

Codecs Draft Recommendations

From Transmission

Contents

- 1 Introduction
 - 1.1 Scope and Method of Research
 - 1.2 Recommendations
 - 1.3 Supporters
- 2 Summary of Recommendations
 - 2.1 Adoption of Existing Tools
 - 2.2 Development of Tools
- 3 Codecs and Containers
 - 3.1 Download to Screen / Watch on PC
 - 3.1.1 Avi/Mov Xvid/LAME
 - 3.1.2 Ogg Theora/Vorbis
 - 3.2 Playback Within Web Browser
 - 3.2.1 Flash video using FOSS tools
 - 3.2.2 Ogg Theora/Vorbis
 - 3.3 Video for Mobile Devices
- 4 Players
 - 4.1 FOSS Players for FOSS Codecs
 - 4.1.1 VLC
 - 4.2 Support in Proprietary Players
- 5 Transcoders
 - 5.1 Windows
 - 5.1.1 MediaCoder for Xvid and Ogg
 - 5.2 Mac
 - 5.2.1 ffmpegX for Xvid
 - 5.2.2 Simple Theora Encoder for Ogg
 - 5.3 Linux
 - 5.3.1 Avidemux for Xvid
 - 5.3.2 Ogg Convert for Ogg
 - 5.4 Cross-Platform
 - 5.4.1 Avidemux for Xvid
 - 5.4.2 VLC for Ogg
 - 5.5 Support in Proprietary Transcoders for FOSS Codecs
 - 5.5.1 Ogg for Quicktime - XiphQT Component
 - 5.5.2 Xvid for Quicktime

- [5.5.3 Ogg for Windows - Directshow](#)
- [5.5.4 Xvid for Windows - Directshow](#)
- [6 Server-Side Transcoding](#)
 - [6.1 MEncoder - Xvid/LAME](#)
 - [6.2 ffmpeg2theora - Ogg Theora/Vorbis](#)
 - [6.3 Transcoding Web Services](#)
- [7 Transcoder/Uploader](#)
 - [7.1 v2v / Transmission Uploader](#)
 - [7.1.1 Functionality - Details](#)
 - [7.1.2 User Experience](#)
- [8 Browser Playback](#)
 - [8.1 Flash Pseudo-Streaming and Browser Playback](#)
 - [8.1.1 Indytube and OS Flash](#)
 - [8.1.1.1 Indytube Extension for Flash Streaming - Red5](#)
 - [8.2 Ogg Streaming and Embedded Playback](#)
 - [8.2.1 Cortado Java Applet](#)
 - [8.2.1.1 vPIP and Cortado](#)
 - [8.2.2 Browser Plugins](#)
 - [8.2.2.1 Annodex Firefox Plugins](#)
 - [8.2.2.2 MPlayer Plugin for Linux](#)
 - [8.2.2.3 VLC Firefox Plugins](#)
 - [8.2.2.4 OggPlay Firefox Plugins](#)
 - [8.2.3 Native Browser Playback of Ogg](#)
 - [8.2.3.1 Opera](#)
 - [8.2.3.2 Mozilla](#)
 - [8.3 Javascript Support for Multiple Playback Methods - Ogg](#)
 - [8.3.1 MV_Embed Script](#)
 - [8.3.2 ArcticGiant](#)
 - [8.4 Indytube and Ogg/Vorbis](#)
 - [8.4.1 Indytube Extension for Ogg Transcoding - ffmpeg2theora](#)
 - [8.4.2 Indytube Extension for Ogg Embedding - Cortado](#)
 - [8.4.3 Indytube Extension for Ogg Streaming - Icecast](#)
 - [8.4.4 Indytube Development for Ogg Embedding - MV_Embed](#)
- [9 Support in Popular CMSs](#)
 - [9.1 Drupal](#)
 - [9.1.1 Film Forge](#)
 - [9.2 Plone](#)
 - [9.2.1 Plumi](#)
 - [9.3 Joomla](#)
 - [9.4 WordPress](#)
 - [9.4.1 Show In A Box](#)
- [10 Future of Web Video](#)
- [11 Where to From Here](#)

- [12 Glossary of Terms](#)

Introduction

Over the last 40 years the means to create and disseminate the moving image have become increasingly democratised. From Portapak to camcorders, VHS to DVD, video technologies have continued to open access to ordinary people to tell their own stories. The explosion of online video distribution has brought new opportunities and challenges for those committed to making sure these technologies are not just available to purchase and use, but that their evolution can be shaped collectively; that they are free to be modified, built upon and shared.

There is a growing need to ensure that the technologies for distributing our media are not held in the hands of a small number of corporations, locked in onerous patents or trapped by proprietary software that may be priced out of our reach at any moment.

These ideas and needs bring us to some rather technical questions about the compression and decompression of video today. This research report seeks to outline the current state of Free and Open Source Software (FOSS) codecs and containers, particularly how they might evolve and be implemented by social justice organisations within the [Transmission network](#) for which this report was originally envisioned. Transmission is a network of citizen journalists, video makers, artists, researchers, programmers and web producers who are developing online video distribution tools for social justice and media democracy.

Beyond the ethical reasons for choosing FOSS there are also a range of practical ones. Most social change video makers cannot afford proprietary tools or have more pressing financial priorities. Additionally, with the right approach open source development practices can facilitate the rapid development of software that is just as good or even better than their proprietary counterparts. Whilst many activist groups have a commitment to using FOSS generally, most currently employ or promote proprietary video codecs such as Flash, Quicktime and Windows Media. This however is largely due to the historical lack of easy ways to employ FOSS solutions in this area, a problem this research aims to help change.

[EngageMedia](#) has lead this research project. The report has been authored in the most part by Anna Helme with support from Andrew Lowenthal and Andy Nicholson as well as input from various members of the Transmission network and other FOSS video tool developers. All recommendations are from EngageMedia only, though this draft will be submitted to the Transmission network for further input in order to make any

changes necessary and encourage consensus around the best ways forward for collaborative development in this area.

Scope and Method of Research

The research was comprised of two distinct areas. One through the traditional gathering, analysis and filtering of materials. The other through a degree of experimental development to examine the potential to extend available tools and open up new ways to use FOSS codecs within our video-sharing platforms. The research asked a number of questions which evolved somewhat during the research project. Questions addressed were:

- What FOSS codecs are currently available and how do they compare?
- What media players are available that handle playback of FOSS codecs? What are the best players to use for each platform, and the best for Transmission members to recommend to their users?
- What desktop tools currently exist that support encoding to open source codecs? What can be done to improve the ability to encode to FOSS codecs in available tools?
- What are the best available FOSS server side auto-transcoding applications to take the onus off video producers to encode to FOSS video formats?
- How can we make it as easy as possible for a user to transcode to FOSS formats then upload it to the video platform? Conduct research into the development of a desktop transcoder/uploader application that could work locally to encode the file and then upload it to a specified site.
- What software is available for the playback of FOSS video within the browser? What is the best way to go amongst the various options given changes in technology, and what further development is required to make these solutions more viable?
- What is needed within the most popular FOSS content management systems to enable the upload and playback of FOSS codecs?

Recommendations

The report provides a general understanding of FOSS audio/video codecs and recommends what we believe are the most viable options for implementing them within social change online video projects. These recommendations consist of a list of available tools, suggests the most usable amongst them, finds areas that need to be documented further and contains a set of recommendations for development which Transmission projects and others may contribute to. It aims to encourage collaboration among projects committed to free technologies to avoid replication, conserve resources and ultimately fire up development to build great tools.

A process for following up this report is outlined [here](#) at the end of the report.

Supporters

This research was funded by the Open Society Institute with additional assistance from the Donkey Wheel Foundation and Open Channel.

Summary of Recommendations

Adoption of Existing Tools

Primary recommendations for software that the Transmission network can adopt to increase the uptake of FOSS codecs are as follows:

Codecs and Containers

The use of two FOSS video and audio codec combinations are recommended:

- **Xvid/LAME** Use of the Xvid codec for video and LAME codec for audio in the medium term as both codecs are well developed and generally well supported but are encumbered with patent issues that mean they may be plagued with legal issues into the future.
- **Ogg Theora/Vorbis** Use of the Ogg format, Theora video codec and Vorbis audio codec as they are patent-free technologies that have seen great improvements in usability and the number and quality of tools available to create and play them is increasing.

Players

There are many good FOSS players of both Xvid/LAME and Ogg Theora/Vorbis available, some more suitable for certain purposes than others. However the recommendation is that Transmission projects promote VLC as it is highly-developed for all platforms and means we can share documentation and training resources more easily by adopting the same software:

- **VLC** Use VLC as it is highly-developed FOSS media player for FOSS codecs on all platforms.

Transcoders

The following transcoders are recommended to create Ogg Theora/Vorbis video files:

- **MediaCoder** (Windows) Use MediaCoder as it is a very usable and highly-configurable FOSS transcoding application with good presets
- **Simple Theora Encoder** (Mac) Use Simple Theora Encoder as it is a very easy-to-use FOSS application for creating Ogg files
- **Ogg Convert** (Linux) Use Ogg Convert as it is a very simple and usable FOSS application to create Ogg files

The following transcoders are recommended to create Xvid/LAME video files:

- **MediaCoder** (Windows) Use MediaCoder as it is a very usable and highly-configurable FOSS transcoding application with good presets
- **Avidemux** (Linux) Use Avidemux, an easy yet configurable FOSS transcoding application with good presets
- **FfmpegX** (Mac) Use FfmpegX as it is a simple yet configurable Shareware transcoding GUI application built on FOSS transcoding software

Server-Side Transcoding

- **ffmpeg2theora** Install the FOSS ffmpeg2theora on the server to create Ogg Theora/Vorbis files automatically for the user, that can then be played back in the browser with other FOSS technologies
- **Mencoder** Install the FOSS Mencoder on the server to create Xvid/LAME files automatically for the user

Development of Tools

Primary recommendations for further development by members of the Transmission to increase the usability and implementation of FOSS codecs are as follows:

Transcoder/Uploader

- **v2v/Transmission Uploader** Contribute to the development of the v2v Upload transcoder/uploader application created by Jan Gerber, specifically by:
 - generalising the application for Transmission projects by creating a configuration file that can be edited by each project to include its own settings with the program
 - bugfixes and testing
 - implementation of the Transmission metadata standard
 - promotion and documentation

Browser Playback

- **Ogg Theora/Vorbis Support in Indytube** Develop Ogg support in Indytube from present alpha stage by:
 - adding the MV_Embed script to support all available Ogg playback methods - the Cortado java applet, browser plugins and native browser support for Ogg
 - use of the Icecast streaming server
 - promotion and documentation
- **Cortado Java applet** Contribute to development of Cortado java applet by:
 - re-writing to include support for latest version of Theora
 - enhanced controls including full-screen, timecode display and volume
 - promotion and documentation

Support in Popular CMSs

- **Integration of Indytube with Ogg Support** Integration, documentation and promotion of Indytube with Ogg support into Plumi, FilmForge and Showinabox for server-side transcoding and browser playback
- **Integration of Transcoder/Uploader** Integration, documentation and promotion of transcoder/uploader into Plumi, FilmForge and Showinabox for easy creation and upload of Ogg into CMSs
- **Integration of vPIP with Ogg Support** Integration, documentation and promotion of vPIP with Ogg support into Plumi, FilmForge and Showinabox for browser playback

Codecs and Containers

What FOSS codecs are currently available and how do they compare?

We asked this question in order to establish which FOSS codecs [1] should be adopted by the Transmission network. Many comparisons of codecs and containers [2] already exist, and though there is a need to update these often as the technology is constantly changing, it was not considered necessary to replicate these individual comparisons but to aggregate their results. Research involved reading existing research on the subject and identifying what was most appropriate for the needs of Transmission network members, considering:

- Development community and future potential
- Future relevance due to external factors
- Support or potential for support in players, transcoders and browsers
- Relative quality of results (quality-compression ratio) to proprietary codecs and formats
- Efficiency - time to encode
- CPU load on playback / encode - compatibility with older/low-powered machines
- As-is potential for use by Transmission network members, and future potential given more development in terms of usability
- Documentation
- Ability to embed metadata within the video file such as license info
- Patents that may inhibit the distribution of codecs and containers, despite them being FOSS and licensed under free or open source licenses such as the GPL.

The chapter by Denis Rojo (Jaromil) and Eleonora Oreggia [Video Streaming Across Wide Area Networks](#) resulting from R&D activities in Montevideo/TimeBasedArts includes tabled results of benchmarks that are very useful. Wikipedia's comparisons on container formats and codecs were considered, as were the video codec comparison website doom9 and other codec comparisons in addition to project websites with documentation about the codecs and containers themselves.

Containers

The FOSS and patent-free container formats Ogg, Ogm, Matroska and NUT were examined. [Ogm](#) (Ogg Media)

is a format that may contain codecs such as Theora and Vorbis but also non-Xiph codecs and has features centering on chapters and multiple audio tracks like on DVDs. Ogm is considered a temporary solution and will not likely have much continued development. Matroska is a format designed to contain any codec and an unlimited number of audio and video streams, and is being developed to allow chapters and DVD-like menus, but is not popular or well enough supported by players and transcoders to be considered at this stage though it should be kept in mind for the future. NUT is another container developed by some people from the Mplayer and ffmpeg projects but is in early stages of development and not ready to be considered at this stage.

Codecs

Research was focused on video codecs rather than audio codecs, though audio codecs complementary to the video codecs were considered. Both LAME and Vorbis are well known and excellent options for FOSS audio codecs, though LAME has patent issues to do with the use of mp3 software developed originally by Fraunhofer.

Most video codecs considered were based on the DCT (discrete cosine transform) method of video compression. Newer wavelet-based compression techniques were considered also, though these are not ready for adoption.

Both Xvid and x264 are based on different parts of the MPEG4 standard, for which many patents are held and managed by the MPEGLA (MPEG Licensing Authority) despite being free and open source development projects. This means that codecs based on this standard are encumbered by patent issues and are often only distributed as source code by the developers and available as binaries by third-parties only. X264 may or may not offer superior quality over Xvid, but is less popular and well-supported than Xvid so was left out of the recommendations.

Theora is based on vp3, which is patented by On2 Technologies, but On2 has released any users of this software from any patent restrictions so it can be considered patent-free and is recommended for its quality and its increasing usability and potential to be included in future as part of native browser support of video.

Dirac (BBC) and Snow (ffmpeg) are both wavelet-based codecs that are not mature enough to be used at this stage, but promise significant gains in compression ratio in the future. The wavelet-based Tarkin (Xiph) is little more than a proof-of-concept and is not being developed at present. OpenAVS is an open-source fork of a Chinese patented codec but is still in early development so is not part of these recommendations either.

Research findings are split into three use-cases:

- to watch on PC or to screen video after downloading
- viewing video within the web browser
- viewing on mobile devices

Download to Screen / Watch on PC

Various container-format/codec combinations were considered in terms of usage on personal computer using a player application such as VLC player, once the file has already been downloaded.

Two FOSS video/audio codec and container format combinations are recommended here. The Xvid/LAME video/audio codecs are judged to be a good temporary measure, to be phased out with the increasing usability of Theora/Vorbis.

Avi/Mov Xvid/LAME

Both the AVI and MOV (Quicktime standard) container formats support the use of the Xvid video codec and the LAME audio codec. Both these codecs are capable of producing good or excellent results with a high quality to file-size ratio especially at the bitrates of Xvid 900-1400 kbit/s and LAME 96-160 kbit/s.

Both Xvid and LAME were originally based on proprietary codecs - Xvid upon the DivX 3.11a hack of a Microsoft MPEG4 codec and LAME on the Fraunhofer mp3 codec. Though they have evolved somewhat since then (or have been substantially or entirely re-written) they still have underlying patent issues particularly with Fraunhofer and the pool of MPEG-4 patent owners that hinder their distribution and may cause further difficulties in future if the owners of those patents pursue legal proceeding against anyone using these codecs.

In their favour, they both have a relatively high level of uptake. Any hardware device or software player capable of playing mp3 will be able decode the LAME audio codec. As Xvid is based on the MPEG4 standard, many hardware DVD players are set up to play back DivX and are therefore also capable of playing Xvid, which hugely increases its usability. An Xvid component is available for Quicktime, and a directshow filter is available for Windows so most people will be able to play Xvid without too much trouble. Of course, the VLC player will play both codecs in either the AVI or MOV container format on Linux, Mac and PC in addition to many other platforms.

However it is thought better not to invest heavily in FOSS codecs that are unencumbered by patent issues which may result in a legal risks to anyone using them in future. The recommendation is that Xvid is still supported by the Transmission network until such time as Ogg Theora/Vorbis is supported by the members' user base

adequately. It is not recommended that Transmission members contribute substantially to development of the Xvid codec or applications built around it.

It is worth noting that encoding applications that use the the ffmpeg libavcodec do not actually encode to Xvid but to an ffmpeg re-implentation of an mpeg4 standard codec that is compatible with Xvid and may be tagged as Xvid by the application.

Xvid Downloads, Documentation and Development



Download: (Windows/Linux) <http://www.xvid.org/Downloads.15.0.html> (Mac) <http://www.xvid.org/Software.83.0.html>



Documentation: <http://www.xvid.org/FAQ.14.0.html>



Development <http://www.xvid.org/Xvid-Codec.2.0.html>

Lame Downloads and Documentation



Download: <http://lame.sourceforge.net/links.php#Binaries>



Documentation: <http://lame.sourceforge.net/using.php>



Development: <http://lame.sourceforge.net/>

Ogg Theora/Vorbis

The Ogg container format, Theora video codec and Vorbis audio codec are developed by the Xiph.org Foundation and are patent free technologies as well as FOSS. Theora and Vorbis have very good quality to file-size ratio, especially in the Theora 900-1400 kbits/sec and Vorbis 64-160 kbits/sec ranges. Vorbis in particular has a very good reputation for quality at this bitrate range.

The problems facing Ogg Theora/Vorbis are really about usability and uptake. While there are many Vorbis transcoders available, until recently Theora was only transcodable on the command line. While some argue that

this is not difficult, the fact is that most general users and video makers would never even contemplate using an application without a graphical user interface.

This situation has begun to change in the last few months, with the work by Jan of v2v on the Simple Theora Encoder - a very simple graphical user interface for his ffmpeg2theora application. Currently a disk-image containing a binary of this application is available for Intel Mac which is easy to install and very easy to use. Versions of this application for other platforms are in the pipeline. This application (and its related application, the transcoder/uploader v2v Upload) is key to the uptake of Ogg Theora/Vorbis, by enabling users to create these files easily on their personal computers.

In terms of players, Ogg codecs are supported by VLC and Mplayer but by few other players. For the Transmission network this should not be seen as too much of a hurdle, as it is possible to distribute the FOSS VLC player or offer a link to download it from the VLC website. Both installation and use of VLC is very easy.

Theora Downloads and Documentation

Theora is not released as a codec to download, but as libraries available to be implemented into applications that can encode it or decode it (either make Ogg Theora files or play them back). Many of these applications for using Theora are listed in this report.



Documentation: <http://en.flossmanuals.net/theoracookbook>



Development: <https://trac.xiph.org/report/10>

Vorbis Downloads and Documentation

The Get Setup area of Vorbis.com lists binaries for download that utilise the Vorbis codec. The link to documentation is a FAQ rather than a user guide. User documentation can be found with the transcoder or player software that uses Vorbis, as for Theora also.



Download: <http://www.vorbis.com/setup/>



Documentation: <http://www.vorbis.com/faq/>



Development: <https://trac.xiph.org/report/11>

Playback Within Web Browser

Two solutions are recommended here for the meantime - one is currently available but only uses FOSS tools in combination with the proprietary Flash video format. The other solution of using Ogg Theora/Vorbis for playback within the web page needs work urgently to bring it to a more usable stage, though this could be done fairly easily at this point. Details on software available to support the playback of FOSS codecs within the browser is covered in another section of this report [below](#).

Flash video using FOSS tools

Flash video is now the most common software for viewing video within the browser, having defeated RealNetworks, Quicktime and Windows Media. The Flash browser plugin was originally designed for shockwave vector animations by Macromedia, and adapted for video using codecs developed by other companies Sorenson and On2. Adobe (who now own Flash) claim that 98.7% of internet users have the Flash plugin (they do not specify which version in this statistic, or if that version is video-capable).

There is a large open source Flash community and much development in this area. The Red5 server is available as a FOSS tool for streaming Flash video files. The open source Flash video player Flowplayer is available for any Transmission project to include within their web pages, and there are many others available also (see Browser Playback section below). Mencoder can be used to transcode to Flash video on the server-side.

The FOSS Indytube "YouTube-like" code package uses both Flowplayer and mencoder to enable server-side transcoding and browser-playback of Flash video, and has ensured that this is a very viable option for Transmission network members and beyond (see [http://wiki.transmission.cc/index.php/Codecs_Draft_Recommendations#Indytube_and_OS_Flash] section on Indytube and OSFlash below).

Ogg Theora/Vorbis

Server-side transcoding of Ogg Theora/Vorbis is available with ffmpeg2theora and currently works at alpha level in the Indytube package (see below). Uptake of this should be fairly straightforward from this point, with implementation into the popular content management systems Plone and Drupal happening in the near future.

Browser playback is covered in more detail below, but the options here are improving with the advent of an

experimental VLC plugin for Mozilla browsers and experimental work on native support for Ogg Theora/Vorbis by both Mozilla and Opera, in addition to improvements to the Cortado Java applet which will play Theora back in any browser so long as the user's system has Java installed.

Ogg streaming is available using the Icecast server (see Streaming section below)

Video for Mobile Devices

This area has not been investigated as part of this research. Further research into the availability and usability of FOSS codecs for mobile devices needs to be conducted. Many more people will have access to a mobile phone than a personal computer in the next few years. When considering the digital divide the importance of mobile technology globally cannot be ignored.

Transmission network members are encouraged to have input into this area in particular at this stage and to begin a discussion about relevant technologies, including which video formats may be appropriate to use.

It is worth noting that Opera has released a version of its browser called Opera Mini which is designed to run on mobile devices. As Opera is including support for native playback of Ogg video in future (see below) this may mean Ogg will be supported in this trimmed down version of their browser also, and thus allow for playback on mobiles.

Players

What media players are available that handle playback of FOSS codecs? What are the best players to use for each platform, and the best for Transmission members to recommend to their users?

It is important to consider which media players support the codecs we are recommending, in order to ensure our audiences can watch the files we distribute. There are many players available for different platforms with varying support for Theora/Vorbis and Xvid/Lame. There is a very good and exhaustive comparison page on Wikipedia that lists the features and compatibility of media players with various video and audio codecs and formats Comparison of media players. This was cross-referenced with documentation and some testing of the players themselves.

FOSS Players for FOSS Codecs

This is a subset of available FOSS media players that support Theora/Vorbis and LAME/Xvid, discounting

those without much continued development:

NAME	OS	STREAMING	Lame/xvid	vorbis/THEORA	VODCAST
Kaffeine (KDE; xine)	U	yes	yes	yes	no
Kplayer (KDE; mplayer)	U	yes	yes	yes	no
Media Player Classic	W	yes	yes with plugin	yes with plugin	no
Mplayer (libavcodec, libavformat, xine)	W M U	yes	yes	yes	no
Miro	W M U	no	yes	yes	yes
NicePlayer	M	yes	yes with plugin	yes with plugin	no
Songbird	W M U	yes	yes	yes	yes
Totem Media Player (GNOME; gstreamer/xine)	U	yes	yes	yes	no
VLC Media Player	W M U	yes	yes	yes	no
XBoxMediaCenter	W(p)	yes	yes	yes	no
Xine-ui (xine)	U	yes	yes	yes	no

U = Linux or Unix-like operating systems M = Mac OS X W = Windows (p) = partial support

The ability to act as a streaming client or vodcatcher (RSS video podcast feed reader) is noted for those Transmission members for whom this is relevant. Often users of Transmission member sites may use one application for streaming content, one for vodcasts and one for general media playing.

Songbird is an interesting application, as it aims to be a media player / web browser to enable you to "play the web". Miro is a vodcatcher and player and is very useful for both browsing content and presenting it in a playlist for screenings.

Linux users in particular have a good range of options to choose from, and will probably choose their favourite player to suit their own needs. XBMC (XBoxMediaCenter) is for the Xbox game hardware, but will also run on some Windows OSs. Mplayer, Miro, Songbird and VLC will all play on Linux, Mac and Windows but out of the four VLC is the most user-friendly application for an audience who simply wishes to view downloaded media (in addition to other functions such as streaming and encoding which can be found deeper in the menus).

Any of the players listed are good choices for individuals who wish to playback FOSS codecs on their computers, but for the sake of simplicity and for the reasons outlined below it is recommended that Transmission members promote the use of VLC.

VLC

The major recommendation for player software for FOSS codecs is VideoLan Client (VLC). It's advantages are that it is cross-platform, stable and usable, and can play back nearly any audio or video format available).

The multi-platform VLC can run on Windows, OSX, Linux and PocketPC / WinCE handhelds along with other systems. VLC also works well as a streaming server and video transcoder too (to both Xvid and Theora, the two recommendations of this report).

Another advantage of encouraging the use of this player by our network members and their users is that we will be referring to one documentation base if we all use the same player.



Screenshot of VLC player

VLC Documentation, Downloads and Development



Download: <http://www.videolan.org/vlc/>



Documentation: <http://www.videolan.org/doc/play-howto/en/play-howto-en.html>



Development <http://www.videolan.org/developers/>

Support in Proprietary Players

Quicktime / iTunes

Perian is a multi-codec package for Quicktime on Mac that currently supports Xvid and many other codecs Quicktime does not natively support. Apparently only for OS 10.4 and above "Requirements: Mac OS X 10.4.7 because 10.3 makes the iceweasel cower in fear." The issue tracker lists .ogg support as an upcoming addition.

There is a Quicktime Xiph component that will allow the playback of Xiph codecs including Ogg Theora/Vorbis using Quicktime on both the Mac and on Windows, available for download from Xiph.org.

These Quicktime components will allow the playback of Theora/Vorbis and Xvid/LAME files in any application that uses Quicktime architecture on a Mac, including iTunes.

Windows Media Player / Winamp

The Illiminable DirectShow Ogg Filters are available as a binary to download and install which will allow the playback of Ogg Theora/Vorbis files using Windows Media Player or other applications that use directshow filters such as Winamp.

From the Xvid.org website there are links to download the Xvid codec that will enable the playback of Xvid files in Windows Media Player or Winamp.

Transcoders

What desktop tools currently exist that support encoding to open source codecs? What can be done to

improve the ability to encode to FOSS codecs in available tools?

There are many transcoding applications available for many different platforms. An assessment was made of currently available tools in order to make a comparison and identify the most usable tools for each platform. In general we did not look at high-end proprietary transcoding applications such as Cleaner and Pro Coder, and focused on FOSS applications available at no-cost.

Graphical user interfaces were also favoured, as there is unlikely to be a large uptake of command line interfaces, despite how easy they can be to use once the the command line paradigm has been mastered. It can be difficult to convince programmers of this fact, but most video makers will not be persuaded to use the command line to encode video. Transcoders for the command line are recommended for Linux users only below.

Lists of video transcoders can be found at Afterdawn.com, VideoHelp.com. There is no comprehensive comparison of different transcoders on Wikipedia, but as there are so many different programs, and many applications are weighted towards particular use-cases or codecs this would be a huge task. Other lists of video transcoders can be found on download.com or versiontracker.com and searching in the correct categories. A list of [Theora encoders](#) can be found on the Xiph.org wiki.

There was input from two Transmission members into this research question. Firstly Jaromil from Dyne highlighted the need to address development of clean encoder/decoder examples for software developers in various languages, in form of libraries, as well direct support to the FFMpeg project. Mick Fuzz from ClearerChannel indicated the need to implement easy presets in transcoding applications in order to make transcoding easy for beginner users and video makers without extensive computer skills outside video editing applications.

Windows

There are numerous FOSS transcoders available for the Windows platform. [Virtual Dub](#), derivatives of this program and other early video converter applications have had a large base of users converting to DivX and subsequently Xvid, especially in the DVD-ripping community. Virtual Dub itself has a confusing and complicated user-interface and cannot be recommended for beginner users. [SuperC](#) is a freeware GUI for mencoder, ffmpeg and ffmpeg2theora that will transcode to Xvid, Ogg Theora, x264 and just about any other codec/format. There is a relatively new application available called MediaCoder that has proved to be very usable and can transcode from just about any codec/format to any other.

MediaCoder for Xvid and Ogg

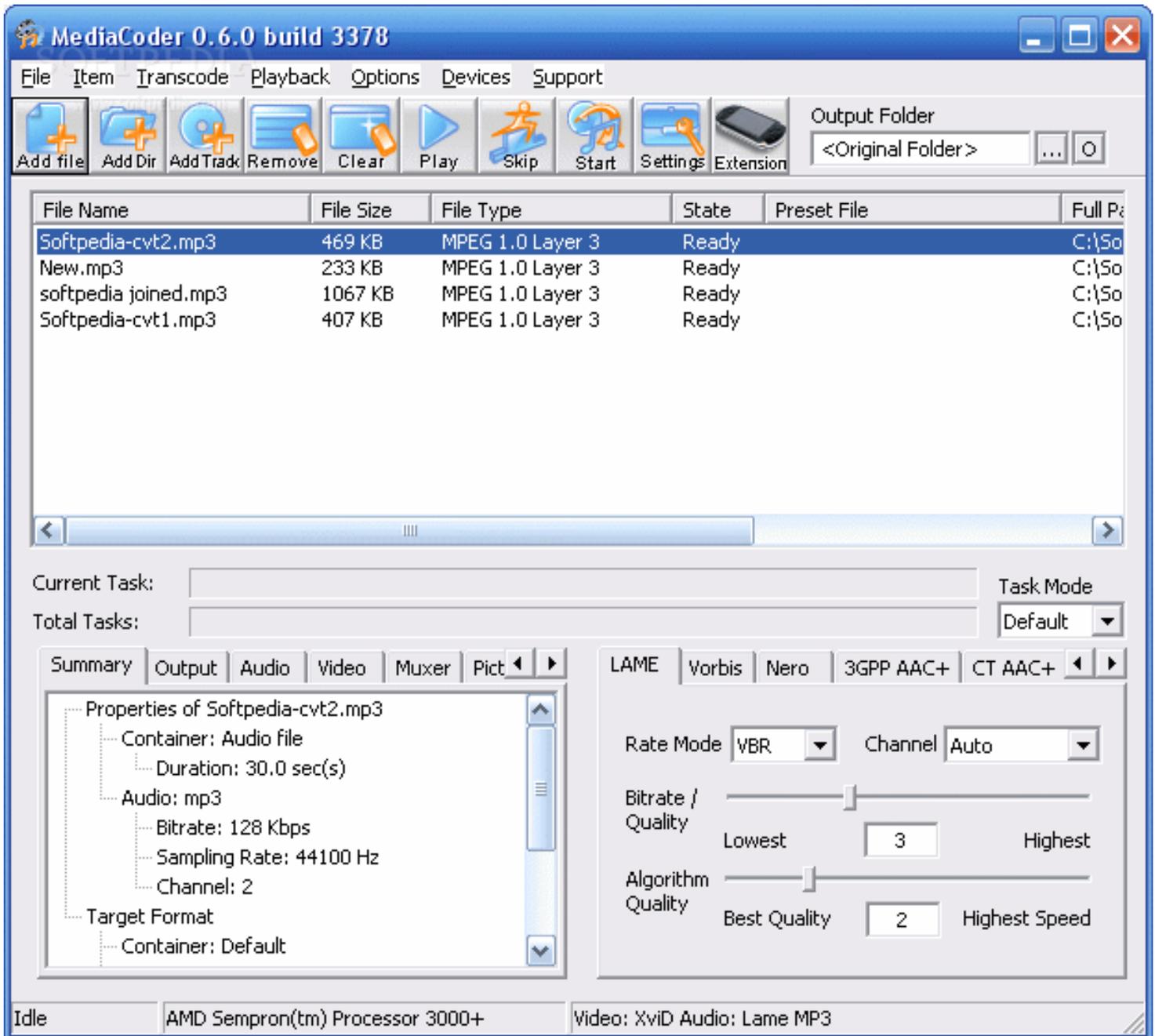
We recommend MediaCoder as the most user-friendly and stable option for transcoding video to Theora/Vorbis and Xvid/LAME for Windows operating systems. From the website: "MediaCoder is a free universal audio/video batch transcoder distributed under GPL license, which nicely integrates lots of excellent audio/video codecs and tools from the open source community into an all-in-one solution, capable of transcoding among different audio/video formats. With many extra features and a expandable architecture, MediaCoder is more than a GUI of a bunch of command line tools."

MediaCoder comes with good presets for transcoding, one of the best features for ease of use.

The user guide on the MediaCoder site is incomplete. It is recommended that this gap be filled during the Transmission Documentation project.

Features In Brief

- Convert to and from many audio and video compression formats and re-multiplex into various container formats, on the fly, in batch
- Give you the control over the transcoding process and all the parameters of the transcoded files, so you can learn about, compare among and play with various codecs
- Specific user interfaces for specific devices with device plug-ins and extensions
- Fully standalone, not dependent on any media player, filter or codec, no registration of any components into system
- Extensions to expand custom features and user interfaces



Screenshot of MediaCoder Interface

MediaCoder Downloads and Documentation



Download: <http://mediacoder.sourceforge.net/download.htm>



Documentation: http://mediacoder.sourceforge.net/wiki/index.php/Main_Page#Documentation



Development: http://sourceforge.net/svn/?group_id=151884

Mac

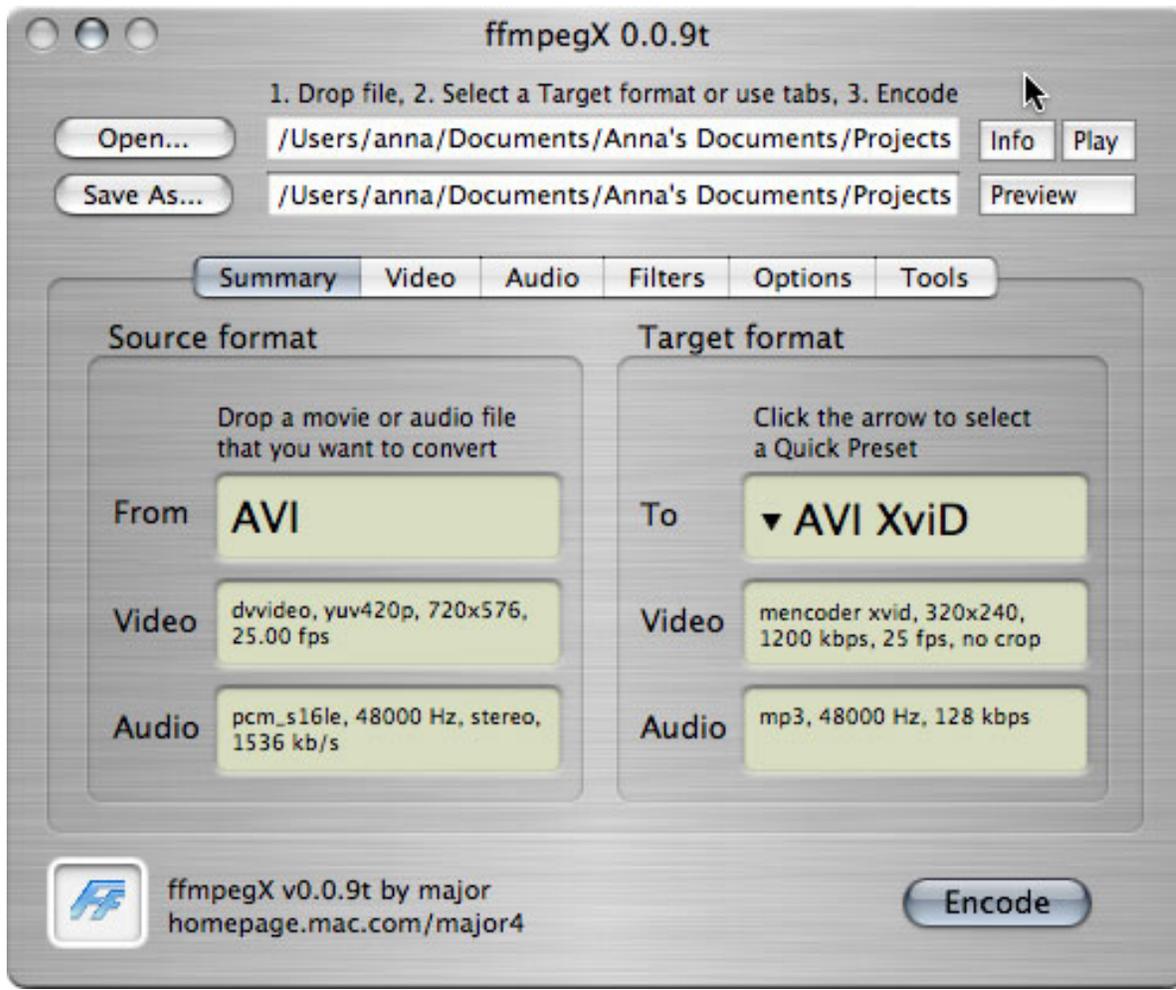
There are two recommendations here - one to use ffmpegX produce Xvid/Lame and one to use Simple Theora Encoder for Theora/Vorbis.

Talk about the use of XiphQT for transcoding and exporting direct from FinalCut etc.

ffmpegX for Xvid

ffmpegX is the frontrunner in the field for FOSS video transcoders. It is not a FOSS application itself, it is a Shareware GUI (\$15 registration) for the FOSS ffmpeg and mencoder transcoders and can transcode to Xvid/LAME(mp3) files amongst others. When Xvid is selected a preset is activated that only requires a small number of changes to create a decent output.

Documentation for making Xvid files with ffmpegX is scarce. EngageMedia will add this guide to the EngageMedia.org website and contribute this to the Transmission Documentation project if appropriate as part of the research. In the meantime there is a certain amount of documentation in the ffmpegX How-Tos.



Screenshot of ffmpegX interface

ffmpegX Downloads and Documentation



Download: <http://www.ffmpegx.com/download.html>



Documentation: <http://www.ffmpegx.com/howtos.html#divx>

Simple Theora Encoder for Ogg

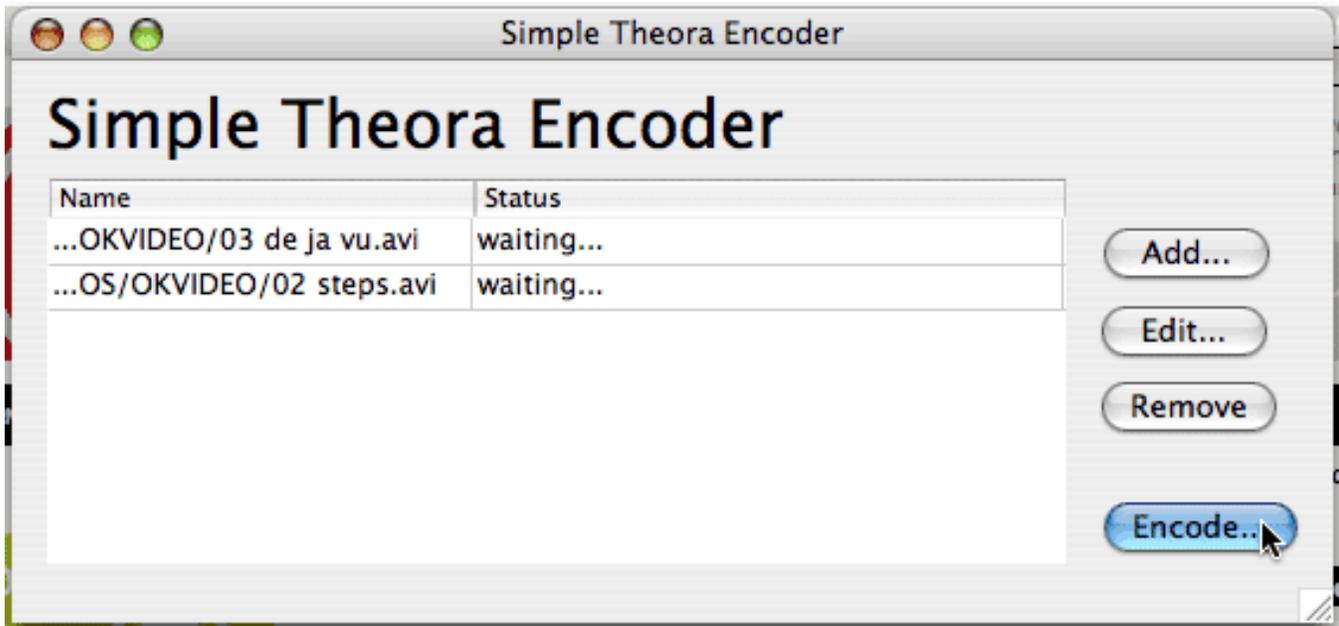
v2v, the developers of ffmpeg2theora, have recently built a graphical user interface for ffmpeg2theora that means users no longer have to use the command line. This application is still under development, but is able to be used now by people using the Mac OS X operating system on an Intel Mac. It has been built by the developer of ffmpeg2theora, Jan Gerber, and is called the Simple Theora Encoder.

Documentation was not available, so was completed as part of this research. EngageMedia have produced a

[Simple Theora Encoder tutorial](#) with screenshots to take people through how to download, install and use the application, which is very simple and usable. This will be contributed to the Transmission Documentation project and has been incorporated by v2v on the Simple Theora Encoder project page.

The beauty of Simple Theora Encoder is its simplicity. It is very easy to use and does not require the user to add any encoding settings. The transcoder is fast and outputs a high quality video.

It is recommended that support be given to J to make the application available for other operating systems in a stable version as soon as possible.



Screenshot of Simple Theora Encoder interface

Simple Theora Encoders Downloads, Documentation and Development

 Download: <http://v2v.cc/~j/SimpleTheoraEncoder/Simple%20Theora%20Encoder.dmg>

 Documentation: <http://v2v.cc/~j/SimpleTheoraEncoder/>

 Development: <http://svn.xiph.org/trunk/ffmpeg2theora/frontend/>

Linux

Command line interfaces are available for MEncoder, Transcode, ffmpeg and ffmpeg2theora. Many Linux users will feel comfortable using these applications, however only GUIs are considered here as we wish to focus on the needs of video makers without much computer experience. As more people move to Linux, a higher percentage of Linux users will fall into this category.

AcidRip is a GUI based DVD ripping/transcoding application written in perl. It is basically a wrapper for MPlayer and MEncoder. It is available for installation using Synaptic in Ubuntu and other package managers on other brands of Linux. If you would prefer to build it yourself, you will need Mplayer and MEncoder, lsdvd and acidrip. Handbrake is another DVD-ripper that can be used for transcoding to Xvid from DVD, it is a command line interface on Linux, but a very easy to use GUI on other platforms.

Shazam is a graphical user interface for ffmpeg and ffmpeg2theora which will create Ogg Theora/Vorbis files, developed by Estudiolivres. The user loads a video file into the first interface pane, then follows a series of wizard dialog boxes in Portuguese.

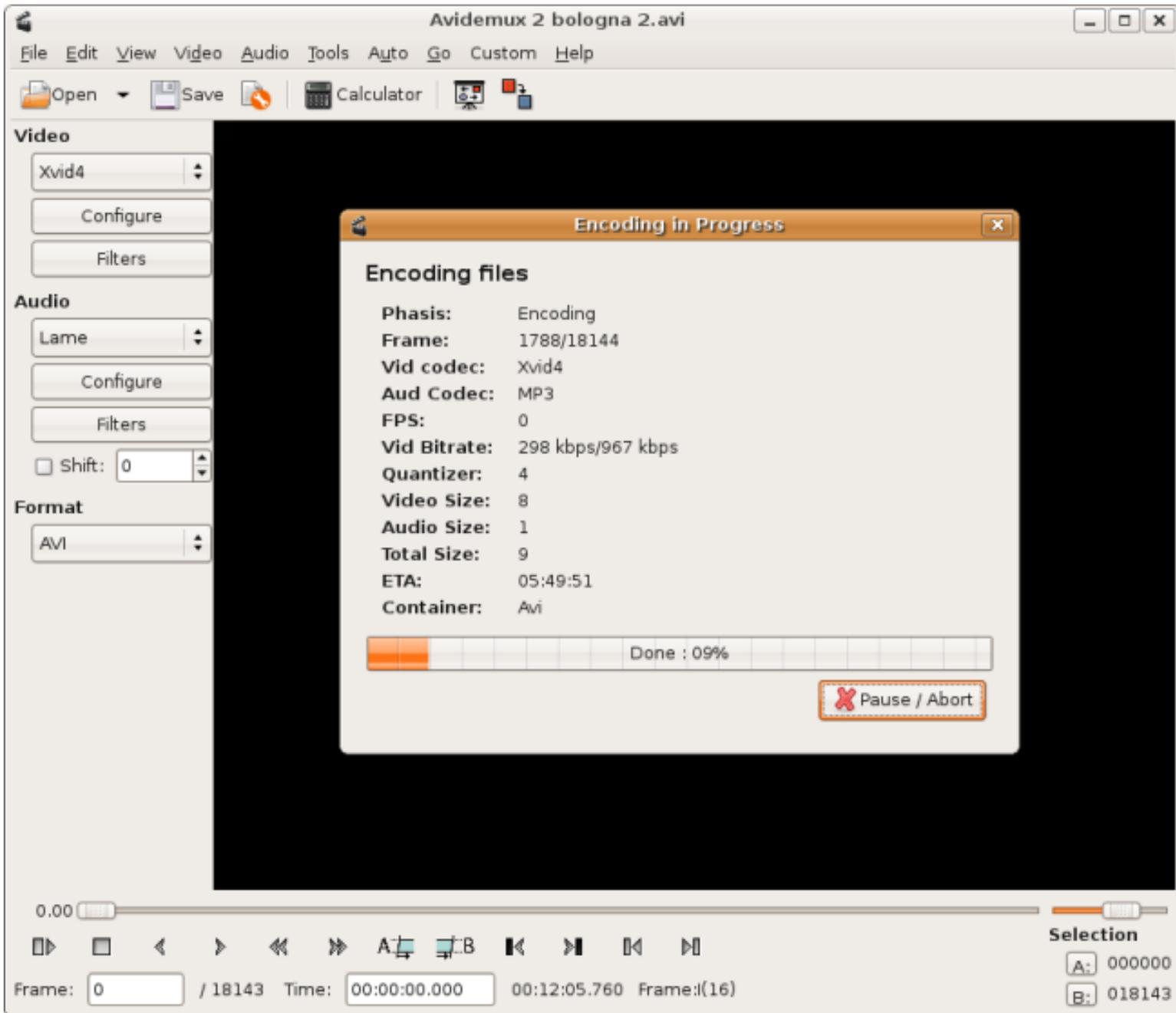
GTranscode is a graphical user interface for the Transcode command line utility for Linux that can transcode, apply filters, do re-sampling, resizing, cropping, and more. You can use it to create Xvid files amongst many others. Documentation is available [here on the FLOSSManuals website](#).

Avidemux for Xvid

An intuitive yet complete application for cutting, resizing, effecting and transcoding is Avidemux. It is not just an interface for transcoding, it has other features such as analog video capture, scripting and a joblist (a list for batch encoding, meaning the user can load up many files to transcode while they are doing something else).

Avidemux is multi-platform but developed primarily for Linux. It uses borrowed code from projects like ffmpeg and MEncoder - in particular a version of the libavcodec library from ffmpeg.

Avidemux was tested for Xvid/LAME transcoding and performed well using the presets for both Xvid (variable bit rate) and LAME (constant bit rate 128mbits/sec). There were a range of filters available to deinterlace the input file which will come in handy for videomakers transcoding from DV master files outputted from their editing applications. There were only a few easy steps involved in opening a file to transcode, selecting Xvid and selecting LAME, and clicking SAVE to start the transcoding process.



Screenshot of Avidemux interface

Avidemux Downloads, Documentation and Development



Download: <http://fixounet.free.fr/avidemux/download.html>



Documentation: http://www.avidemux.org/admWiki/index.php?title=Main_Page



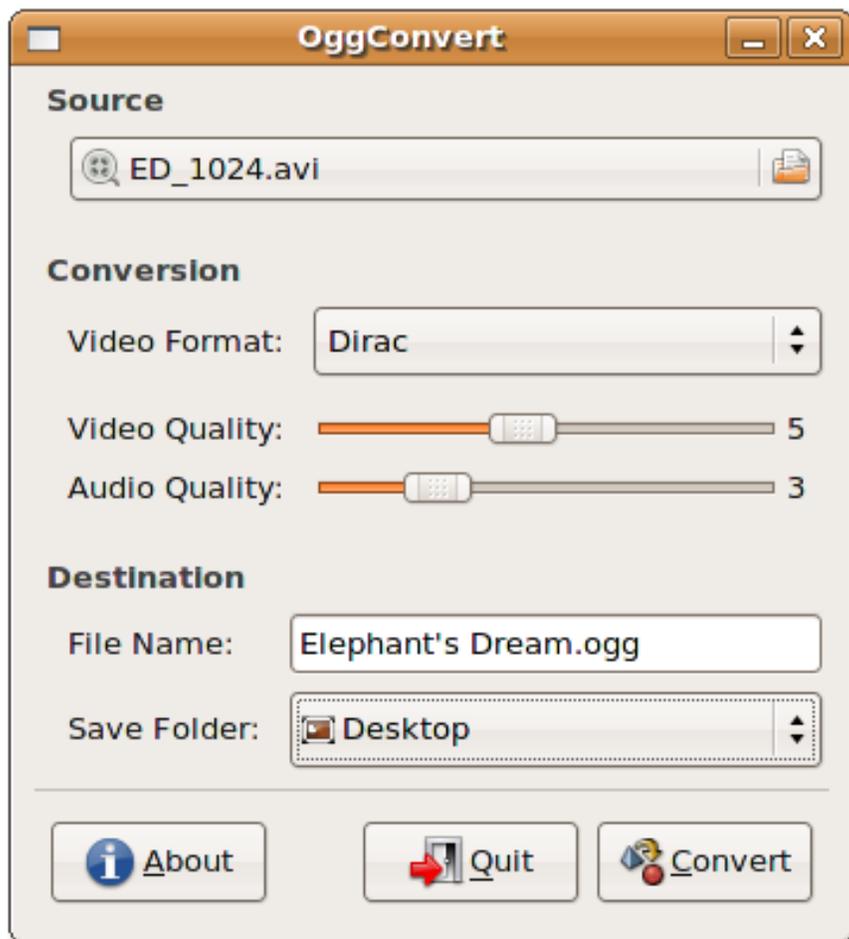
Development: <http://bugs.avidemux.org/>

Ogg Convert for Ogg

OggConvert is a small Gnome utility which uses GStreamer to convert (almost) any media file to the patent-free Ogg Vorbis, Theora and Dirac formats. Features (from the website) include:

- It's painfully easy to use: drag a file onto the source bar (or use the file chooser) and hit convert. Of course, you can also change the video format, quality settings and the output filename if you like
- It uses GStreamer, so it can convert (almost) any file which Totem can play
- It can deal with audio-only files, video-only files, and files with many audio tracks (such as DVD rips with a commentary track).
- Thanks to the magic of GStreamer, metadata (for example, title and artist info on an MP3) is preserved

Documentation is very limited. A tutorial should be created and added to the Transmission Documentation project.



Screenshot of Ogg Convert interface

Ogg Convert Download, Documentation and Development



Download: <http://oggconvert.tristanb.net/download/>



Documentation: <http://oggconvert.tristanb.net/about/>



Development: <https://launchpad.net/oggconvert>

Cross-Platform

There are two applications that are available for transcoding on Mac, Linux and Windows. Avidemux will transcode to Xvid but not Ogg Theora/Vorbis but has only been tested as part of this research on Linux. VLC is not just a media player, but has the ability to stream media and to transcode it to Ogg Theora/Vorbis not Xvid (though it offers the Xvid equivalents of DivX and Mpeg4 video). The advantage of using VLC is that it is cross-platform, and one documentation base can be shared amongst users of different operating systems across all Transmission member projects.

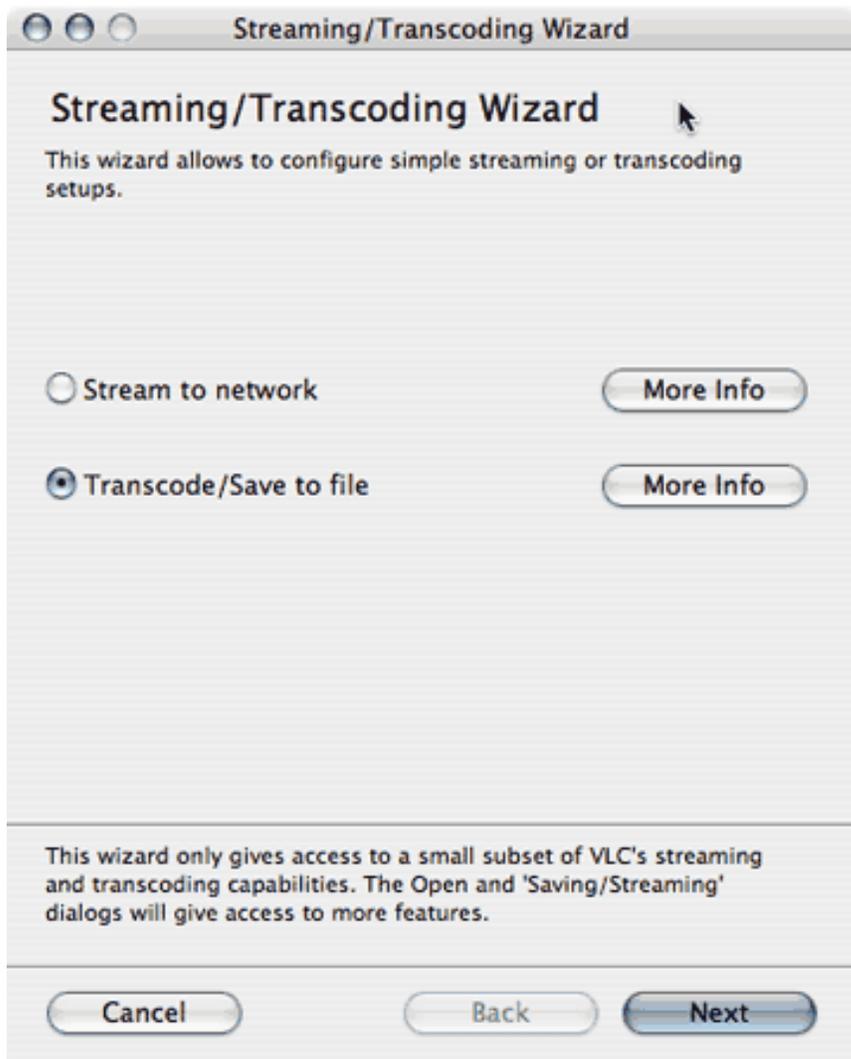
Avidemux for Xvid

See above Linux Transcoders section for details.

VLC for Ogg

VLC is not just a media player but can stream audio and video and transcode to MPEG4 and Ogg Theora/Vorbis also. Transcoding with VLC involves a few easy steps. A How-To transcode to Ogg can be found [here on YouMakeMedia.com](#).

As stated earlier, VLC's advantages generally are that it is cross-platform, stable and very usable but during testing some errors did occur, which suggests that further development is needed to make VLC a truly reliable transcoder. As VLC already has a few other features such as playing and streaming video it is hard to tell if development will be focused in this area by VLC or if they will leave this to other applications to concentrate on.



Screenshot of VLC transcoding wizard

VLC Downloads, Documentation and Development

See section on [VLC](#) above.

Support in Proprietary Transcoders for FOSS Codecs

With decent GUIs available to transcode to both Xvid/LAME and Ogg Theora/Vorbis using FOSS applications it was felt unnecessary to do a major review of proprietary applications for transcoding to these formats. However it is worth mentioning the XiphQT and Xvid components for Quicktime and directshow filers for Windows applications - as these allow for exporting both FOSS formats from any application using the Quicktime architecture and many Windows programs.

This means that videomakers can export Xvid or Ogg directly from their editing applications - a common request from videomakers.

Ogg for Quicktime - XiphQT Component

The XiphQT component for Quicktime is GPL and has the ability to produce Ogg files and encode video and audio content into Theora and Vorbis, respectively. Save directly to Ogg from iMovie or other QuickTime-based application. XiphQT now comes in a single, statically-linked component bundle and doesn't require any additional frameworks, so you can simply drag'n'drop to install it.

Many videomakers use FinalCut to edit their video - it has become an industry standard for video production. The XiphQ component makes it possible to export Ogg video directly from the timeline without having the extra step of using a transcoding application.

This component is still in development and users may encounter bugs, however it is stable enough to begin using now.

It is worth noting that in order to export from Quicktime you need to buy a license for Quicktime Pro (approximately \$25 USD).

XiphQt Downloads, Documentation and Development



Download: <http://xiph.org/quicktime/download.html>



Documentation: <http://xiph.org/quicktime/about.html>



Development: <https://trac.xiph.org/report/16>

Xvid for Quicktime

The [Xvid website](#) informs us that there is a Xvid quicktime plugin available for Mac only, but it is not well maintained at the moment. They are currently working on improved Quicktime support, which will become part of the Xvid core code base.

It is worth noting that in order to export from Quicktime you need to buy a license for Quicktime Pro (approximately \$25 USD).

Xvid Quicktime Plugin Downloads, Documentation and Development



Download: <http://n.ethz.ch/student/naegelic/download/>



Documentation: <http://n.ethz.ch/student/naegelic/faq/>



Development: <http://n.ethz.ch/student/naegelic/map/>

Ogg for Windows - Directshow

It is possible to export Ogg Theora/Vorbis directly from any program on Windows that uses the the DirectX9 SDK (currently only and. illi from the Illiminable project informed us that though it may be fairly straightforward to develop the ability for this filter to be used by video-editing programs such as Premiere and Vegas Video, this is unlikely to happen soon.

Illiminable Downloads, Documentation and Development



Download: <http://www.illiminable.com/ogg/>



Documentation: http://www.illiminable.com/ogg/about_sdk.html



Development: <http://www.illiminable.com/ogg/developers.html>

Xvid for Windows - Directshow

The Xvid codec for Windows allows any application that uses Directshow filters to transcode to Ogg, allowing for direct export from applications such as Premiere and Vegas Video.

Xvid Codec for Windows Downloads, Documentation and Development



Download: <http://www.digital-digest.com/software/download.php?sid=1052&ssid=0&did=1>



Documentation: <http://www.xvid.org/General-Info.19.0.html>



Development: <http://www.xvid.org/Xvid-Codec.2.0.html>

Server-Side Transcoding

What are the best available FOSS server side auto-transcoding applications to take the onus off video producers to encode to FOSS video formats?

There are often two versions of a video file that a Transmission network member may wish to create and distribute. The first is a relatively high-resolution version of a video that can be played full-screen on computer or projected at a community screening. This generally needs to be created on the user's computer, as the original video master will be in DV or HDV format and thus be too large to be transportable on the web. It needs to be compressed to this level in order to be uploaded. This can be referred to as "screening quality" video.

The second version of a video file is more useful for viewing on a computer as video on-demand or streaming video in the browser. This needs to be lower resolution, and more compressed so it can be transferred to the user's computer at an equal or faster rate than it is being viewed in order to watch it as it downloads or streams. This can be referred to as "web-viewing quality" video.

To save the user having to create two versions of a video to suit these two purposes, to make it easier for them to adopt FOSS solutions and to ensure that a video file designed for web-viewing quality can be standardised to be played back in a majority of browsers, it is better to transcode to web-viewing quality on the server-side.

There are various tools available to do this on your own server, and web-services available to do the transcoding for you - though these can cost money or have limited options available.

The method recommended here is to use mencoder, ffmpeg and ffmpeg2theora. These can be used in conjunction to transcode various file formats to both Xvid/lame and Ogg Theora/Vorbis - but for the purposes of browser playback Ogg is the only FOSS codec worth considering at this time. Flash video as explained earlier using FOSS tools is a good stop-gap measure at this point, and may be transcoded using mencoder.

MEncoder - Xvid/LAME

(From [Wikipedia](#)) *MEncoder is a free command line video decoding, encoding and filtering tool released under the GNU General Public License. It is a close sibling to MPlayer and can convert all the formats that MPlayer*

understands into a variety of compressed and uncompressed formats using different codecs. '

'Since it uses the same code as MPlayer, it also features the same huge number of highly-configurable video and audio filters to transform the video and audio stream: Filters include Cropping, Scaling, Vertical Flipping, horizontal mirroring, expanding to create letterboxes, rotating, brightness/contrast, changing the aspect ratio of the video's pixels, colorspace conversion, hue/saturation, color-specific Gamma correction, filters for reducing the visibility of compression artifacts caused by MPEG compression (deblocking, deringing), automatic brightness/contrast enhancement (autolevel), sharpness/blur, denoising filters, several different ways of deinterlacing, and reversing telecine.

MEncoder uses Ffmpeg - a set of software libraries that can record, convert and stream digital audio and video in numerous formats. It includes libavcodec, an audio/video codec library used by several other projects, and libavformat, an audio/video container mux and demux library. Most FFmpeg developers are also part of the MPlayer and VideoLAN projects, and FFmpeg is hosted at the MPlayer project server. Input from Jaromil during this research highlighted the need for ongoing support for these projects, which are the key engines that drive many applications listed as part of this research.

Fairly simple command-line arguments are required to transcode to Xvid/LAME, and can be passed to MEncoder from your online video platform. This is an example of a command to encode to Xvid/LAME:

```
mencoder <filename.avi> -ovc xvid -oac mp3lame -o <output.avi>
```

MEncoder Documentation and Downloads



Download: <http://www.mplayerhq.hu/>



Documentation: <http://www.mplayerhq.hu/DOCS/HTML/en/encoding-guide.html>



Development: <http://svn.mplayerhq.hu/>

ffmpeg2theora - Ogg Theora/Vorbis

ffmpeg2theora is a command line interface that uses FFmpeg to decode video and libtheora to encode it, written

by Jan Gerber from [v2v](#). It has many options including deinterlacing, resolution, contrast, gamma, brightness, saturation and the ability to set video and audio quality and bitrate. It can also insert metadata into the Ogg file, stream Ogg to Icecast for encoding on-the-fly and grab DV streams encoding them live to Ogg from a video camera. There is a debian package available for ffmpeg2theora [here](#).

Commands for usage of ffmpeg2theora are fairly simple to use. A copy of the ffmpeg2theora man pages are here on this wiki: [ffmpeg2theora_manpages](#). This is an example of a command to encode to Ogg Theora/Vorbis:

```
ffmpeg2theora --audiobitrate 128 --videobitrate 1000 --width 320 --height 240 --deinterlace <input.mov>
```

ffmpeg2theora Downloads, Documentation and Development



Download: <http://v2v.cc/~j/ffmpeg2theora/download.html>



Documentation: http://en.wikibooks.org/wiki/Ffmpeg_to_theora



Development: [ffmpeg2theora on Xiph Tracker](#)

Transcoding Web Services

[Mux](#) is a commercial web-service developed by [Cruxy](#) that offers transcoding to various formats including Ogg Theora/Vorbis by visiting the site and browsing for a file to transcode. It is designed to be an online "any-to-any" video conversion service, based on [Amazon Web Services](#). It is free for non-commercial individual use, but would be possible for Transmission members to pay to transcode using Mux for a fee. It's obviously better for Transmission members to run the same FOSS applications on their own servers without cost, but this may be a good option for some who don't wish to do their own server-side transcoding.

Transcoder/Uploader

How can we make it as easy as possible for a user to transcode to FOSS formats then upload it to the video platform? Conduct research into the development of a desktop transcoder/uploader application that could work locally to encode the file and then upload it to a specified site.

Many web applications that deal with multimedia offer uploader clients to make adding content to the website easier. FlickrUploadr is one example of this - a user can download and install this program, enter their username and password for their Flickr account and then browse for photos on their computer to upload. They can add metadata to each photo - titles, tags and descriptions - and then click to upload them all as a batch. Thumbnails are created automatically during the process. After clicking upload and waiting for the files to transfer the user can visit the website to see all of their content published on Flickr.

The need for a program like this to upload video is even greater for two reasons. One is the problem with uploading large files via a web form (over http) - if the user's internet connection is disrupted for any reason then the upload fails and must be started all over again.

Secondly - users don't only have to attach their video but transcode it first before they can upload it. Video that originates on DV or HDV tape, or even DVD is far too large to be transferred over the internet, so it must be compressed. We can eliminate this extra step, or make it largely transparent to the user by combining the ability to upload with the ability to transcode to a format and preset determined by the target website. For the user this just means browsing for their file, adding some info and uploading - they don't even have to think about the transcoding part.

Third parties offer clients like this for uploading video to YouTube such as the free uploader by [DVDVIDEOSOFT](#), and the CC Publisher is available to upload video to the [Internet Archive](#). It is important that Transmission members begin to use software such as this to make uploading video to their sites much easier.

EngageMedia looked into the possibilities of creating this kind of application, specifying its functional requirements and user experience, prior to discovering the development of such an application by Jan Gerber at v2v. It seems the best way to proceed is to contribute to and support the great work done here already by Jan.

v2v / Transmission Uploader

A key recommendation of this report is to contribute to the development of the v2v transcoder/uploader. This application joins the ability for users to encode to a screening-quality version of Ogg Theora/Vorbis simply with the ability to upload the resultant video via BitTorrent protocols to allow the resumable upload of larger files than is currently the case over http.

The v2v uploader client developed by Jan at v2v uses the BitTorrent protocol to upload Ogg Theora/Vorbis

files to a v2v video server. This has advantages in that the BitTorrent protocol is a resumable, fault-tolerant distribution mechanism, compared to a standard web form upload process via HTTP. This desktop GUI application (written in python, and a cross-platform GUI library) allows users to browse for non-ogg videos files, transcode them on their own computer using ffmpeg2theora into ogg, then follows the same process as above, that is, a BitTorrent protocol upload.

The uploading process could be extended to allow publishing into certain CMSs eg Plone, Drupal or Wordpress or others - via the use of a configuration panel. Each Transmission member would have the option of implementing their own configuration into the uploader easily and re-distributing it branded as their own uploader, or to release a configuration file that can be inserted into the application in order to customise it to their own CMS. The configuration file could be packaged with the application itself, to be initialised during the installation process.

It is recommended that documentation and promotion of this application be generated as it has great potential to make FOSS codecs more usable by the average person, and will also help to solve the problems of uploading large files via http to Transmission websites, which is the usual method at this stage. It is also recommended that a framework for collaborative development is established through use of a development tracker (such as that currently used by v2v).

Another recommendation is that a collaboration with the [CC Publisher](#) (an uploader for [archive.org](#) - the Internet Archive) is initiated, whereby joint development can ensure a wider uptake and increased developer-base of this application. The CC Publisher does not currently transcode, nor does it upload via bittorrent, so there are advantages to CC in working with Transmission on this project.

Functionality - Details

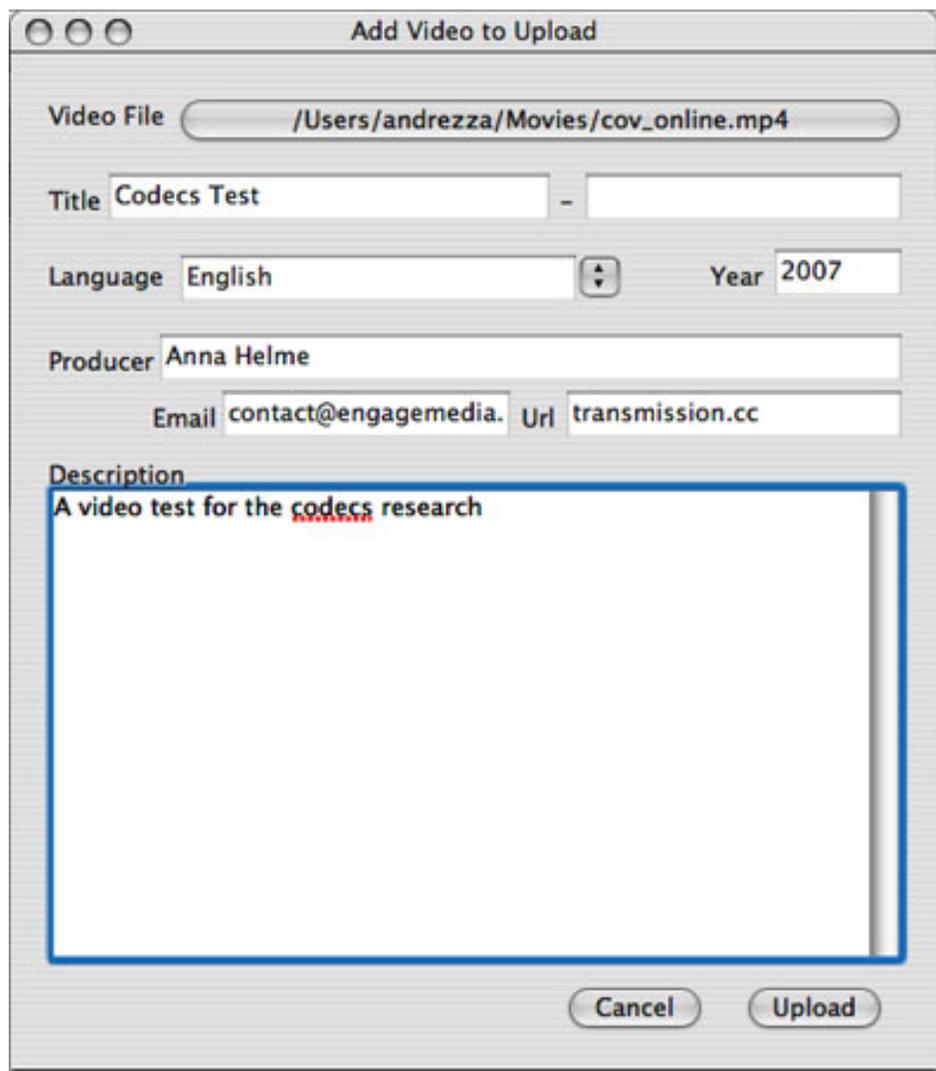
The client must have these functions:

- transcode to one Ogg Theora/Vorbis preset (as defined in configuration file)
- ability to add metadata/licenses conforming to the Transmission Metadata Standard to create an entry in the target CMS that does not also need to be updated via the web (create a one-step process)
- ability to be configured for target CMS - server settings, target folder, username/password - via a configuration file
- upload via a resumable protocol so upload may continue once network is reconnected after dropout
- cross-platform - available for easy install on Linux, Mac and Windows

User Experience

This user experience is described for the member of a website belonging to a project that is part of the Transmission network, which has implemented the Transmission Uploader.

- visit Transmission project website
- download Transmission Uploader client via a link on website
- open Transmission Uploader
- browse for video file
- enter metadata as per Transmission Metadata Standard - Title, Description, Tags etc.
- browse for thumbnail to attach
- click Upload button
- enter username and password for website membership
- video will transcode to Ogg Theora/Vorbis preset - user sees "Transcoding" message with status bar
- video will upload to website via BitTorrent - user sees "Uploading" message with status bar
- process finishes with "Upload Successful" message
- user's default browser is called, to a page on project website showing Video page - as the object or entry has been successfully added to the database



The screenshot shows a window titled "Add Video to Upload" with the following fields and controls:

- Video File:** A text field containing the path `/Users/andrezza/Movies/cov_online.mp4`.
- Title:** A text field containing "Codecs Test" followed by a hyphen and an empty field.
- Language:** A dropdown menu set to "English".
- Year:** A text field containing "2007".
- Producer:** A text field containing "Anna Helme".
- Email:** A text field containing "contact@engagemedia".
- Url:** A text field containing "transmission.cc".
- Description:** A large text area containing the text "A video test for the codecs research".
- Buttons:** "Cancel" and "Upload" buttons at the bottom.

Screenshot of v2v upload

v2v Upload Downloads and Development



Download: <http://v2v.cc/~j/V2VUpload.dmg>



Development: <http://trac.v2v.cc/cgi-bin/trac.cgi/wiki>

Browser Playback

What software is available for the playback of FOSS video within the browser? What is the best way to go amongst the various options given changes in technology, and what further development is required to make these solutions more viable?

Here we consider the methods available to playback video in the browser that is "progressively download"[2] or "pseudo-streamed" (video that you can play back while it is downloading via http) in addition to streaming[3] via true streaming protocols, as pertains to Ogg and Flash using the CMS-agnostic solution Indytube only. Though live streaming was not really considered part of this research, streaming protocols have their advantages for viewing video-on-demand on the web. Progressive download does not allow for scrubbing through video files (moving the video playhead as you watch the video to allow random access to different sections) and videos must be downloaded fully before you can view the entirety of the piece.

It is also possible to use the Darwin Streaming server to stream Xvid files but this has not been explored as Ogg seems the better route to take given its use of patent-free technologies, and Flash is a good method to use with open source tools as an alternative, especially while Ogg browser playback is being developed to a more stable level.

We covered the various methods available to handle playback of Ogg Theora/Vorbis including browser plugins, the Cortado Java applet and experimental support for native video playback in the browser itself. We also did some experimental development to extend Indytube to transcode to Ogg, embed Ogg using Cortado and implemented streaming of both Flash and Ogg.

Flash Pseudo-Streaming and Browser Playback

The secondary recommendation (which will be explained first to cover Indytube as it stands before looking into its extension for Ogg) is to use the FOSS tools that are presently available to transcode and embed .flv Flash video files. Flash is obviously a proprietary format (utilising a range of proprietary codecs within it) so fits

somewhat uncomfortably within this research.

However Flash has the highest level of support amongst web users for any video format, and this cannot be ignored when trying to reach web audiences. This in addition to the fact that stable and transportable code is available to knit FOSS transcoding and embedding software together in the Indytube package means that this is still a good short to medium-term measure for browser playback of video.

Indytube and OS Flash

Indytube is designed to be a simple CMS-agnostic solution for in-browser video playback with asynchronous encoder(s). This is a basic script for reencoding arbitrary video files to flash video (flv), and then including this video in the browser with a flash video player. It is written in Python and requires the installation of Mencoder to do the transcoding, FLVTool2 to write some metadata into the Flash video file, the Cheetah template library for Python and uses a Twisted networking framework to scan a directory files for new videos to encode. It uses the free Flowplayer flash video player to display and control video on the webpage.

Indytube (written by John Duda and expanded by Andy Nicholson of EngageMedia) offers to a Transmission project the ability to asynchronously transcode video that has been uploaded by users into Flash, while retaining the original file to offer for download. To the user it offers a YouTube-like experience - progressive-download video in a neat embedded video player. The user can click the video to start playing, control it by seeking, stopping and pausing, control volume, see how many minutes and seconds have played and are left to play, watch the video full-screen at the click of a button and also copy and paste an <object> or <embed> tag to paste the video player into their blog - just like with a YouTube or Google Video. The user experience is fully up to the standard that users would expect from a commercial service.

Indytube is also worth recommending even though it only supports Flash currently, as Ogg support has been experimentally added by EngageMedia as part of this research project, and has the potential to be available in a stable release soon. Transmission members may install Indytube knowing that a usable solution is available now for Flash, and Ogg will follow shortly. There is more info on this below.

It is a key recommendation of this report that Indytube is adopted as a CMS-agnostic solution for server-side transcoding and browser playback of video and implemented into Drupal (FilmForge), Plone (Plumi) and Wordpress (ShowInABox) and other hand-built CMSs. This is one way that Transmission members can develop on shared codebases, despite choosing or building our own CMSs based on our own needs or circumstances.

It is worth noting that the [project page](#) at Code Coop does not reflect the level of development carried out thus far, and needs additions in terms of documentation.

CAMP BETTY PROCESSION 

by **Anna Helme** — last modified 2007-06-14 15:39

GENRE: DOCUMENTARY

TOPIC: ACTIVISM GENDER / SEXUALITY

COUNTRY: AUSTRALIA

A clip of the procession for Camp Betty... "PROCESSION wishes to dislocate commonplace demonstration clichés by experiencing them in a different way. This will embed new meaning and create new traditions. The focus of PROCESSION will be a carefully-considered chain of ideas stemming from the workshop, that looking back, will relate to sex and gender."

Home Page: <http://www.campbetty.net>
Produced: 2007/06/15
Duration: 6 minutes 51 seconds
File Size: 62.8 MB



00:08/06:51

Copy the following to embed the movie into another web page:

```
<object type="application/x-shockwave-flash" data="http://prod-flv.engagemedia.org/FlowPlayer.sw" data-bbox="418 588 885 631">
```

 **DOWNLOAD VIDEO**

Screenshot of Indytube at work on the EngageMedia.org website

Indytube Downloads, Documentation and Development



Download: http://www.codecoop.org/scm/?group_id=90



Documentation: <http://www.codecoop.org/projects/indytube/>



Development: <http://www.codecoop.org/scm/viewvc.php/?root=indytube>

Indytube Extension for Flash Streaming - Red5

The FOSS Red5 streaming server can be used to stream Flash Video files. This was also implemented experimentally by EngageMedia during the research project and can be made available as part of Indytube relatively easy (all code is currently available separately and may be set up individually by developers as they require it).

Red5 is an Open Source Flash Server is written in Java and supports streaming audio/video (FLV and MP3), recording client streams (FLV only), shared objects, live stream publishing and remoting and is released under the LGPL. Red5 is hosted at the [OSFlash](#) project - like a Sourceforge for Flash-related open-source development.

Red5 Downloads, Documentation and Development



Download: http://osflash.org/red5#download_red5



Documentation: <http://osflash.org/red5/help>



Development: <http://osflash.org/red5/projectinfo>

Ogg Streaming and Embedded Playback

The primary recommendation is that Transmission members begin to utilise methods of playing back Ogg Theora/Vorbis in the browser rather than the proprietary Flash format. This is an area where much activity has been seen in the last few months, and things are still changing rapidly. There are three major ways that support for Ogg in the browser is being developed.

The first option we looked at was the Cortado Java applet. The main advantage of Cortado is that it will run on any system with a Java Runtime Environment (JRE) so has the capacity to work on most users' platforms.

Secondly we looked at browser plugins - the creation of a plugin that would work similarly to the way the Flash plugin decodes video within Indytube, or the way a Quicktime plugin works for Quicktime media in the

browser. There are four plugins in development to play back Ogg within the browser currently.

Thirdly we looked at the specification for HTML5 that is in development, and at experimental development by both Mozilla and Opera to include support for the HTML5 <video> element with baseline support for native decoding of Ogg Theora/Vorbis.

Lastly we looked at efforts to encompass all three of these options by using Javascript to invoke the appropriate playback method in the user's browser - which should increase coverage overall for Ogg support while leaving room for each of these initiatives to develop. Leaving all options open creates space for the best solution to emerge.

Cortado Java Applet

Cortado is a Java applet developed by Fluendo and released under the GPL that currently includes Java decoders for Ogg Theora, Ogg Vorbis, Mulaw audio, MJPEG and Fluendo's Smoke codec. It depends on the user having a Java Runtime Environment installed on their machine. JRE (Java Runtime Environment) is a software bundle from Sun Microsystems that allows a computer system to run a Java application. The JRE software bundle consists of the JVM (Java Virtual Machine) and programming interface (API). Applications can be written in Java for the JVM that can run on any machine using the same bytecode - and can thus run on any platform with a JRE installed.

It has been around for some time and works well - it's current disadvantage being the relative lack of features such as playback controls and fullscreen. These controls have been added by the iTheora project in PHP and there remains the ability to use Javascript or other methods to control the Ogg stream also, but this is yet to be explored.

It is recommended that development resources are contributed towards the improvement of the Cortado Java applet - specifically to update it for the use of the latest version of Theora and to improve the controls. Further testing should be done to assess the efficiency of Cortado in decoding video on various systems to ensure it's viability on older machines and across all platforms. During this research we found that as no JIT compiler is available for PPC (older Mac CPU architecture) Cortado runs sub-optimally on these systems.

Cortado Downloads, Documentation and Development



Download: <http://www.flumotion.net/jar/cortado/>

 Documentation: <https://core.fluendo.com/flumotion/trac/browser/cortado/trunk/README>

 Development: <https://core.fluendo.com/flumotion/trac/report/18>

vPIP and Cortado

vPIP stands for video Play-in-Place. It is Javascript that allows a thumbnail image to be swapped out for the appropriate plugin to play back the embedded video without having to reload the page. EngageMedia/Plumi had this installed to play back any uploaded format before transcoding to Flash using Indytube was available. The advantage of vPIP is that no transcoding is required. The disadvantage is that it is necessary for the user to have the appropriate plugin installed. Having to support any format with any plugin means that the user experience varies and no consistent support can be given as is the case when only one plugin/playback method is employed.

Some projects such as the ShowInABox video-blogging platform for Wordpress are using vPIP rather than any transcoding/playback solution such as Indytube. This is a simpler system that projects who don't wish to invest server or development resources into auto-transcoding may prefer.

A hack for adding Ogg to vPIP using the Cortado Java applet was added at the Big Apple Plone Sprint in 2006 by people from EngageMedia and the University of Louisville. It was forwarded to Enric from vPIP recently, who has included this support in the new release of vPIP. This means that Transmission members who use vPIP will be able to support Ogg playback in the browser.

There is a [Wordpress plugin for vPIP](#) to make it easy to use vPIP within your Wordpress site. There is also an easy installation for [Blogger](#) available.

vPIP Downloads, Documentation and Development

 Download: http://wiki.vpip.org/index.php?title=All_Other_Installs

 Documentation: <http://www.vpip.org/>



Development: <http://groups.google.com/group/vpip>

Browser Plugins

Browser plugins are available for many video formats in many browsers, for example Quicktime or WindowsMedia. It is relatively simple for a user to download and install a plugin if a decent installer package is created.

There are various plugins being developed to play back Ogg Theora/Vorbis files in the browser. Most are for Firefox, while one is for Internet Explorer. It is hard to make an accurate assessment of which of these efforts is worth supporting at this time, as they are all fairly experimental. With limited testing facilities we can only say that it seems that none of the options are entirely stable as yet.

It is important to note that Firefox is only in use by approximately 14.55% of internet users according to Wikipedia (statistics vary), though statistics for users of Transmission projects may be significantly higher, so these plugins are only applicable to part of our audiences.

Annodex Firefox Plugins

The Annodex project is developing open standards for annotating and indexing networked media. Originally the Annodex project was developed mostly within the CSIRO, Australia's largest government science research organisation. Over time there was increasing development of Annodex technology from the open source community, started with debian packages by Jamie Wilkinson, python bindings by Ben Leslie, and perl bindings by Angus Lees. In parallel, the CSIRO wanted to focus its work more on closed-source research and products built on top of the technology, losing interest in the open source and open standards components of it. Thus, it made sense to separate out the open parts into its own organisation and to create an Annodex Association similar in spirit to the many other foundations that have been created around other FOSS technologies.

The Annodex technology is a set of open specifications and open source software that allow the creation of Webs of audio and video integrated with the text-based view of the current Web. One of Annodex's major development initiatives is the Annodex Firefox plugin. The Annodex Plugins for Firefox are the best way to view Annodex media. The plugins install as embeddable video elements in web pages. Use these to view Annodex media in CmmlWiki, Annodex media from a mod-annodex enabled web server, or Ogg content anywhere on the web. The Annodex Plugins for Firefox provide a javascript API for movie control, seeking,

playlist, annotation retrieval, volume control, and buffer status. Callbacks are available for annotation and playlist events.

Annodex were originally working on an Internet Explorer ActiveX control but appear to have discontinued development of this plugin due to the relative ease of developing a cross-platform solution for viewing Annodex and Ogg media using Mozilla.

Annodex Plugin Downloads, Documentation and Development



Download: <http://www.annodex.net/software/plugin/index.html>



Documentation: <http://trac.annodex.net/wiki/AnnodexMozillaBrowserDevelopment>



Development: <http://trac.annodex.net/report/3>

MPlayer Plugin for Linux

A plugin has been developed for Mozilla browsers on *nix operating systems for the Mplayer media player. It requires that Mplayer is installed and supports playback of Ogg Theora/Vorbis.

MPlayer Plugin Downloads, Documentation and Development



Download: http://sourceforge.net/project/showfiles.php?group_id=71239



Documentation: <http://mplayerplug-in.sourceforge.net/faq.php>



Development: <http://sourceforge.net/projects/mplayerplug-in/>

VLC Firefox Plugins

Browser plugins are being developed at the VideoLan project for the VLC player. Testing of the VLC plugin was conducted on a PPC Mac running OS 10.3 during this research. The plugin was found to be buggy and crashed the browser and is not considered stable for this platform at this stage.

The VLC plugin is an option during installation of the package on Windows. If this option is not checked there are instructions for installing the plugin at a later date [here](#). The VLC plugin for Mac OS X can be downloaded [here](#) by scrolling down the page to either the Intel or PPC plugin release. Instructions for installing on Linux can be found [here](#).

The VLC plugin for Mac OSX can also be used in the Safari browser that comes with OS X.

Here is a [test page](#) for the playback of various mpeg files using the VLC plugin, and Javascript controls.

VLC Plugin Downloads, Documentation and Development



Download: <http://www.videolan.org/vlc/>



Documentation: <http://www.videolan.org/doc/play-howto/en/ch04.html#id293992>



Development: <us&component=Mozilla+plugin&order=priority> Videolan Tracker

OggPlay Firefox Plugins

[OggPlay](#) is a library designed to allow the playback of Xiph.org media developed by Shane Stephens (shans in #theora).

It has been implemented in the OggPlay Mozilla Plugin currently in development as well as being used in Chris Double's patch for native support of Ogg in Firefox. They have released details of the [Javascript API](#) which will be used to control video via the plugin.

The plugin is being developed as part of the Annodex project and is available experimentally for Linux. Instructions on how to compile it can be found [here](#).

This library should be differentiated from Symbian OggPlay which allows for playback of Vorbis files on Smart Phones.

OggPlay Plugin Downloads, Documentation and Development



Download: <http://svn.annodex.net/liboggplay/trunk/README>



Documentation: <http://wiki.xiph.org/index.php/OggPlay>



Development: <http://svn.annodex.net/liboggplay/trunk/>

Native Browser Playback of Ogg

There is currently a specification for HTML5 being developed by the WHATWG. The possibility of including a new <video> element in HTML5 is being proposed, as is the native support for Ogg Theora/Vorbis as a baseline video format by browsers. This is a major step in the progression of browser-playback of video, as users would not have to download any plugin or install any additional software to play back Ogg video in the pages of Transmission sites. It would also be a huge boost for the Ogg format and a big win for FOSS video in general.

It is unclear where these discussions will head, as there are many players in the browser market who have interests in maintaining support for proprietary video formats, as do the makers of the proprietary formats themselves.

It seems likely that both the <video> element and baseline support for decoding of Ogg Theora/Vorbis video will make it into the HTML5 specification. It is more unclear as to which browsers will actually support HTML5 or include the baseline Ogg support. Both Mozilla and Opera have said that they will include this, and have released experimental builds of their browsers with Ogg support. Apple have indicated they do not support Ogg - a fairly natural response as they continue to support H.264, however Safari holds a very small share of the browser market. It is more important to consider whether or not Internet Explorer will support HTML5, the <video> element and Ogg - there is no indication at present that it will.

Firefox continues to increase its market share, so the dominance of IE in controlling the direction of video within the browser faces a serious challenge. It is recommended that Transmission network members play a part in advocating for the support of Ogg Theora/Vorbis by browsers using HTML5 and inbuilt Ogg codecs. It is also recommended that we update our recommendations as things progress in this area.

Opera

Opera has an experimental version of Opera available as a binary download for Windows operating systems. Opera apparently support the efforts of the WHATWG group to have Ogg Theora/Vorbis support as a baseline in native browser support for the proposed <video> element, citing its advantages due to being patent-free. There is a video demonstration of native support for video in Opera on YouTube.

There is a good blog post here with details of the user's experience both testing out video support on the experimental build of Opera, comparing this to the performance of VLC on a low-medium powered machine and implementing the <video> element in HTML.

It is not recommended that Transmission projects use this experimental version of Opera with Ogg support at this stage, but that we look towards using this capacity in future.

Opera Experimental Build Downloads and Documentation



Download: (Windows only) http://people.opera.com/howcome/2007/video/op950_8762_video_repack_with_lgpl_signed.exe



Documentation: <http://my.opera.com/ResearchWizard/blog/experimental-opera-video-build-with-native-ogg-theora-support>

Mozilla

Chris Double is working on a patch for native support of the <video> element in Mozilla browsers, and hopes to have this included in Firefox3. Downloads of experimental builds of Firefox with the patch installed for Linux, Mac and Windows are available here. There is a demonstration of its use in conjunction with SVG technology here. The ticket for implementation of the <video> element as per the WHATWG spec is here on Bugzilla.

The Javascript API for controlling video decoded natively by Firefox is available here in the WHATWG HTML5 specification. There is limited support for this API in the experimental builds of Firefox with Ogg video support - play, pause, currentTime, videoHeight and videoWidth.

Once again, this is very interesting in terms of what we will be able to use in the future, rather than something we can use now. Below are details of how we can begin to future-proof ourselves for the use of the <video> element by using the MV_Embed script.

Firefox Experimental Build Downloads, Documentation and Development



Download: http://www.double.co.nz/video_test/



Documentation: <http://www.bluishcoder.co.nz/2007/08/firefox-video-element-patch-version-4.html>



Development: https://bugzilla.mozilla.org/show_bug.cgi?id=382267

Javascript Support for Multiple Playback Methods - Ogg

As there are obviously a few varied efforts being concurrently developed for Ogg Theora/Vorbis playback in the browser, a couple of solutions have been developed to check the user's browser for these methods and invoke the appropriate playback solution. This makes it easier to choose which method of playback to recommend, as it isn't really necessary to make the choice - merely provide support for all of them.

The simplest method is to check for <video> element support in the browser, and call the Cortado Java applet if this is not available. Both of these scripts support plugins to varying extent also, if they are found to be present.

It is recommended that Transmission members implement one of these options for viewing Ogg in the browser using most of the available methods, and that the Indytube code be adapted to use the MV_Embed script also (see below).

MV_Embed Script

The MV_Embed script developed by [Metavid](#) at the University of Santa Cruz is a method of testing the user's browser for native Ogg Theora/Vorbis support using the <video> element, and if this is not the case then employing the appropriate method of playing back the Ogg file - either a browser plugin or the Cortado Java Applet.

Native Ogg video support is being experimentally developed by Opera and Mozilla currently as is explained above. Plugins are only available experimentally for Mozilla currently, in early stages of development (see above).

The Cortado Java Applet will work on any system with a Java Runtime Environment installed, and is a good

solution for those without the VLC plugin or native Ogg support in their browser as explained above.

MV_Embed is the best compromise for supporting all of these efforts at present, in a way that is transparent to the user and has the best chance of finding a support method for Ogg streams in the browser.

From the Metavid website:

"The script works by creating a html element / javascript object that maps html 5 video tag (ed. the <video> element) javascript calls to the appropriate methods on clients supported playback system. Obviously not all of the html5 features will be supported for every playback method, but the idea is to build robust support for basic video embedding while letting web users immediately take advantage of simplicity of usage for embedding theora video.

Native (html5) browser support for ogg media is developing nicely...Chris Double is working on ogg video support for firefox3, experimental builds of Opera have been shown to support it, and the oggplay plugin is under very active development as a firefox plugin for the win, linux and osx platforms. Even an ActiveX version for IE is planned. The mv_embed script should detect any of these "native" support mechanism and do nothing. As these playback systems should more or less conform to the html5 spec."

MV_Embed Downloads, Documentation and Development



Download: http://metavid.ucsc.edu/wiki/index.php/Mv_embed



Documentation: <http://metavid.ucsc.edu/blog/2007/06/07/html5-video-the-future-is-now/>



Development: <http://slugforge.cse.ucsc.edu/gf/project/metavid/>

ArcticGiant

The ArcticGiant script is similar to the MV_Embed and is being used by WikiMedia for the WikiMedia player developed by [Greg Maxwell](#), which is in beta version now. A copy of the script written by MaikMerten can be found here on the wiki: [ArcticGiant](#)

Support for Ogg using an earlier version of this script can be found on Wikipedia. [Example](#)

Demo of latest ArcticGiant script

ArcticGiant Downloads



Download: [ArcticGiant](#)

Indytube and Ogg/Vorbis

It is recommended that Indytube is adapted to use methods of playback for Ogg, and that Indytube is adopted by Transmission members where possible to increase collaboration on CMS-agnostic solutions for embedded playback. Further development is needed here to test and implement the MV_Embed script for use within Indytube, and further testing of the MV_Embed script itself.

Indytube Extension for Ogg Transcoding - ffmpeg2theora

Support for transcoding using ffmpeg2theora to the Ogg preset defined by ffmpeg2theora was added experimentally to Indytube as part of this research. This is working successfully and Indytube now produces both an Ogg file and a Flash file.

See notes on ffmpeg2theora above.

Indytube Extension for Ogg Embedding - Cortado

During this research experimental support for the Cortado applet in Indytube was tested. This meant calling the Cortado applet rather than the Flash Flowplayer as part of a page template - used here in Plumi for testing. The test was successful, and an early version of support using Cortado was added to the Indytube code - though it is not enabled in the current release.

See notes on Cortado above.

Indytube Extension for Ogg Streaming - Icecast

Icecast streaming functionality has been added to the Indytube package experimentally as part of this research. Icecast was originally developed to stream Vorbis but is now equipped for video streaming using Theora as well.

It is also possible to stream Ogg Theora/Vorbis with Flumotion. The Flumotion streaming server was developed

by Fluendo using the GStreamer multimedia framework (C) and the Twisted networking framework (Python).

See notes on Icecast above.

Indytube Development for Ogg Embedding - MV_Embed

The recommendation for development here is to implement the MV_Embed script within Indytube. Currently Indytube has experimental support for Cortado (see above) but it would be better to adapt the MV_Embed script for use within Indytube to utilise both browser plugins and future native Ogg support in browsers as well as the Cortado java applet within Indytube.



Development: https://www.codecoop.org/tracker/index.php?func=detail&aid=945&group_id=90&atid=428

Support in Popular CMSs

What is needed within the most popular systems to enable the upload and playback of FOSS codecs?

Independent video community sites appear to be converging around popular content management systems such as Drupal, Wordpress and Plone. In this section we document the current state of support for Ogg within these systems and what would be necessary to increase that level of support.

Drupal

Drupal has basic support for uploading and embeddedding Ogg within the browser. Video uploading is enabled using the Drupal [video module](#) and embedded playback utilises the Cortado Java Applet. [Demo](#)

Some development is occurring around [YouTube style flash modules](#) however to date this does not include auto-transcoding to Ogg. Another recipe for YouTube style site based on Drupal can be found [here](#). The recipe here utilises ffmpeg to encode to flash. Ogg support is not available.

[OpenPackage Video](#) provides another option. It uses ffmpeg for encoding to Flash either locally or via the OpenPackage Video server at a cost of \$1/100megabytes of source video. Commercial use of the flash player OPV uses [requires you to purchase a license](#) in order to waive the Creative Commons non-commercial share-alike license.

In theory as OpenPackage Video and the Tuna Special recipes are based on ffmpeg adding support for Ogg may not be a huge stretch but would require deeper investigation. The current list of future features for OpenPackage Video does not include adding Ogg support. [OPV Drupal page](#)

Drupal Video Module Downloads, Documentation and Development



Download Video Module: <http://drupal.org/project/video>



Documentation: <http://drupal.org/handbook/modules/video>



Development: <http://drupal.org/project/issues/video?states=8,13,14>

Film Forge

The [FilmForge video CMS project](#) is a version of Drupal, tailored to the needs of videomaking communities. FilmForge (to be released in the 4th quarter of 07) makes it simple to install and run a video sharing site. FilmForge is currently considering adding [IndyTube](#) support to the package to enable Ogg support. To our knowledge Indytube is yet to be tested in Drupal however it would be relatively simple to enable.

Plone

Video support for Plone centres around two key products: [ATVideo](#) and [Plone4Artists Video](#). Plone4Artists Video is a newer initiative based on the Zope 3 architecture. It is less developed and currently lacks many of the features ATVideo offers but looks set to grow. At the time of writing it does not support Ogg transcoding or embedded playback however some work has been done on embedding Ogg video using Cortado and that work is set to continue.

Plumi

[Plumi](#) is a Free Software video sharing Content Management System based on Plone. The EngageMedia branch of ATVideo (which handles video in Plumi) currently offers embedded playback of Ogg at an alpha level using the Cortado applet. [IndyTube](#) is used to manage the transcoding.

Additionally as [vPIP](#) now has Ogg support and already works in Plumi Ogg embedding would be relatively simple to add for pre-transcoded files.

Plumi Downloads, Documentation and Development



ATVideo: <https://svn.plone.org/svn/collective/ATVideo/branches/engage/>



IndyTube documentation: <https://svn.codecoop.org/svn/indytube/README>



Plumi transcoding set up documentation <https://svn.plone.org/svn/collective/Plumi/tags/plumi-0.1-rc2/INSTALL.txt>



Development: <http://codecoop.org/projects/indytube/>

Joomla

Most Joomla video components/add-ons only seem to offer display of videos hosted on another service, or very limited video functionality. There are some commercial add-ons for media publishing however though many of the components listed don't appear to be maintained, or are ads of some sort for services/software.

The key video component appears to be [Seyret Video Component](#) however it does not offer any transcoding or Ogg functionality.

The possibility of implementing Cortado for playback of Ogg files does not seem to have been explored. Given the quality of the components on offer, Cortado support would probably have to be written from scratch.

As Joomla has limited support for video it may be some work to get it to a stage where FOSS codecs were viable.

WordPress

Most WordPress video plugins revolved around the embedding of Flash video and have no support for FOSS codecs and embedded playback. A listing of Wordpress video plugins can be found here <http://codex.wordpress.org/Plugins/Video>.

Show In A Box

Show In A Box is a specialised video blogging package for the Wordpress CMS. It works similarly to FilmForge and Plumi. Show-in-a-box has a more limited array of functionality, however that is in no way a disadvantage. For small scale projects requiring basic video sharing or video blogging features and easy installation it is quite apt.

On August 27 2007 (just a few days before the release of this report) Show In A Box added support for Ogg playback using vPIP | example. The Ogg files currently must be transcoded by hand. Auto-transcoding is not presently available however implementing IndyTube may be a relatively simple process. The Show In A Box community appears quite supportive of FOSS formats and so support and implementation is likely to grow.

Show In A Box Downloads, Documentation and Development



Download: <http://showinabox.tv/wordpress/download/>



Documentation: <http://showinabox.tv/wordpress/about/>



Development: <http://groups.google.com/group/vpip>

Future of Web Video

As mentioned above in the section on native browser support for Ogg there is currently a specification for HTML5 being developed by the WHATWG including the possibility of including a new <video> element in HTML5 with native support for Ogg Theora/Vorbis as a baseline video format by browsers. It is important that Transmission projects play a part in advocating for the support of Ogg Theora/Vorbis by browsers using HTML5 and inbuilt Ogg codecs. It is also recommended that we update the recommendations as things progress in this area.

Another important area to watch is that of mobile devices and which formats and codecs will be supported. Mobile devices often use hardware decoders rather than software decoders to playback video due to the relatively smaller capacity of the processor.

H.264 is being touted as the major format for mobile video, as well as being used for new generation HD DVD video discs, IPTV and for terrestrial and satellite delivered digital television broadcast. Apple has invested heavily in H.264 and this is the format that Apple TV, iPhones and the video iPod are designed to use.

Unfortunately, due to patents held in the USA, open source versions of H.264 such as x.264 are encumbered by potential legal problems (similarly to Xvid).

Ogg Theora/Vorbis does have the potential to be supported by mobile devices using the software decoder, but it remains to be seen what the degree of uptake of this will be. Further investigation and regular updates of shared information about the development of video for web and mobile devices need to be initiated by Transmission members through research projects such as this.

Where to From Here

This is the first draft of report and set of recommendations from this research project. The next steps are as follows:

- Submit draft to Transmission network via discussion list and website
- Request feedback from Transmission members regarding recommendations
- Aggregate input and complete final draft
- Submit final draft to Transmission network via discussion list and website
- Publish and promote final report on the web through Transmission network and beyond
- Track implementations of recommendations using the Transmission wiki and document to aid others in implementation. this can be completed in partnership with the Transmission Documentation project
- Present report to the FOMS Foundation of Open Media conference in Melbourne 2008
- Update recommendations after 6 months on Transmission wiki
- Ongoing development as outlined in [Summary of Recommendations](#)
- Continued engagement with Xiph, ffmpeg, mencoder and other major FOSS video compression projects

Regular updates to these recommendations are necessary given the rapidly changing nature of online video technology. In particular the browser support of Ogg files is poised to become more usable with updated version of Indytube.

A 6-month review of the report by EngageMedia and other contributing members of Transmission will ensure this discussion is kept alive in the network and that our members are able to advocate for the use of FOSS codecs and lead by example by implementing their use.

Implementation of the final draft of recommendations should increase the uptake of FOSS codecs within the Transmission network. Ongoing development of FOSS tools in this area by Transmission members including v2v and EngageMedia and hopefully others will continue to enhance the usability of FOSS codecs. Further support for funding of development will be sought to enable this process.

Glossary of Terms

[1]**Codec** refers to the compression algorithm that is used to make audio or video information smaller for transport over the web, on DVD or VCD or for broadcast. The codec needs to be available for the transcoder to compress the file initially, and available for the player to decompress or decode the file at the users end. Codecs can be either lossless (which uses mathematics to compress the video similarly to a Zip file and results in no loss of video information) or lossy (which additionally removes visual information which is imperceptible or less important to give a more or less accurate representation of the image over time, losing detail and accuracy as the compression ratio is increased).

[2] A **container format** encapsulates raw compressed audio and video data and allows the interleaving of audio and video data inside a single convenient format. Examples of multimedia container formats are Quicktime .mov, Ogg and AVI.

[3]**Progressive download** is the transfer of audio/video data via http as opposed to streaming protocols, that allows the user to begin to watch a file as it progressively downloads and before it has downloaded entirely. Usual transfer over http would require the video file to be downloaded in its entirety before being able to be watched from the beginning. Progressive download gives a more immediate experience to the user without requiring the use of a special streaming server.

[4]**Streaming** is the transfer of audio/video data in a stream of packets that are interpreted and rendered, in real time, by a software application as the packets arrive. The user may seek or scrub through the video and watch any part of the video immediately, given time to buffer the stream. Streaming is considered here in its on-demand form rather than as a live stream of events, as it is generally the usage of Transmission members.

Retrieved from "http://wiki.transmission.cc/index.php/Codecs_Draft_Recommendations"

-
- This page was last modified 10:42, 31 August 2007.
 - This page has been accessed 531 times.
 - Content is available under [GNU Free Documentation License 1.2](#).
 - [Privacy policy](#)
 - [About Transmission](#)
 - [Disclaimers](#)