Web Generated PDF Document

This document was generated automatically. The original (web) copy of this material can be found at **everwicked.com**.

This document is copyrighted and should not be changed without the permission of the author. Feel free to distribute this PDF document as it is without any further notifications.

Updates to this document are released frequently in order to be up to date with the latest software releases. For the latest version of this document, consult **everwicked.com**.

Also feel free to send comments and/or suggestions regarding this document via our **web forums**.

The Author

Table of Contents

1 Introduction – What is MPEG4IP

This guide's name is not very attracting and kind of cryptographic but it is a very useful piece of documentation as you are about to find out.

MPEG4IP Project

The MPEG4IP project was recently created to provide an open–source standards–based system for encoding, streaming, playing and even broadcasting MPEG–4 encoded audio and video. The project integrates numerous open source applications along with new made code to complete the bundle.

The project's README file states that there legal issues when the package is used by end–users but we are going to refer to this later on.

The main platform supported by MPEG4IP is Linux but since the source code is available, anyone with a C/C++ compiler can spare a few CPU cycles and create binaries for any other platform. Windows users will lack full support of the features but there is still a lot you can do, read on!

MPEG4IP has been compiled and tested on: Linux, FreeBSD, BSD/OS, Solaris, Mac OS X and Windows. For *NIX platforms, X11 is required and for Windows, DirectX 8.1.

MPEG4IP's official web site is located at mpeg4ip.net and the development takes place at SourceForge.

This guide is based on MPEG4IP version 0.9.8.13. The code was retrieved from the CVS repository and compiled on Linux and Windows by myself. More information on this later on. The guide will be updated when there are actual changes in the process as the project matures.

2 Building and installing MPEG4IP

First of all we need to download and compile/install MPEG4IP. I will provide instructions for Linux and Windows, to compile MPEG4IP on some other platform the procedure is very similar (detailed instructions in the README file).

If you face any difficulties on other Operating Systems, you are encouraged to contact the developers and share you experience at the Sourceforge forums.

Linux

Required packages (GNU)

1. Autoconf
2. Automake
3. Libtool (later than 1.3.5)
4. X11 with Development Libraries

The above software can be obtained from any GNU mirror although most decent and updated Linux distributions have them preinstalled or provide binaries, so check your distribution's binary repository before downloading the source.

If you have the above tools installed you now need to download the latest CVS source code (0.9.3.8). I do not recommend using nor will support 0.9.3 release of MPEG4IP.

You will need the CVS utility for this. Yet again consult your distribution's package repository to get a copy. As far as I know, linux distributions that do not provide binaries to the CVS utility do not exist.

Let's login to the CVS server and fetch the source.
everwicked@daemon:~$ cvs −d:pserver:anonymous@cvs.mpeg4ip.sourceforge.net:/cvsroot/mpeg4ip login
Tap enter when asked for password

everwicked@daemon:~$ cvs −z9 −d:pserver:anonymous@cvs.mpeg4ip.sourceforge.net:/cvsroot/mpeg4ip co mpeg4ip

Assuming that the parent directory if your home directory, then mpeg4ip/ will be created and the source will be placed there. I hope you have a fast internet connection because even with maximum compression this is going to take a while on slow speeds.

When the downloading is done, change dir and start the compiling.
everwicked@daemon:~$ cd mpeg4ip/
everwicked@daemon:~$ ./bootstrap *Generation of the nessecary scripts*
everwicked@daemon:~$ make *Compilation of the source*
everwicked@daemon:~$ su root *Root privileges required to install the binaries*
root:/home/everwicked# make install *Installation of the binaries under /usr/local/bin.*

Hopefully, you had no errors while compiling and installing but if you did, don't hesitate to read them and install what's missing.

You may also like to compile only the player by passing the "−−disable−server" parameter to bootstrap or compile everything but the player by passing "−−disable−player" to bootstrap.

Windows

You will need WinCVS to download the CVS code and also Microsoft Visual C++ with the ATL libraries to compile MPEG4IP. I had Service Pack 5 installed when I did the compilation but it should work with earlier versions as well. WinCVS can be downloaded here or here.

Launch WinCVS and press CTRL+F1. It will launch the admin preferences.
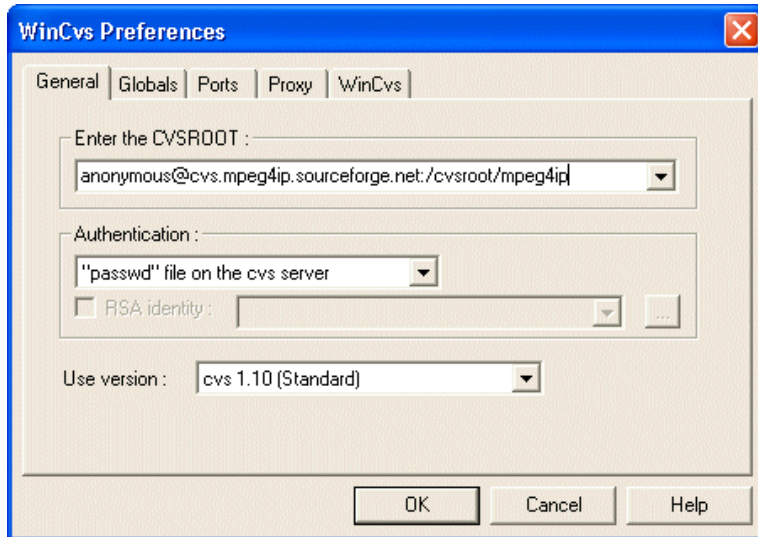
In the CVSROOT field type: anonymous@cvs.mpeg4ip.sourceforge.net:/cvsroot/mpeg4ip and select "passwd" file on the cvs server as the Authentication type.
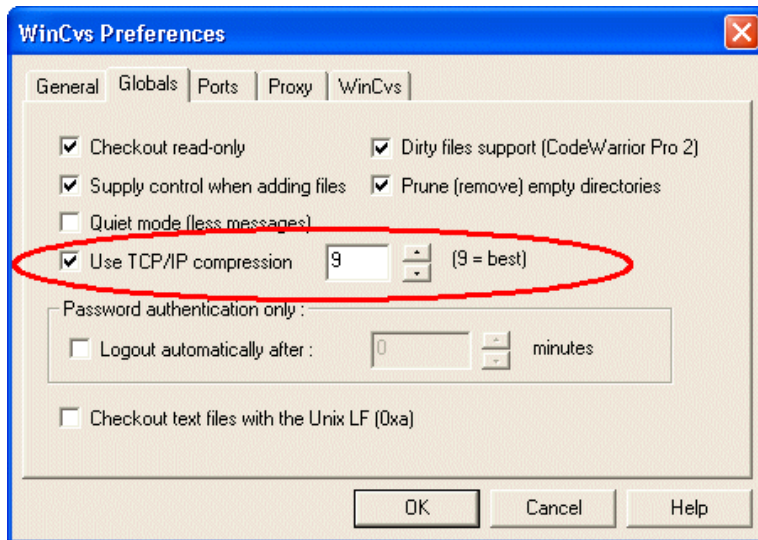
Now switch to the Globals tab and make sure the TCP/IP compression is set to 9 as shown on the left to minimise download time. Press OK.

Next, select Login through the Admin menu. When WinCVS asks for a password, just press enter.

Now select Checkout Module via the Create menu on WinCVS. Enter mpeg4ip as the module name and select a directory to save the source. The program will automatically create an mpeg4ip subdirectory. Press OK and the download will start.
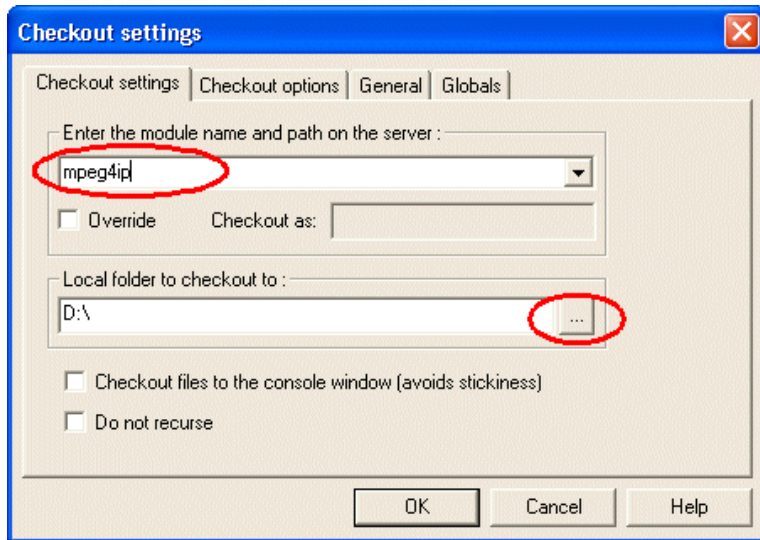
When the download is finished, you need to compile the source. As mentioned above, you will need Microsoft Visual C++ 6 or later.

Assuming you have the compiler installed, launch it. Through the File menu, select Open Workspace and browse to the directory you saved the MPEG4IP CVS. Select and open the encoding60.dsw workspace.
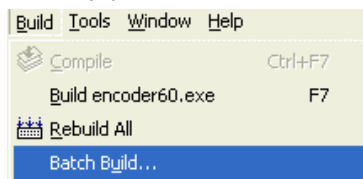
Select Batch Build from the Build menu.

Now we need to check the correct packages to build.
You need to select the following *Win32 Release* packages:
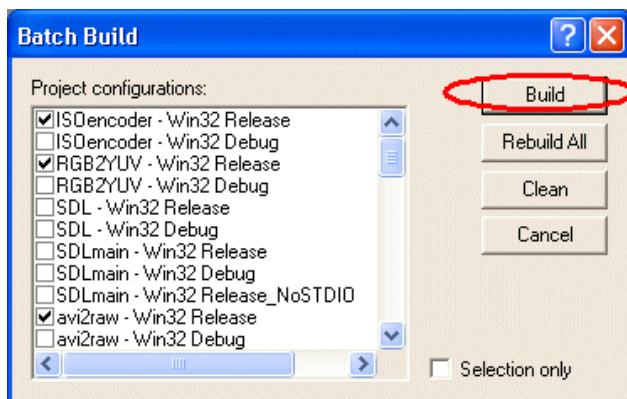
- ISOencoder
- RGB2YUV
- avi2raw
- avidump60
- faac
- lame60
- mp4creator
- mp4dump
- mp4extract
- yuvdump
- xvidenc

Click Build and let VC++ alone to calculate dependencies and compile the required libraries.

This was for the encoding applications. In the same way, we need to compile the player and its GUI.

Open the player60.dsw workspace that is located in player\src\ taking as reference the mpeg4ip CVS directory. If you get a warning about wav_plugin just ignore it.

Do another Batch Build and this time select the following (Win32 Release) projects:

- aac_plugin
- mp3_plugin
- mp4player
- mpeg4_iso_plugin
- wmp4client60
- wmp4player60
- xvid_plugin

Click Build and let the compile to take place.

You should by now have the following files in your possesion:

- avi2raw60.exe
- avidump60.exe
- divxenc60.exe
- encoder60.exe
- faac60.exe
- lame60.exe
- mp4creator60.exe
- mp4dump60.exe
- mp4extract60.exe
- RGB2YUV60.EXE
- yuvdump60.exe
- aac_plugin.dll
- mp3_plugin.dll
- mp4player60.exe
- mpeg4_iso_plugin.dll
- SDL.dll
- wmp4client.exe
- wmp4player.exe
- xvid_plugin.dll

To install the utilities, simply copy them to a directory in your Program Files folder and then add that directory to your path. For Win9x/ME the PATH can be edited via the sysedit command and for Win2k/XP just edit the Environment Variables from My Computer properties. For example your PATH should look like this:
C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\Program Files\mpeg4ip

You can also create a shortcut on your Desktop for wmp4player.exe, it's the GUI player so it doesn't have to be provoked from the command line.

If you are a bit lost, don't worry the hard part was just completed.

3 Creating MPEG−4 content

At the moment, MPEG4IP offers two ways to create content.

The first method assumes that you have somehow captured raw audio and/or video to a file. To clarify, that is not .avi or something similar. Raw video would be just YUV12 video without any file headers. Raw audio would be PCM.

The other method is to capture and encode video on the fly with mp4live. Windows users: Sorry guys this is supported by Linux only. This is quite exciting though since it offers you the chance to broadcast live audio/video streams over the Internet!

For that you will need a video capture device that is supported by the Video for Linux drivers. So far, MPEG4IP has been tested with the bttv driver for Brooktree based video capture cards, and the qce driver for Logitech QuickCam Express USB webcams. Therefore for about $50 USD you can stream from your camera over the Internet. Well, not exactly, there's legal information coming up, keep on reading.

Once we get our content ready we can stream it with the Apple Darwin Streaming server. As a sidenote, any server that understands MP4 files can be used. Whilst other servers can work with MPEG4IP, the Apple team has been working closely with MPEG4IP to ensure compatibility to the standards and with each other.

This is what MPEG4IP has to offer for encoding. However there are a few other ways to create content with our preferred technologies and methods. You are not far away from your first MP4 file with audio!

4 MPEG−4 Video

This is the most varied part.

If you are using Windows and you want to use MPEG4IP to stream your backup to your kids' room for fun, then check the backup guides.

MPEG4IP offers two solutions for encoding raw video/audio to MPEG−4. It comes bundled with an ISO MPEG−4 encoder and XviD. The ISO MPEG−4 encoder is quite slow and XviD rivals its quality.

For both Windows and Linux platforms, if your source is a DVD, there are other utilities than can encode the video for you. For Linux there is the excellent transcode tool, for Windows, you can check the DVD transcoding guides.

To encode with the MPEG4IP utilities, you will need an AVI file with audio being PCM and video format YUV12, aka YV12. Any frame size and frame rate will work. But you don't want the framesize to be big except if you appreciate fat hard disks.

Linux/Unix

The *NIX installation comes with mp4encode which makes our life easier. It will convert an AVI file directly to mp4.

Usage: mp4encode [options] filename.avi [filename.mp4]

Flags:
−w X, where X is the input video frame width in pixels, default value is 320
−h X, where X is the input video frame height in pixels, default value is 240
−r X, where X is the video frame rate in frames per second, default value is 24
−V X, X being the desired video bitrate in Kbps, default value is 500
−A X, X being the desired audio bitrate in Kbps, default value is 96
−R, when the input is RGB24 and conversion to YUV12 is needed
−a X, with X the desired aspect ratio, default value is 1.33 (4:3). Typically letterbox is 2.35.
−I will use the ISO MPEG−4 video encoder instead of the OpenDivX encoder
−M will use the MP3 audio encoder instead of the AAC encoder
−d Debug mode, leave intermediate files.

Therefore the example command:
mp4encode −w 640 −h 480 −r 29.97 −I −V 1000 −A 128 MyMedia.avi MyMedia_ISOMPEG4_1000_AAC_128.mp4
will encode MyMedia.avi with the output being 640x480, 29.97FPS, ISO−MPEG4 Video at 1000kbps and AAC audio at 128kbps.

Note: There is also xvidenc which will encode raw video with XviD. It will accept the −w, −h, −r and −b. Additionally there is −i which allows you to set the Keyframe Interval. e.g. a value of 30 will make the encoder place at least once Keyframe every 30 frames.

Windows

In Windows you can use xvidenc. However there are numerous other utilities that let you encode most video types and take full advantage of the encoder.

5 MPEG–4 Audio

Now that you have your video encoded you need to encode the audio. The MPEG–4 file format supports either AAC (MPEG–4 Audio) or the more familiar MP3 (MPEG–1 Layer 3 Audio). MPEG4IP is bundled with the LAME MP3 encoder but you can encode your audio with any other utility and therefore I will not go into detail.

Let's have a look at the AAC encoders though. We have two choices. That is either FAAC (Linux/Windows) and PsyTEL (Windows only). I have not attempted to compile FAAC on Windows but you should be able to work with mingw32 and the Makefile bundled. I haven't compared the quality between the two encoders either. Whilst the PsyTEL encoder is known to beat versions of FAAC 1.5 and earlier in terms of quality output, I have yet to see any quality tests on FAAC 2 (latest CVS) which is meant to be far better.

FAAC

FAAC accepts WAV input just like most encoders. A reference of options follows.

Option	Use	Valid arguments
–m X	Where X is the AAC MPEG version	X can be 2 or 4
–o X	Selects the AAC object type	X can be: LC, MAIN or LTP
–n	Doesn't use mid/side coding	
–t	Use TNS coding	
–c X	Set the bandwidth	X is the frequency in Hz
–b X	Set the bitrate per channel	X is the bitrate in bps

Therefore the command faac –o LTP –m 4 –b 128 input.wav output.aac will create encode input.wav to output.aac using the LTP MPEG–4 profile and a bitrate of 128kbps.
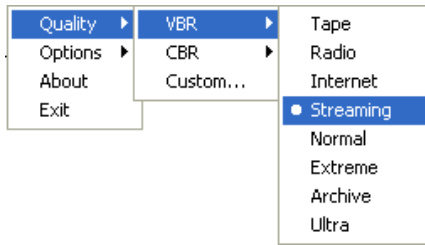
PsyTEL AAC Encoder

The PsyTEL encoder is a command line encoder just like FAAC. However, there exists a GUI named PsyTELDrop that we are going to use.

Launch PsyTELDrop (must be in the same directory with aacenc.exe).
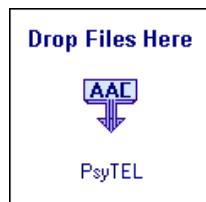
The window shown in the screenshot on the left will appear.

To configure the encoder, right click on it and select of one the VBR (preferably) presets. Extended info on each preset follows.

Note that the PsyTEL encoder supports WAV files of 1 to 6 channels making possible a 5.1 setup, although the resulting bitrate might be very close to that of AC3. AACenc will preserve the mapping of the WAV file.

PsyTEL AACenc has quite a few predefined VBR presets. Namely, Tape, Radio, Internet, Streaming, Normal, Extreme, Archive and Ultra. For music, the Normal preset should be enough whereas for movies (speech) the Streaming preset (averages at about 130kbps) is quite good. Feel free to do your own tests though.

Once you selected the preset of your liking, you can Drag'n'Drop the WAV file in the small window and the encoding will start in a DOS window.

If you are interested in the CBR capabilities of the PsyTEL encoder, the PDF bundled with it has more information.

6 Multiplexing

MPEG4IP offers a little nice utility that can add media tracks, add "hint" tracks, delete a track and optimize the layout of an existing MP4 file or create one from scratch.

There no GUIs designed for any of the MPEG4IP utilities yet and therefore you will need to use the command line.

Currently the mp4creator can read MP3, AAC (ADTS), and MPEG−4 video files. There is also a patch that allows to mux Vorbis in an MP4 file.

Note: MP3 ID3 or ID3V2 information is not carried in the file.

Creating an MP4 file from AVI

Assuming that your video is already DivX/XviD/ISO MPEG−4 encoded you can make an mp4 file from an AVI file using the following command sequence:
mp4creator −rate=25 −c=MyVideo.avi MyMP4File.mp4
And to add audio: mp4creator −c=MyAudio.xxx MyMP4File.mp4

Note that the audio extension can be either MP3 or AAC as mentioned above. The −rate flag takes the framerate of the video track and can be any integer or floating point number like 29.970.

Creating an MP4 file from a DivX 5 .divx or .mp4 file

DivX 5 gave us the MP4 output feature which will produce the video stream in a MPEG−4 File Format wrapper. To add audio, just use the command:
mp4creator −rate=25 −c=MyAudio.xxx DivX5file.divx

Note that .divx and .mp4 is the exact same thing.

Preparing a media track for streaming

In order to stream the file later, you will need to optimize and "hint" the file. But let's start from the beginning.

Yet again, assuming you have a ready AVI file encoded with DivX/XviD/ISO MPEG−4 you can hint the track on the fly:
mp4creator −rate=25 −hint −c=MyVideo.avi MyHintedMP4File.mp4
And to add audio: mp4creator −hint −interleave −c=MyAudio.aac MyHintedMP4File.mp4
The −interleave option uses an interleaved RTP payload format for the audio hinted track but only supports AAC. Therefore if you are using MP3, ignore that flag.

Then to optimize the file: mp4creator −optimize MyHintedMP4File.mp4
This feature will optimize the layout of the mp4 file to interleave the data in time, hence reducing disk seeks during streaming or playback. This also arranges the mp4 file so that the media control information is at the beginning of the file and allows HTTP streaming of the mp4 file.

DivX 5 has also the Data Partitioning feature that will implement error recovery on the stream.

Additional MP4 manipulation

mp4creator −list MyMP4File.mp4 will print a list of tracks in the file.

If you wish to delete or hint a track later you may:

mp4creator –delete=Id MyMP4File.mp4 or mp4creator –hint=Id MyMP4File.mp4

Id is determined from the list flag above.

For debugging purposes or if you are addicted to it, you may use the –verbosity=X flag to display more details:

1. Concerning reading and writing
2. Control tables
3. Media samples
4. Media hints
5. Everything

I like verbosity level 5 – wonder why..

Please note you don't need to have a video track. You can stream MP3 or AAC files as well.

We now have our MP4 file. Let's proceed to playback.

7 ISO MPEG−4 Playback

The MPEG4IP Player might not be the best considering its interface and options but it's the one of the best if not the best when it comes to actually playing the streams. I have tried all MPEG−4 Players that I am aware of and only this one played the file 9 times out of 10. Versions 0.9.8.2 and later support plugins and therefore expect this to be massive with the required attention.

MPEG4IP offers two versions of the player as mentioned earlier. The original player is command line based and then there is a GUI available. The CLI player is mostly there for people that cannot use the GUI for some reason.

Assuming that you have copied all files in the same directory after compilation, you may execute wmp4player.exe. If you want the player only you will need the following files:

- wmp4player.exe
- wmp4client.exe
- SDL.dll
- mp4player.exe (CLI player)

And all the plugins you can have of course. They have the .dll extension and at the moment they ones available are:

- aac_plugin.dll – AAC playback
- mp3_plugin.dll – MP3 Playback
- mpeg4_iso_plugin.dll – ISO MPEG−4 Video decoder
- xvid_plugin.dll – XviD playback

Now launch wmp4player.exe as suggested earlier.
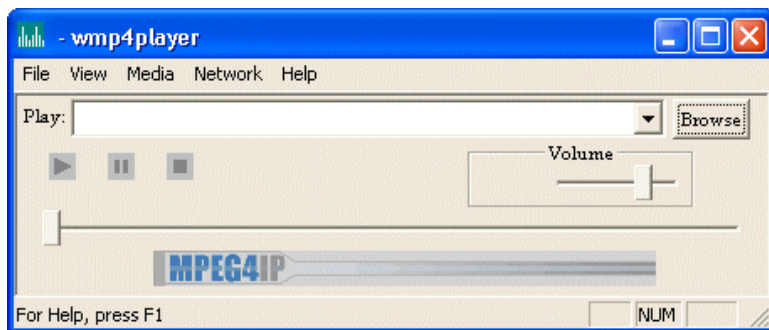
Screenshot of the neat player.

You can now Open your mp4 file and enjoy!

MP4Player (CLI) key shortcuts

If you are using mp4player, the following key short cuts can be used, assuming you've got a video window:

- HOME – begin video from beginning
- PAGE UP – increase video size (200% max)
- PAGE DOWN – decrease video size (50% min)
- ALT−ENTER – fullscreen
- ESC – return from fullscreen
- LEFT ARROW – go back 10 seconds
- RIGHT ARROW – go forward 10 seconds

- UP ARROW – volume up 1/10th
- DOWN ARROW – volume down 1/10th
- SPACE – pause or continue
- CTRL–C – close video (next if running playlist)
- CTRL–X – close mp4player

Now to the most interesting – Streaming!

8 Setting up Darwin and streaming

To stream MP4 files you will need the Apple Darwin Streaming server. You can download it from here.

Installing the Darwin Server

I assume that you have downloaded the binaries for your platform. Binaries exist for the most popular platforms. If you would like to use it elsewhere you can have a look at the README on how to compile it.

Linux/Unix Installation

Uncompress the gunzipped tar archive:
tar xvzf DarwinStreamingSrvr4–platfom.tar.gz

Change dir to the directory the files are:
cd DarwinStreamingSrvr4–platform

Then simply type:
./Install

The installation script will launch streamingadminserver.pl. This Perl script will let you set the username and password for the Web Administration of the Darwin Server. The media folder will then be /usr/local/movies.

Windows NT/2K/XP Installation

Execute the self–extracting .exe and unzip the files in a directory. e.g. C:\Darwin Streaming Server. Be aware that you need Perl for Windows for the Web Administration Server that can be downloaded here.
Yet again execute Install.bat in that directory and follow the instructions.

When the process is done, the server will be installed in C:\Program Files\Darwin Streaming Server\ and will be added as a Service and loaded on every boot. The media folder will then be C:\Program Files\Darwin Streaming Server\movies.

Configuring the Server

Before we start playing with the server you may want to launch streamingadminserver.pl during every bootup.

If this is a one–time configuration, then simply execute streamingadminserver.pl. On Linux/Unix just type the command on the console and on Windows, double click on the file assuming you have Perl installed.

Open your browser and open the URL http://localhost:1220. You can do that remotely as well by replacing localhost with the server local host name or IP address.

The server will require you to login with the same username/password you defined when installing.

Log In to Darwin Streaming Server

Log in with an administrative username and password.

QuickTime

User Name: everwicked

Password: ******

Log In

Server Snapshot

Server:	localhost
Status:	Started Thu, 18. Apr 2002 08:54:54
Current Time On Server:	Thu, 18. Apr 2002 18:35:06
Up Time:	9 hrs 40 min 11 sec
DNS Name (default):	localhost
Server Version:	4.0 [v410]
Server API Version:	4.0
CPU Load:	4.86%
Current # of Connections:	0
Current Throughput:	0 bps
Total Bytes Served:	0 Bytes
Total Connections Served:	0

Congratulate yourself with the Server Snapshot.

Main

Connected Users
Relay Status

General Settings
Port Settings
Relay Settings
Log Settings

Playlists

Error Log
Access History

Log Out

The menu on the left will give you all the options you need.

The General Settings will let you define the Media directory, the Maximum number of connections and Max Throughput. It also allows you to enable SSL for the administration. If you would like to change the admin's login information you may do that there as well.

The Port Settings will allow you to Enable streaming over port 80 for hosts that connect via a firewall. This might conflict with a web server though, use it wisely.

There are more options you can play with but these should be enough to start broadcasting.

To test the server, copy the file samples/test_qcif_200_aac_64.mp4 from the MPEG4IP directory to the server's media directory and launch wmp4player.exe

Instead of browsing for a file, type rtsp://localhost/test_qcif_200_aac_64.mp4 and tap enter. If you get a small window with a nice guy talking to his camera then there you go, you have a Streaming Server!

Some people might have problems streaming off a server because of firewalls block RTP packets. In that case, check the Use RTP over RTSP option under the Network menu in the MPEG4IP player.

9 Live Broadcasting with mp4live

mp4encode is a Linux ONLY audio/video capture utility that enables to capture and encode audio and/or video in realtime. The output can be written locally or even broadcasted directly!
Note: I do not have a capture card to provide accurate screenshots of this but this document is the best resource I could find available at the moment. If you do have a capture card and use mp4live I would be happy to post screenshots here.

Let's start with the requirements though.

Hardware Requirements

- Pentium III class machine of at least 500 MHz. The developers state that the performance is not only due to the CPU but other factors as well.
- A sound card with an OSS compatible driver and capture ability; a video device with a video4linux (v4l) compatible driver and memory mapped capture ability.
  Known to work are: Video capture cards based on the Brooktree 8x8 chip with an analog A/V source, Logitech Quickcam Express Webcam.

Software Requirements

- Linux Kernel 2.4 or later
- Drivers for sound and video devices
  - ♦ bttv 0.7 video capture driver or
  - ♦ qce webcam driver

Process

Launch mp4live (must be in X11 for that). Before you start capturing with your own settings, it's best to just hit Start. The program will record 1 minute of video and you will be able to check if something is broken before you go deeper.

If everything is OK, you can start chaning the settings to your needs.

The GUI is quite self–explanatory, you can click on the settings button of each part and change the Video, Audio, Recording and Transmission settings.

From there you will be able to change the Capturing/Brodcasting Format, the target bitrate and the framesize/frame rate.

You can also choose between AAC and MP3 for the Audio and the target bitrate as well.

As mentioned above, you can capture
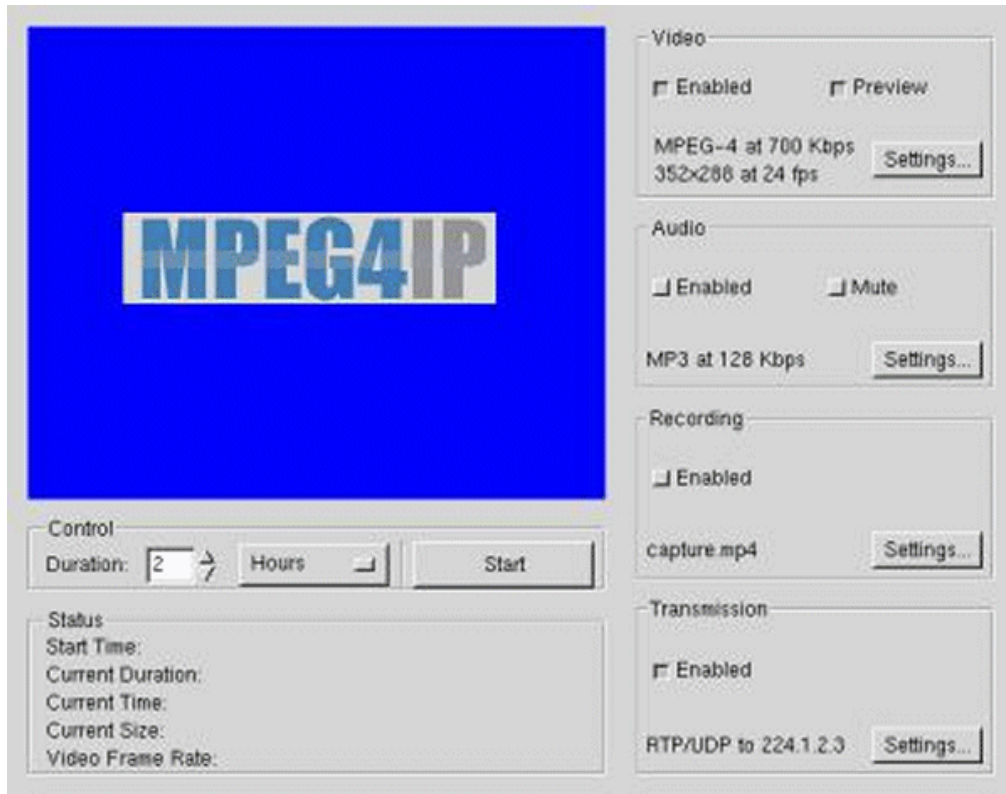
to disk, broadcast or do both! Therefore to enable local recording, tick the Recording option or the Transmission option accordingly.

Broadcasting the file is really easy. Click on the Transmission settings. Change the default IP to something that fits your server and save the .sdp file in your media directory. To stream the broadcast, simply redirect wmp4player to rtsp://myserver/live.sdp, where myserver is the IP of your server.

Your settings will be remembered and if you have a standard broadcast for example your daughter's cat, you may invoke mp4live on X11 startup with the −automatic flag to start broadcasting immediately or even with −headless to dismiss the interface.

Next is the sad part, legal information..

10 Legal Information

Just like every other MPEG–4 application, using MPEG4IP and the Apple Darwin Streaming Server is not free for every use. MPEGLA will stand for its rights each time you violate the MPEG–4 licence. Mostly for companies that is.

Anyway, this guide involves a lot of issues in that manner so I thought I should put some legal information on these issues. If you find any errors on this page, feel free to contact me with corrections.

From this point on, *I assume that the content is not copyrighted or if it is you have a legal copy of the original and you will use the technology for your own purposes.*

DivX

If you choose to encode your video with DivX Lite or Pro, you are limited to use the content for your own personal use. If you would like to use that content commercially then you will need to contact DivX Networks. In that case DivX Networks can deal with the legalities of streaming MPEG–4 media.

XviD

XviD's source code is distributed under the General Public licence (GPL).

The GPL allows the source to be changed but the problem is still with MPEGLA. Even though XviD as an MPEG–4 implementation is free to use for any purpose, any MPEG–4 media has to be licensed. For the same reason, it is illegal to distribute XviD binaries in most countries since any MPEG–4 encoder has to pay licences to MPEGLA for every download.

Thus, before you fire up Virtualdub and molest its job list, make sure your network is really "Local". Well, alternatively you can visit MPEGLA for a more pleasant experience.

LAME

The LAME MP3 encoder is under the same situation with XviD. You may use it freely but be aware, Fraunhoffer holds the patents.

AAC Audio

FAAC falls in the same category with LAME and XviD. Just like with the other two applications you will need to contact the AAC patent holders.

The PsyTEL AAC encoder is a little different, read on if you are going to use it.

The PsyTEL encoder is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties.

There are Windows binaries released by PsyTEL that may be installed and used free of charge but only of testing and evaluating the encoder. To use their software for another purpose you need to licence it.

Of course you may not create content that is going to be sold, published or distributed without the commercial licence. That includes all kinds of distribution i.e. forget streaming AAC audio created with the PsyTEL encoder.

For more information contact PsyTEL.

MPEG4IP

MPEG4IP contains original code, and copies of other open source packages.

The original code that the MPEG4IP developers have written, is under the Mozilla Public licence 1.1.

Each other package that is included in MPEG4IP is listed here and you should consider the patents issue before you start building up your system for streaming.

Apple Darwin Streaming Server

The Streaming Server is under the Apple Public Source licence (APSL) and yet again the server may not be used for commercial uses. You may licence the Quicktime Streaming Server for that purpose.

All in all, the software is free but if you want to use it commercially you need to deposit money to the patent holders. Beware that the list of software packages mentioned above is not complete and if you consider using any of the above commercially you should read the licences of other transparent software bundled with MPEG4IP such as libraries.

Links: APSL | Mozilla licence | GPL

11 Developing with MPEG4IP

The world of MPEG4IP is very exciting and I would urge anyone that has experience in C++ and some time to spare to start working on the project.

At the moment there are only two (great) developers working on MPEG4IP and they could use some help as far as I know.

Please let me send a big Kudos to them:
Kudos to Bill May and David Mackie from Cisco!

If you are interested in this, download the source, check what needs to be done and see if you can do it. If you don't know how but you are willing to, the developers will be glad to point you to the right resources.

As a note of what this project is going to bring I am stating that the developers are working on adding support for H.26L and DV support as well as more SSE2 optimizations. Also, more codec plugins for the player are being developed.

Enjoy streaming across your neighbourhood :)