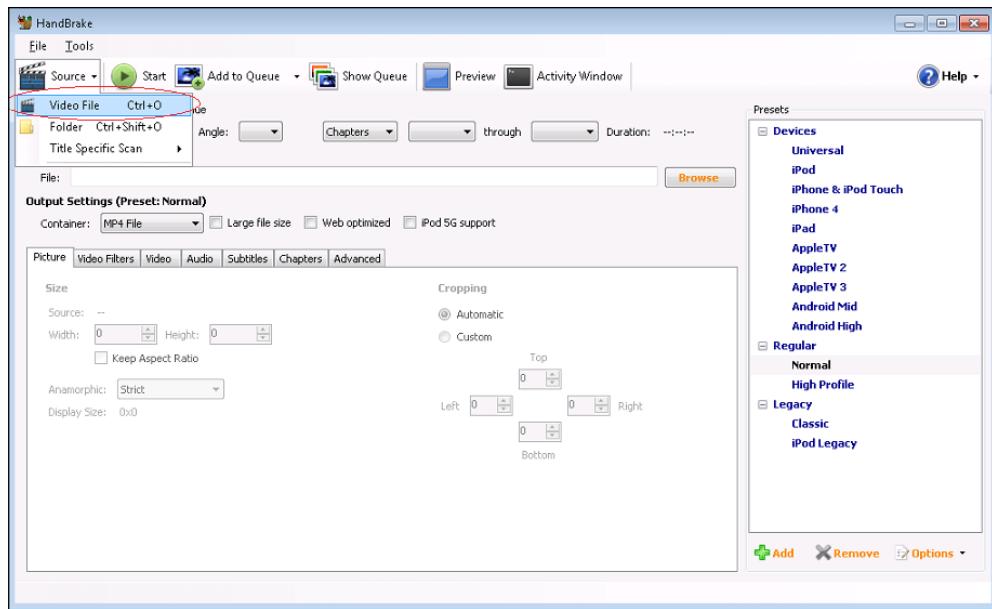
Let's see how to convert video files:

1. First, select the **Source** tab and choose **Video File**.
2. Find the video file you would like to convert and select it.

3. Now, we should set the destination filename, profile level, and **Container** type as shown in the following screenshot:
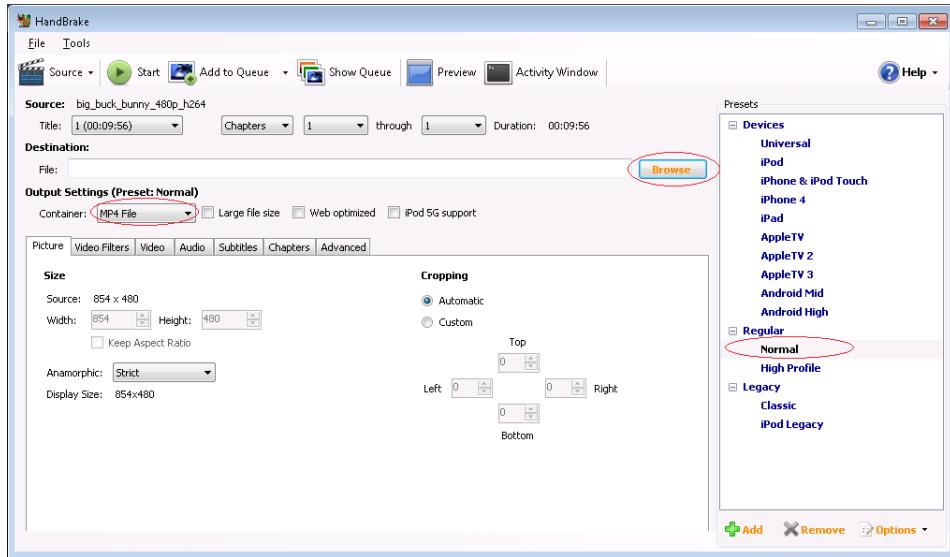
4. Select the **Container** type as **MKV File**. The Matroska container is a versatile container that allows the embedding of multiple streams.

5. Click on the **Browse** button and specify a filename. This is the name of the file that will be created.

6. Select the profile level as **Regular | Normal**. This will give you a decent level of quality taking into consideration the average amount of time required to convert the videos.

7. Now, we need to configure video-specific settings. This will allow us to set the desired subtitle streams and audio streams (if available). Furthermore, we will be able to set the level of quality that Handbrake outputs in the new video file.

The aspect ratio and cropping can be adjusted easily in the **Picture** tab shown as follows:

8. Handbrake provides standard stretching options under the **Anamorphic** setting.

9. Video filters can be applied as well. For example, if you are transcoding interlaced material, you should turn the **Deinterlace** filter on.
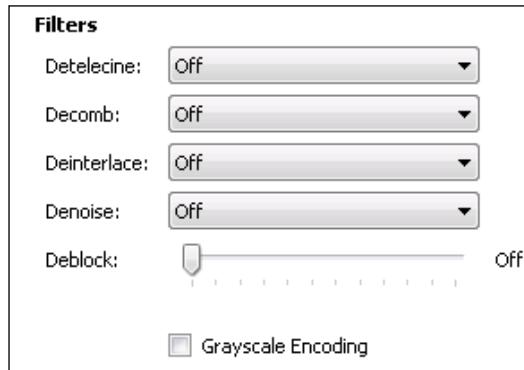
10. Under the **Video** tab, you can adjust the video encoding quality of the codec being used. I recommend you keep the video codec at **H264 (x264)**. The framerate can be adjusted, but this is generally only necessary if you are deinterlacing. In terms of quality options, the following are available:

    ° **Constant quality**: This will keep image quality constant regardless of the file size. **RF** stands for reference frames, and as a general rule of thumb, I would say 20 is acceptable for DVD and 18 is acceptable for high-definition content. Feel free to experiment with this in any way you like.

- **Average bitrate**: This will allow you to transcode at a constant bitrate. As such, you will be able to manage the file size produced by Handbrake. This is generally not as high a quality as constant, quality-based encoding. A two-pass encoding will distribute the bits used for encoding more efficiently as one pass is taken to analyze the video and the other is used to encode it. The downside is that this takes longer.
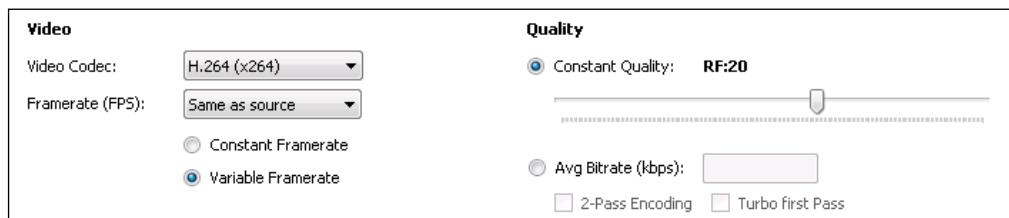
11. When adding audio streams to the new video, we can add or remove multiple streams if the original video has them.
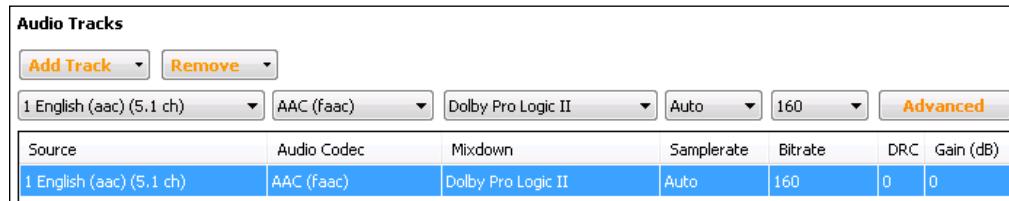
The table shown in the preceding screenshot is broken down as follows:

° **Source**: This is the original track that is embedded in the video.

° **Audio Codec**: This is the new track format that will be created.

° **Mixdown**: This is the number of channels that will be supported.

° **Samplerate**: This is the highest frequency of audio in the track that will be encoded. It is best to leave it as **Auto**.

- ° **Bitrate**: This is the number of kilobits per second that the audio stream is encoded with.
- ° **DRC**: This is dynamic range compression.
- ° **Gain** (in decibels): The volume can be changed using this option.

> It is possible to add multiple tracks even if they are originating from the same stream. One could simply select the desired source track and press **Add Track**. This is particularly useful if a device does not support a specific audio codec but if you wish to retain the one with a higher quality for devices that do.

12. The subtitles tab operates on the same principle where you can add or remove subtitle streams if they are in the original file, so we will not cover this in detail.

    Chapters, if not already available in the original file, can be imported from an external file.

13. As this is a simple introduction to the basics of video encoding, we will not go into detail about the advanced settings.

    Now that we are ready to start encoding, we can click on the **Start** icon, which is shown in the following screenshot:

14. The progress of your conversion will be shown at the bottom of Handbrake's screen:

When the conversion is complete, the file will appear in the location you specified. Provided you have used the default settings outlined in this chapter, it will now be ready for playback in Raspbmc. You can add this to one of your predefined sources in Raspbmc.

Ripping DVDs and Blu-ray Discs

The ripping of DVDs and Blu-ray Discs is heavily dependent on your platform. It should also be noted that you should only rip discs if it is legally permitted in your territory. Here is a list of the recommended packages:

- Mac OS X: MacTheRipper
- Linux: `libdvdcss2` is used at the command line
- Windows: AnyDVD (`http://www.slysoft.com/en/anydvd.html`) and MakeMKV (`http://www.makemkv.com`)

After ripping the disc, Handbrake can be used again to convert the files we have ripped from the disc. If Handbrake can find the `libdvdcss` library on Linux/Windows and OS X systems, it will be able to rip and convert discs directly.

Summary

At the end of this chapter, we've now touched on why we need to convert content, a breakdown of codecs that Raspbmc supports, and how to convert content using Handbrake.

Congratulations! You've made it to the end of this book. I hope you've learned how to turn your Raspberry Pi into an excellent media center with this aid and get the most out of this device.

A
Useful Raspbmc Commands

This will give you a short summary of useful commands that can be executed at the command line in Raspbmc either by an attached keyboard or via an SSH connection.

Shown as follows are some useful commands and a description of the task they achieve. It should be noted that this is not a comprehensive list of commands.

Description of command	Command to run on Raspbmc
Exit to a command line	Select the power icon in XBMC, select **Exit,** and press *Escape* on your keyboard
Start XBMC	`sudo initctl start xbmc`
Stop XBMC	`sudo initctl stop xbmc`
Reboot your Raspberry Pi	`sudo reboot`
Check your system log and view it in a scrollable manner	`less /var/log/syslog`
Upload several logs for diagnostics to Pastebin	`helpme`
Remove XBMC Nightly Build from startup and revert to the default XBMC build instead	rm /home/pi/.xbmc-current
Check free space available	df -h
Check how long Raspberry Pi has been on for	uptime
Delete a file	rm filename
Copy a file	cp filename newname
Move a file	mv /path/to/filename/path/to/ newfilename
Show the contents of a directory	ls -llh /some/folder (substitute)

Description of command	Command to run on Raspbmc
Execute a command with root privileges	sudo command
View CPU usage	top
View memory information	free -m
Install a package via apt	sudo apt-get update
	sudo apt-get install packagename (substitute packagename)
Remove a package previously installed via apt	sudo apt-get remove packagename
Edit a file	nano filename
Save a file after editing with nano	*CTRL + O* then press *Enter* to save
	CTRL + X to exit
Access XBMC userdata folder	cd /home/pi/.xbmc/userdata
Manually disable Raspbmc updates	touch /home/pi/.noupgrades
Manually enable Raspbmc updates	rm /home/pi/.noupgrades
Change the password	passwd
Change the hostname	hostname newname (substitute newname)
Disable Raspbmc updates via command line	touch /home/pi/.noupgrades
Force a Raspbmc kernel update	sudo rm /scripts/upd_hist/kver
Force an XBMC update	sudo rm /scripts/upd_hist/xbmcver

Unix/Linux command cheat sheet

The *Unix/Linux command cheat sheet* given in this topic has been made by Jacob Peddicord and is free to be used commercially. You can visit this page at `http://fosswire.com/post/2007/08/unixlinux-command-cheat-sheet/`.

Unix/Linux Command Reference

FOSSwire.com

File Commands

ls – directory listing
ls -al – formatted listing with hidden files
cd *dir* - change directory to *dir*
cd – change to home
pwd – show current directory
mkdir *dir* – create a directory *dir*
rm *file* – delete *file*
rm -r *dir* – delete directory *dir*
rm -f *file* – force remove *file*
rm -rf *dir* – force remove directory *dir* *
cp *file1 file2* – copy *file1* to *file2*
cp -r *dir1 dir2* – copy *dir1* to *dir2*; create *dir2* if it doesn't exist
mv *file1 file2* – rename or move *file1* to *file2* if *file2* is an existing directory, moves *file1* into directory *file2*
ln -s *file link* – create symbolic link *link* to *file*
touch *file* – create or update *file*
cat > file – places standard input into *file*
more *file* – output the contents of *file*
head *file* – output the first 10 lines of *file*
tail *file* – output the last 10 lines of *file*
tail -f *file* – output the contents of *file* as it grows, starting with the last 10 lines

Process Management

ps – display your currently active processes
top – display all running processes
kill *pid* – kill process id *pid*
killall *proc* – kill all processes named *proc* *
bg – lists stopped or background jobs; resume a stopped job in the background
fg – brings the most recent job to foreground
fg *n* – brings job *n* to the foreground

File Permissions

chmod *octal file* – change the permissions of *file* to *octal*, which can be found separately for user, group, and world by adding:
- 4 – read (r)
- 2 – write (w)
- 1 – execute (x)

Examples:
chmod 777 – read, write, execute for all
chmod 755 – rwx for owner, rx for group and world
For more options, see **man chmod**.

SSH

ssh *user@host* – connect to *host* as *user*
ssh -p *port user@host* – connect to *host* on port *port* as *user*
ssh-copy-id *user@host* – add your key to *host* for *user* to enable a keyed or passwordless login

Searching

grep *pattern files* – search for *pattern* in *files*
grep -r *pattern dir* – search recursively for *pattern* in *dir*
command | grep *pattern* – search for *pattern* in the output of *command*
locate *file* – find all instances of *file*

System Info

date – show the current date and time
cal – show this month's calendar
uptime – show current uptime
w – display who is online
whoami – who you are logged in as
finger *user* – display information about *user*
uname -a – show kernel information
cat /proc/cpuinfo – cpu information
cat /proc/meminfo – memory information
man *command* – show the manual for *command*
df – show disk usage
du – show directory space usage
free – show memory and swap usage
whereis *app* – show possible locations of *app*
which *app* – show which *app* will be run by default

Compression

tar cf *file.tar files* – create a tar named *file.tar* containing *files*
tar xf *file.tar* – extract the files from *file.tar*
tar czf *file.tar.gz files* – create a tar with Gzip compression
tar xzf *file.tar.gz* – extract a tar using Gzip
tar cjf *file.tar.bz2* – create a tar with Bzip2 compression
tar xjf *file.tar.bz2* – extract a tar using Bzip2
gzip *file* – compresses *file* and renames it to *file.gz*
gzip -d *file.gz* – decompresses *file.gz* back to *file*

Network

ping *host* – ping *host* and output results
whois *domain* – get whois information for *domain*
dig *domain* – get DNS information for *domain*
dig -x *host* – reverse lookup *host*
wget *file* – download *file*
wget -c *file* – continue a stopped download

Installation

Install from source:
./configure
make
make install
dpkg -i *pkg.deb* – install a package (Debian)
rpm -Uvh *pkg.rpm* – install a package (RPM)

Shortcuts

Ctrl+C – halts the current command
Ctrl+Z – stops the current command, resume with **fg** in the foreground or **bg** in the background
Ctrl+D – log out of current session, similar to **exit**
Ctrl+W – erases one word in the current line
Ctrl+U – erases the whole line
Ctrl+R – type to bring up a recent command
!! - repeats the last command
exit – log out of current session

* use with extreme caution.

B
Useful MySQL Commands

This appendix will give you a short summary of useful MySQL commands that can be executed at the command line on the system that is hosting the MySQL database.

Some useful commands are given in the following table. It should be noted that this is not a comprehensive list of commands.

Description of the command	Command to be run on Raspbmc
Connects to MySQL with the user as `root` and the password as `password`	`mysql -u root -ppassword`
	Remember, there must be a space after `-u` to specify the user, but the password must be attached to `-p`
Creates a user with a specific password	`CREATE USER 'sam'@'localhost' IDENTIFIED BY 'password';`
Creates a database	`CREATE DATABASE somedb;`
Grants a user full privileges to access all databases	`GRANT ALL PRIVILEGES ON *.* TO 'sam'@'%';`
Grants a user full privileges to access a specific database	`GRANT ALL PRIVILEGES ON somedb.* TO 'sam@localhost';`
Deletes a MySQL database	`DROP DATABASE dbname;`
Makes a backup of a database	`mysqldump  -u root -ppassword somedb > backup.sql`
Restores a backup of a database	`mysql -u root -ppassword somedb < backup.sql`

Note that `somedb`, `sam`, and `dbname`, used in preceding table, are just examples and should be substituted accordingly to match your configuration.

More information about MySQL commands can be obtained from `http://dev.mysql.com/doc/refman/5.0/en/mysql-commands.html`.

C
Supported Peripherals

This appendix will give you a short summary of supported peripherals in Raspbmc.

Supported remote controls

Raspbmc supports a variety of remote controls, including:

- Web-based remotes, including smartphone applications
- TV remote
- XBOX DVD playback remote
- ATI/X10 remote
- CyberLink 0766:0204
- MCE remote
- Hama remote

> For a full list of supported remotes, please consult the Wiki at `http://www.raspbmc.com/wiki/user/configuring-remotes/`.

Supported Wi-Fi adapters

Raspbmc supports several out-of-the-box Wi-Fi adapters; they are as follows:

- Realtek 8188CUS/8192CU chipset-based adapters
- Realtek 8712U chipset-based adapters
- Realtek 2870 chipset based adapters

Supported TV tuners

Raspbmc's TVHeadend implementation ships with TV tuner support for the following devices:

- HDHomerun
- AF9015 DVB-T chipsets
- AF9013 DVB-T chipsets
- AF9035 DVB-T chipsets
- AZ6027 DVB chipsets
- Nova-T DVB
- Hauppage HVR930C
- SkyStar USB devices
- Terratec USB tuner

Supported SD cards

- Sandisk SD cards (Ultra Series recommended)
- Samsung SD cards
- Integral SD cards (UltimaPro series recommended)
- Transcend SD cards

Class 10 SD cards are recommended for ultimate performance. A USB installation can also improve performance.

> At the time of writing, Kingston SD cards are known to be problematic with the Raspberry Pi firmware and should be avoided.

Supported power supplies

In theory, a power supply with a specification of 5V 1A is more than ample for Raspbmc; however, this is often not the case. This is because the specifications printed on these supplies are unregulated, and are, in fact, printed and determined by the manufacturer. The best method of determining if a power supply is able to deliver 5 volts to the Raspberry Pi while under load (that is, playing a video) is by placing a multimeter between the points marked TP1-TP2 on the Raspberry Pi (shown in the following figure). By doing this, you can determine if the power supply you are using is delivering sufficient power.
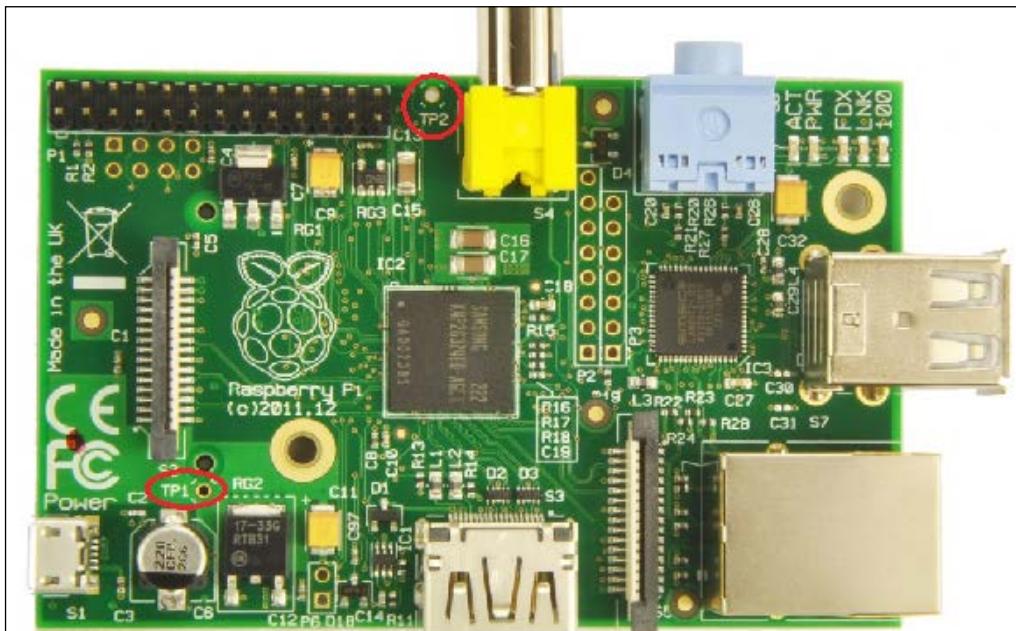
A suitable reading under load is ~4.8V. However, this will vary on each device depending on the resistance of the polyfuse F3.

Index

[PACKT] PUBLISHING open source*
community experience distilled

Thank you for buying
Raspberry Pi Media Center

About Packt Publishing

Packt, pronounced 'packed', published its first book "*Mastering phpMyAdmin for Effective MySQL Management*" in April 2004 and subsequently continued to specialize in publishing highly focused books on specific technologies and solutions.

Our books and publications share the experiences of your fellow IT professionals in adapting and customizing today's systems, applications, and frameworks. Our solution based books give you the knowledge and power to customize the software and technologies you're using to get the job done. Packt books are more specific and less general than the IT books you have seen in the past. Our unique business model allows us to bring you more focused information, giving you more of what you need to know, and less of what you don't.

Packt is a modern, yet unique publishing company, which focuses on producing quality, cutting-edge books for communities of developers, administrators, and newbies alike. For more information, please visit our website: `www.packtpub.com`.

About Packt Open Source

In 2010, Packt launched two new brands, Packt Open Source and Packt Enterprise, in order to continue its focus on specialization. This book is part of the Packt Open Source brand, home to books published on software built around Open Source licences, and offering information to anybody from advanced developers to budding web designers. The Open Source brand also runs Packt's Open Source Royalty Scheme, by which Packt gives a royalty to each Open Source project about whose software a book is sold.

Writing for Packt

We welcome all inquiries from people who are interested in authoring. Book proposals should be sent to author@packtpub.com. If your book idea is still at an early stage and you would like to discuss it first before writing a formal book proposal, contact us; one of our commissioning editors will get in touch with you.

We're not just looking for published authors; if you have strong technical skills but no writing experience, our experienced editors can help you develop a writing career, or simply get some additional reward for your expertise.
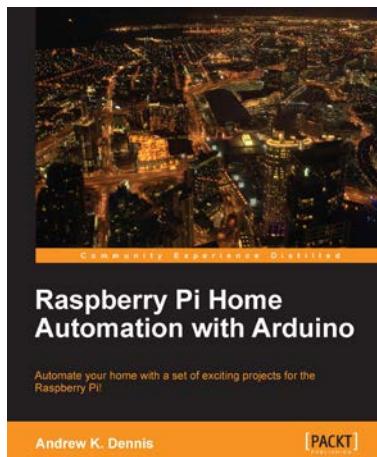
Raspberry Pi: Home Automation

ISBN: 978-1-849695-86-2   Paperback: 176 pages

Automate your home with a set of exciting projects for the Raspberry Pi!

1. Learn how to dynamically adjust your living environment with detailed step-by-step examples

2. Discover how you can utilize the combined power of the Raspberry Pi and Arduino for your own projects

3. Revolutionize the way you interact with your home on a daily basis

Linux Shell Scripting Cookbook

ISBN: 978-1-849513-76-0   Paperback: 360 pages

Solve real-world shell scripting problems with over 110 simple but incredibly effective recipes

1. Master the art of crafting one-liner command sequence to perform tasks such as text processing, digging data from files, and lot more

2. Practical problem solving techniques adherent to the latest Linux platform

3. Packed with easy-to-follow examples to exercise all the features of the Linux shell scripting language

4. Part of Packt's Cookbook series: Each recipe is a carefully organized sequence of instructions to complete the task as efficiently as possible

Please check **www.PacktPub.com** for information on our titles
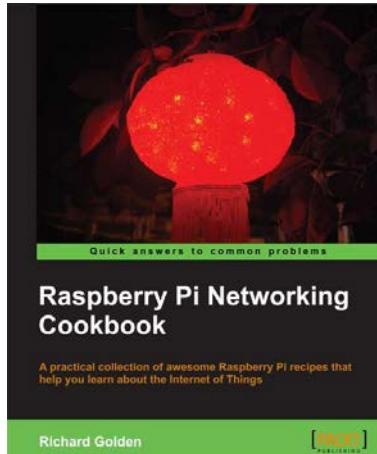
Raspberry Pi Networking Cookbook

ISBN: 978-1-849694-60-5   Paperback: 185 pages

A practical collection of awesome Raspberry Pi recipes that help you learn about the Internet of Things

1. Learn how to install, administer, and maintain your Raspberry Pi

2. Create a network fileserver for sharing documents, music, and videos

3. Host a web portal, collaboration wiki, or even your own wireless access point

4. Connect to your desktop remotely, with minimum hassle
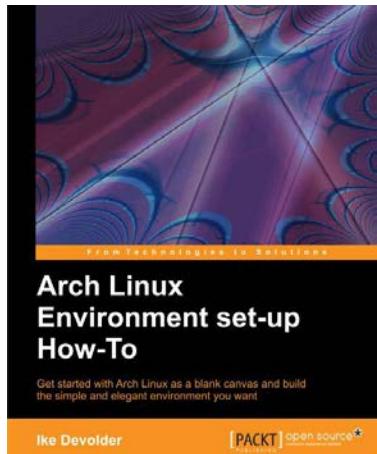
Arch Linux Environment Setup How-to

ISBN: 978-184951-9-72-4   Paperback: 68 pages

Get started with Arch Linux as a blank canvas and build the simple and elegant environment you want

1. Learn something new in an Instant! A short, fast, focused guide delivering immediate results

2. Install and configure Arch Linux to set up your optimum environment for building applications

3. Boot and manage services, add and remove packages

Please check **www.PacktPub.com** for information on our titles