****

|  |
| --- |
| Till Tönshoff  2/14/2009 |

|  |
| --- |
|  |
| RetroCode™  Documentation  Version 0.09 (GPL) |
|  |

Contents

[Copyright 4](#_Toc208040197)

[Author 4](#_Toc208040198)

[License 4](#_Toc208040199)

[Warning 4](#_Toc208040200)

[Description 5](#_Toc208040201)

[Features 5](#_Toc208040202)

[Project History 5](#_Toc208040203)

[Supported Platforms 5](#_Toc208040204)

[Contact and Support 5](#_Toc208040205)

[Compatibility 7](#_Toc208040206)

[Downloading 8](#_Toc208040207)

[Compilation 9](#_Toc208040208)

[Compilation on Windows 9](#_Toc208040209)

[Compilation on Unix / Linux / MacOSX 9](#_Toc208040210)

[Dependency 9](#_Toc208040211)

[Compiling avcodec, avutil and avformat libraries for Windows 12](#_Toc208040212)

[Install MSYS and MinGW 12](#_Toc208040213)

[Compile libfaac 13](#_Toc208040214)

[Compile libfaad2 14](#_Toc208040215)

[Compiling libmp3lame 15](#_Toc208040216)

[Compiling libogg 15](#_Toc208040217)

[Compiling libvorbis 15](#_Toc208040218)

[Compiling libamrnb 15](#_Toc208040219)

[Compiling libamrwb 15](#_Toc208040220)

[Compiling libmp4ff 15](#_Toc208040221)

[Compiling ffmpeg (avformat,avutil,avcodec) 16](#_Toc208040222)

[Installing 17](#_Toc208040223)

[Usage 19](#_Toc208040224)

[File Format Changes 19](#_Toc208040225)

[Sample Frequency Changes 20](#_Toc208040226)

[Supplying Meta Data 20](#_Toc208040227)

[Important Filter Functions 20](#_Toc208040228)

[Examples 20](#_Toc208040229)

[Syntax 22](#_Toc208040230)

[GENERAL PARAMETERS 22](#_Toc208040231)

[FORMAT SPECIFIC PARAMETERS 25](#_Toc208040232)

[Yamaha Synthetic Music Application Format (SMAF/MMF) 25](#_Toc208040233)

[MacroMedia Flash (SWF) 30](#_Toc208040234)

[Wave 31](#_Toc208040235)

[Beatnik Rich Music Format (RMF) 32](#_Toc208040236)

[Panasonic MxxFxxMxx (MFM) 35](#_Toc208040237)

[VOX 36](#_Toc208040238)

[AMR NB 37](#_Toc208040239)

[AMR WB 38](#_Toc208040240)

[Qualcomm Compact Multimedia Format (CMF/PMD) 39](#_Toc208040241)

[Advanced Audio Coding (AAC) 40](#_Toc208040242)

[MPEG-4 (MP4) 41](#_Toc208040243)

[3GPP File Format (3GP) 42](#_Toc208040244)

[Qualcomm PureVoice (QCELP/QCP) 43](#_Toc208040245)

[Audio Interchange File Format (AIFF) 44](#_Toc208040246)

[Windows Media Audio (WMA) 45](#_Toc208040247)

[Real Audio (RA) 46](#_Toc208040248)

[OGG 47](#_Toc208040249)

[AVI 48](#_Toc208040250)

[uLaw 49](#_Toc208040251)

[aLaw 50](#_Toc208040252)

[MPEG-1,2 Level 3 (MP3) 51](#_Toc208040253)

[Frequently Asked Questions 52](#_Toc208040254)

[How do I create a Wave-file compatible with a Siemens S65? 52](#_Toc208040255)

[How do I create a Wave-file compatible with a Sagem handset? 52](#_Toc208040256)

[Notes for Developers 53](#_Toc208040257)

[Basic function overview of the RetroCode commandline application 53](#_Toc208040258)

[Important CMobileSampleContent (codec) methods 53](#_Toc208040259)

[Developing new codecs 53](#_Toc208040260)

[CREDITS 54](#_Toc208040261)

[Author 54](#_Toc208040262)

[Contributors 54](#_Toc208040263)

[IP Holders 54](#_Toc208040264)

[Appendix A: GPL License Version 3 56](#_Toc208040265)

# Copyright

Copyright (C) 2004-2005, Retro Ringtones LLC, Copyright (C) 2006-2009, MMSGURU

# Author

Till Toenshoff  
Knauerstrasse 23  
20249 Hamburg  
Germany  
[me@mmsguru.com](mailto:me@mmsguru.com)

License  
This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Please see [Appendix A](#_Appendix_A:_GPL) of this document for a complete copy of the GNU General Public License Version 3.

Alternative licensing including fully licensed CODEC packages and customized versions are being offered by the author directly. Please consult the author [Till Tönshoff](mailto:me@mmsguru.com) at [me@mmsguru.com](mailto:me@mmsguru.com)

# Warning

Multimedia these days is a minefield. The author does not hold or claim the intellectual property of any of the used encoding/decoding algorithms. This software in its GPL-release has been created and released to the public solely for the purpose of evaluation, education and auditing. The author can and must not be held responsible for any claims/threads you might receive when using this version of RetroCode.

**Be aware that some of the used CODECs are based on fragmential specifications, released by their original IP holders. Namely Retro's RMF codec is not licensed or in any way approved by Beatnik Software. Please make sure that you read all mentioned documents below to find out if you have to pay additional licensing fees when rendering files using this software.**

# Description

RetroCode™ is a universal mobile sound sample content converter.

# Features

RetroCode™ reads and writes pretty much all common mobile sample content files. RetroCode™ takes all possible measures for making sure that meta-data is maintained and converted between the formats. It features filters for adapting audio content to the abilities of small handheld devices. RetroCode™ also includes a ringback-signal synthesizer allowing you to mix your audio content with standard ITU ringback signals.

# Project History

For some historical details on the RetroCode SDK development process, please see build/ChangeLog

**2004**  
The RetroCode™ development started in 2004. Those days it was developed only for allowing the producers of RetroRingtones LLC to maintain a high quality level by automizing and unifying as much of their dayly work as possible. The challenge they met with using all these scattered tools from the various format developers were hardly bearable.   
**2005**  
By the end of 2005, RetroCode™ was included into RetroFolio™ as a core component.  
**2006**  
By the end of 2006, the total ownership of RetroCode™ was transferred from Retro Ringtones LLC over to MMSGURU.  
**2007**  
In late 2007 the GPL version of RetroCode™ went public. Mac OS X port was added to the Windows and Linux versions.  
**2008**  
After some time of RetroCode being asleep for a few months, finally binary distributions for various platforms are being built. This is finally possible due to the fact that we got Autotools (autoconf, automake, libtool) to correctly process the build-process of the RetroCode SDK.  
**2009**  
Early 2009 the QCELP en/decoding was not relying on Qualcomm’s QSCL binaries anymore, making QCELP en/decoding fully portable.

# Supported Platforms

Microsoft Windows (NT, 2000, XP, Vista)  
Linux,  
Mac OS X (PPC, Intel)  
Let us know if you got the RetroCode SDK working on any other machine

# Contact and Support

Please use the RetroCode™ mailing list in case you have any problems, questions or doubts using RetroCode™.

For subscribing to that list, go to: <https://lists.sourceforge.net/lists/listinfo/retrocode-news>

Please use this list and its archive instead of contacting the author directly for these reasons;

1. Your question may already be answered in the list-archives.
2. You will be read by more users, giving you a higher chance of a quick and helpful answer.
3. The author is reading and answering within that list.

# Compatibility

RetroCode™ currently supports the following formats

|  |  |  |
| --- | --- | --- |
| **Format Name** | **Read / Decode** | **Write / Encode** |
| Beatnik RMF (uncompressed) | + | - |
| Beatnik RMF (ima adpcm) | + | + |
| Beatnik RMF (mpeg) | + | - |
| Yamaha MA2 | + | + |
| Yamaha MA3 | + | + |
| Yamaha MA5 | + | + |
| Yamaha MA7 | + | + |
| Panasonic MFM | + | + |
| Sagem Wave | + | + |
| Qualcomm QCELP | + | + |
| Qualcomm CMX | + | + |
| AMR NB | + | + |
| AMR WB | + | + |
| MPEG 2 Layer 1,2 and 3 | + | + |
| AAC | + | + |
| Macromedia Flash | + | + |
| AIFF | + | + |
| 3GPP | + | + |
| MP4 | + | + |
| uLaw | + | + |
| aLaw | + | + |
| Dialogic VOX | + | + |
| OGG | + | + |
| RealMedia Audio | (+) | (+) |
| Microsoft WMA | (+) | (+) |
| Microsoft AVI | + | - |
| Microsoft Wave | + | + |

# Downloading

For binary release-versions, go to <http://sourceforge.net/projects/retrocode>

For the latest development source snapshot, use CVS.

Anonymous CVS Access

This project's SourceForge.net CVS repository can be checked out through anonymous (pserver) CVS with the following instruction set. The module you wish to check out must be specified as the modulename. When prompted for a password for anonymous, simply press the Enter key. To determine the names of the modules created by this project, you may examine their CVS repository via the provided web-based CVS repository viewer.

cvs -d:pserver:anonymous@retrocode.cvs.sourceforge.net:/cvsroot/retrocode login

cvs -z3 -d:pserver:anonymous@retrocode.cvs.sourceforge.net:/cvsroot/retrocode co -P modulename

NOTE: UNIX file and directory names are case sensitive. The path to the project CVSROOT must be specified using lowercase characters (i.e. /cvsroot/retrocode)

For building the entire RetroCode SDK, you have to check out the following modules:

build  
code

fid  
include  
retroBase  
retroApple  
retroBeatnik  
retroG711  
retroGraphics  
retroMidi  
retroMono  
retroMpeg  
retroPanasonic  
retroQualcomm  
retroTheme  
retroWave  
retroYamaha

# Compilation

RetroCode™, RetroFID™ and the RetroLibraries are currently compiling fine on Microsoft Visual Studio C++ 2008 (.net), Linux GNU C++, Apple OS X GNU C++

Note: Before being able to compile RetroCode™ and the RetroLibraries, you will need to compile and install all libraries it is depending on. Some hints on specific challenges are provided in this manual.

## Compilation on Windows

Prerequisites for builing  
Microsoft Visual Studio 2008

Build Process  
Open code.sln from within Visual Studio as found in the code directory. Select Release and build the package.

## Compilation on Unix / Linux / MacOSX

Prerequisites for builing  
M4 (V1.4.11 recommended)  
autoconf (V2.62 recommended)  
automake (V1.10 recommended)  
libtool (V2.2.4 recommended)  
gcc (anything relatively recent should do)

Build Process  
Go into the build directory and run ./prepare. This will create a building environment. Now run ./configure then run make. After the make process has completed with no errors, run make install.

## Dependency

This project depends on some superbe open source libraries as well as some ISO reference implementations. At this point, the author would like to express his respect to the developers of open source libraries; “You are heroes”.

The libraries and the places where you may obtain them are listed below.

**id3lib Version 3.8.3** <http://id3lib.sourceforge.net/>

**faac  
faad2** <http://sourceforge.net/projects/faac/>

**mp4ff** part of faad2, patched for 3GPP compatibility  
<http://downloads.sourceforge.net/retrocode/mp4ff_patch_by_me_at_mmsguru_dot_com.patch?use_mirror=osdn>

**zlib** <http://www.zlib.net>

**amrnb** <http://www.3gpp.org/ftp/Specs/archive/26_series/26.104/>

**amrwb** <http://www.3gpp.org/ftp/Specs/archive/26_series/26.204/>

**mp3lame** <http://lame.sourceforge.net/index.php>

**mad** <http://sourceforge.net/projects/mad/>

**avformat   
avcodec   
avutil** <http://ffmpeg.mplayerhq.hu>

**qscl** <http://www.cdmatech.com/products/purevoice_download.jsp>

**mpeg4ip** <http://mpeg4ip.sourceforge.net/>

**libogg**  
**libvorbis** <http://xiph.org/downloads/>

**qcelp** <http://www.3gpp2.org/Public_html/specs/>

## Compiling avcodec, avutil and avformat libraries for Windows

One of the biggest challenges is getting Windows compatible versions of the FFMPEG libraries out of the sources. Those libraries are again dependent on a big number of libraries. Their build-environment is very complex and not actively supported on plain Windows machines. In other words, those libraries are the only ones that you can not compile using Visual Studio only. You will need to get autotools working and you will have to use GNU C++. The most practical way to do so is using the Minimalist GNU for Windows (MinGW).

Requisites for compiling FFMPEG on Windows:

* Installed Visual Studio 2008 (other versions may work just fine as well)
* Installed MinGW and MSYS

### Install MSYS and MinGW

Make sure you are doing the installation in the given order – and at least on Windows Vista, do not use other paths or drives for MinGW and MSYS.

Download MSYS-1.0.10.exe  
Install MSYS into C:\msys

Download MinGW-5.1.3.exe  
Install MinGW into C:\msys\1.0\mingw

Download msysDTK-1.0.1.exe  
Install msysDTK into C:\msys\1.0

Edit msys.bat and insert those red lines at the top – you will have to adapt that last new line to fit your installation location of Visual Studio:

@echo off  
set MINGW\_PATH=c:\msys\1.0\mingw

set PATH=%PATH%;%MINGW\_PATH%\bin;%MINGW\_PATH%\libexec\gcc\mingw32\3.4.5

set C\_INCLUDE\_PATH=%MINGW\_PATH%\include;%MINGW\_PATH%\lib\gcc\mingw32\3.4.5\include

set LIBRARY\_PATH=%MINGW\_PATH%\lib;%MINGW\_PATH%\lib\gcc\mingw32\3.4.5  
call "C:\Development\Microsoft Visual Studio 8\Common7\Tools\vsvars32.bat”

rem Copyright (C): 2001, 2002 Earnie Boyd

MSYS has finally been officially adapted to work with Vista 64bit. For upgrading your MSYS installation, download MSYS-1.0.11-20071204.tar.bz2 and patch your MSYS installation with it. Also download gcc-core-3.4.5-20060117-1-vista.tar.gz and patch your MingGW installation with it (only in case you are using gcc-core-3.4.5-20060117-1 – for later versions, that fix is not needed anymore).

The latest version of FFMPEG does not compile on bash 2.04 and also not with make 3.78, hence you will need to download and install a more recent versions:  
bash-3.1-MSYS-1.0.11-1.tar.bz2  
make-3.81-MSYS-1.0.11-2.tar.bz2

|  |
| --- |
|  |

Note: The parameters with underscores may only be needed for MINGW compilation.

### Compile libfaac

./bootstrap  
./configure --with-gnu-ld  
make  
make install

You may experience a compiling error like

mpeg4ip.h:74: error: conflicting declaration 'typedef int ssize\_t'

To fix that, edit faac/common/mp4v2/mpeg4ip.h and remove that conflicting typedef as that is not needed anymore (gcc’s stdint.h defines it by today):

[…]

typedef \_\_int8 int8\_t;

typedef unsigned short in\_port\_t;

typedef int socklen\_t;

//typedef int ssize\_t;

#define snprintf \_snprintf

#define strncasecmp \_strnicmp

#define strcasecmp \_stricmp

[…]

You may also experience a compiling error like

mp4atom.cpp:468: error: `isalnum' was not declared in this scope

To fix that, edit and add the inclusion of ctype.h which defines isalnum:

[…]

\* Dave Mackie dmackie@cisco.com

\* Alix Marchandise-Franquet alix@cisco.com

\* Ximpo Group Ltd. mp4v2@ximpo.com

\*/

#include <ctype.h>

#include "mp4common.h"

#include "atoms.h"

[…]

You may also experience a compiling error like

[…}

#define LOG\_INFO 6

#define LOG\_DEBUG 7

/\*

#if !\_\_STDC\_\_ && \_INTEGRAL\_MAX\_BITS >= 64

#define VAR\_TO\_FPOS(fpos, var) (fpos) = (var)

#define FPOS\_TO\_VAR(fpos, typed, var) (var) = (typed)(\_FPOSOFF(fpos))

#else

#define VAR\_TO\_FPOS(fpos, var) (fpos).lopart = ((var) & UINT\_MAX); (fpos).hipart = ((var) >> 32)

#define FPOS\_TO\_VAR(fpos, typed, var) (var) = (typed)((uint64\_t)((fpos).hipart ) << 32 | (fpos).lopart)

#endif

\*/

#define VAR\_TO\_FPOS(fpos, var) (fpos) = (var)

#define FPOS\_TO\_VAR(fpos, typed, var) (var) = (typed)(fpos)

#define \_\_STRING(expr) #expr

[…}

You may also experience a compiling error like

mp4util.cpp:78: error: `tolower' was not declared in this scope

To fix that, edit and add the inclusion of ctype.h which defines tolower

[…]

\*

\* Contributor(s):

\* Dave Mackie dmackie@cisco.com

\*/

#include <ctype.h>

#ifndef \_\_MP4\_UTIL\_INCLUDED\_\_

#define \_\_MP4\_UTIL\_INCLUDED\_\_

#include <assert.h>

[…]

### Compile libfaad2

./bootstrap  
./configure --with-gnu-ld  
make  
make install

You may experience compiling errors like

mp4ff\_int\_types.h:13: error: syntax error before "int64\_t"

To fix that, edit faad2/common/mp4ff/mp4ff\_int\_types.h and remove those extra \_WIN32 typedefs as those are not needed anymore (gcc’s stdint.h defines them by today):

#ifndef \_MP4FF\_INT\_TYPES\_H\_

#define \_MP4FF\_INT\_TYPES\_H\_

/\*

#if defined (\_WIN32)

typedef char int8\_t;

typedef unsigned char uint8\_t;

typedef short int16\_t;

typedef unsigned short uint16\_t;

typedef long int32\_t;

typedef unsigned long uint32\_t;

#else

#include <stdint.h>

#endif

\*/

#include <stdint.h>

#endif

### Compiling libmp3lame

./configure

make

make install

### Compiling libogg

./configure –-prefix=/local –-disable-shared

make

make install

### Compiling libvorbis

./configure –-prefix=/local –-disable-shared

make

make install

### Compiling libqcelp

Download the RetroPatch from sourceforge.net: <http://downloads.sourceforge.net/retrocode/qc13_tia_50_ansi733_20040315-025_retropatch03.zip>

Patch the original 13k\_Spec\_Software\_Distribution\_vA-1.0/ qc13\_tia\_50\_ansi733\_20040315-025 folder with that patchfile:

patch –p0 <qc13\_tia\_50\_ansi733\_20040315-025\_retropatch03.patch

cd qc13\_tia\_50\_ansi733\_20040315-025

As the original reference implementation contained some binary libraries already, make sure you remove those before building:

rm –f dsp\_fx/\*.a

rm –f fxptlibs/ops/\*.a

Now you can finally build the libraries

make

make install

### Compiling libamrnb

Download the libamrnb Makefile from sourceforge.net:

<http://downloads.sourceforge.net/retrocode/Makefile_libamrnb_26.104.zip?use_mirror=ovh>

For creating libamrnb, copy that Makefile into the c-code folder. Now run

make mylib  
install –c libamrnb.a /usr/local/lib  
mkdir /usr/local/include/amrnb  
cp \*.h /usr/local/include/amrnb

### Compiling libamrwb

Download the libamrwb Makefile from sourceforge.net:

<http://downloads.sourceforge.net/retrocode/Makefile_libamrwb_26.204.zip?use_mirror=ovh>

For creating libamrwb, copy that Makefile into the c-code folder. Now run

make mylib  
install –c libamrwb.a /usr/local/lib  
mkdir /usr/local/include/amrwb  
cp \*.h /usr/local/include/amrwb

### Compiling libmp4ff

Download the libmp4ff Patchfile from sourceforge.net:

<http://downloads.sourceforge.net/retrocode/mp4ff_patch_by_me_at_mmsguru_dot_com.patch?use_mirror=ovh>

Patch the original faad/common folder with that patchfile an build that library by running

patch –p0 <mp4ff\_patch\_by\_me\_at\_mmsguru\_dot\_com.patch

cd mp4ff

make  
install –c libmp4ff.a /usr/local/lib

mkdir /usr/local/include/mp4ff

cp \*.h /usr/local/include/mp4ff

### Compiling ffmpeg (avformat,avutil,avcodec)

./configure --**enable-memalign-hack** --prefix=/local --enable-shared --disable-static --enable-gpl --extra-ldflags=-L/local/lib --extra-cflags=-I/local/include --enable-libfaad --enable-libfaac --enable-libmp3lame –-enable-libvorbis -–disable-debug

make

make install

# Installing

### Windows

Simply copy the RetroCode™ executable and the third party libraries into one folder and run code from within the commandline interface.

Full list of files needed for RetroCode™ to run on windows:

code.exe

retroBase.dll  
msvcm80.dll

msvct80.dll

mcvcp80.dll

avcodec.dll needed by retroMpeg.dll  
avformat.dll needed by retroMpeg.dll  
avutil.dll needed by retroMpeg.dll  
libfaac.dll needed by retroMpeg.dll  
libfaad2.dll needed by retroMpeg.dll

qscl.dll optionally needed by retroQualcomm.dll if specified

RetroCode™ uses its plugin mechanism for the use of the various codecs. All those reside in the subirectory codecs.

codecs/retroWave.dll  
codecs/retroYamaha.dll

codecs/retroBeatnik.dll  
codecs/retroPanasonic.dll  
codecs/retroQualcomm.dll  
codecs/retroMpeg.dll

The precompiled windows binaries of RetroCode™ have been built on Microsoft Visual Studion .NET 2008.

### Linux and Mac OSX

Linux and Mac OSX: Install the Third Party Shared-Objects into e.g. /usr/local/lib. Install the RetroLibrary Shared-Objects into e.g. /usr/local/lib. Make sure that ldconfig has been run after you installed those. Install the RetroCodecs into /usr/local/libexec/Retro/codecs. Copy the RetroCode™ executable code into e.g. /usr/local/bin. Run code from within your console.

For validating the installation, simply run the RetroCode™ executable with the parameter –v. This should display a version information of the application and all linked libraries looking like this:

------------------------------------------------------------------------------

RetroCode(tm) 1.40.0

Copyright (C) 2005, Retro Ringtones LLC, Copyright (C) 2006, MMSGURU

$Id: code.cpp,v 1.1.1.1 2007/11/07 15:31:29 lobotomat Exp $

------------------------------------------------------------------------------

This program is free software: you can redistribute it and/or modify

it under the terms of the GNU General Public License as published by

the Free Software Foundation, either version 3 of the License, or

(at your option) any later version.

This program is distributed in the hope that it will be useful,

but WITHOUT ANY WARRANTY; without even the implied warranty of

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

GNU General Public License for more details.

You should have received a copy of the GNU General Public License

along with this program. If not, see <http://www.gnu.org/licenses/>.

------------------------------------------------------------------------------

$Revision: 1.1.1.1 $, $Date: 2007/11/07 15:31:29 $

------------------------------------------------------------------------------

--Components--

RetroBase 1.3

RetroYamaha 1.3.1

RetroBeatnik 1.2.2

RetroMpeg 1.0

LAME 3.96.1

libMAD 0.15.1 (beta)

libFAAC 1.24

libFAAD2 2.1 beta (Floating point version)

mpeg4ip 1.3

libAVcodec 51.7.0

libAVformat 50.3.0

RetroWave 1.2

RetroPanasonic 1.0

RetroQualcomm 1.2

QSCL 2

# Usage

RetroCode™ is a commandline tool. For detailed instructions on all of its parameters please see the chapters on [Syntax](#_Syntax).

Here comes a list of simple tasks for introducing you to the RetroCode interface.

## File Format Changes

Run RetroCode with the input-file-path and the output-file-path, if you want to change the format of a sample based ringtone. The sample data as well as all possible meta-data is taken from the input-file and encoded into the output-file.

The target format is determined by the given file-extension. That is, when you plan to create a SMAF ringtone, you will need to supply an output-filename with the file-extension MMF (not case sensitive). All currently supported encoding file formats and their file-extensions:

|  |  |
| --- | --- |
| mmf | Yamaha Synthetic Music Application Format (SMAF/MMF) |
| swf | MacroMedia Flash (SWF) |
| rmf | Beatnik Rich Music Format (RMF) |
| wav | Microsoft Wave |
| mfm | Panasonic MxxFxxMxx (MFM) |
| amr | AMR NB |
| awb | AMR WB |
| pmd | Qualcomm Compact Multimedia Format (CMF/PMD) |
| mp4 | MP4 |
| mp3 | MP3 |
| aac | Advanced Audio Coding (AAC) |
| 3gp | 3GPP |
| qcp | Qualcomm PureVoice (QCELP/QCP) |
| aiff | AIFF |
| ra | ReaLAudio |
| ogg | OGG Vorbis |
| wma | Windows Media |
| ulaw | uLaw |
| alaw | aLaw |

## Sample Frequency Changes

Add the target frequency to the RetroCode commandline using the “-s” flag. To reduce any damage (interferences) that such sample rate change does to the sample, it is highly recommended to use the filtering functions.

## Supplying Meta Data

RetroCode usually uses all metadata it finds in the source file when generating a destination. When no metadata is present in the source file, the generated destination will have a Meta-Attribute “Title” that contains the destination file-name.

Any metadata can be overwritten via commandline. Once metadata is given within the commandline, that attribute will be encoded instead of the one found in the source-file.

## Important Filter Functions

#### Lowpass filter

Use this (-lpf) whenever you changed a sampling rate. Use at the Nyquist frequency – that is half of the sample frequency.

#### Highpass filter

If used (-hpf) at 300Hz it will reduce/eliminate the low frequencies that could not be played by handsets.

#### Normalize

Use this (-n) whenever you did changes in the sample format or when you generally want to make sure that the sample is using the highest possible dynamics. Note: When activated, normalize is run in multiple steps during the sample adaption process to keep the signal quality as high as possible.

## Examples

code test\_input.wav test\_ouput.mmf -tt "Eat my shorts" -at "B.Simpson" --save true

This will convert the WAVE file "test\_input.wav" into a SMAF file named "test\_output.mmf", encoding the specified title (-tt) and artist (-at) as metadata and setting the saveable bit (--save) to true.

code hq\_input.wav test.aac -br 64000 -c 1 -s 32000 -lpf 16000

This will read and decode the WAVE file "hq\_input.wav", reduce the sample to mono (-c) and the samplerate (-s) into 32kHz. Before reducing the samplerate, a lowpassfilter (-lpf) is applied at 16kHz. The resulting sample is then encoded using the AAC Low Complexity (LC) profile.

code foobar.wav test.mp4 -br 64000 -dt 0 -mp 4

This will convert the WAVE file "foobar.wav" and encode it into the MP4-file "test.mp4" using the AAC Low Complexity [LC] profile, storing the sample as MPEG4 (-mp) raw AAC (-dt) packets.

code in.rmf out.pmd

This will read and decode the RMF file "in.rmf" which is then converted into the CMF file "out.pmd"".

code in.mmf out.amr -br 12200

This will read and decode the SMAF file "in.mmf" which is then converted with a resulting bitrate (-br) of 12200 into the AMR-NB file "out.amr"".

# Syntax

code [-SHORT] [--LONG] SOURCE DESTINATION

GENERAL PARAMETERS

--LONG (-SHORT)

--version (-v)  
 This switch will prevent any file processing and trigger only the display the version information of this tool.

--help (-h)   
This switch will prevent any file processing and trigger onlz the display of a short parameter overview of this tool.

--nometa (-nm)  
 The switch will prevent any metadata encoding into the destination file.

--codecdir(-cod) default: “codecs”  
Use this to supply any folder as a source for retro-codec libraries.

--autoadapt(-aa) default: 0  
To automatically convert in channel count and/or sample rate towards one fitting the destination format, supply a value higher than 0. Future revisions will possibly offer quality levels.

--manual (-m)  
 This switch will prevent any file processing and trigger only the display this manual.

--output (-o) default: 2  
This attribute allows you to set the verbosity level of this tool. The higher the verbosity level, the more this tool will tell you while working.   
0=keep quiet  
1=only errors  
2=warnings & messages  
3=debug1  
4=debug2  
5=debug3

--samplerate (-s) default: 0  
To adapt the sample rate of your source material, use this rate interpolating function. The parameter is given in hertz [Hz].

--channels (-c) default: 0  
Number of output channels   
1=mono  
2=stereo

--lowpassfreq (-lpf) default: 0  
Lowpass filter cutoff frequency. Use this in conjunction with any samplerate reduction at the Nyquist frequency. If the samplerate is e.g. adapted towards 8kHz, you should use the lowpass filter at 4kHz. The parameter is given in hertz [Hz].

--highpassfreq (-hpf) default: 0  
Highpass filter cutoff frequency. The parameter is given in hertz [Hz].

--bandpassfreq (-bpf) default: 0  
Bandpass filter center frequency. The parameter is given in hertz [Hz].

--bandpasswidth (-bpw) default: 15000  
Bandpass filter bandwidth. The parameter is given in hertz [Hz].

--volgain (-vg) default: 0.000000  
Amplification gain in dB to amplitude (6.0 = doubled amplification). This parameter is given in decibel [dB].

--vollimitgain (-vl) default: 0.000000  
Peak limiting gain (0.001 - 0.999). For a first attempt, 0.2 might be a good value. This parameter is given in decibel [dB].

--normalize (-n)  
Normalize sample data. Two pass; analyzer and amplifier.

--rmsnormalizer (-rms) default: 0.000000   
RMS-normalize sample data. This parameter is given in decibel [dB].

--autocrop (-sac)  
Prevent gaps by cropping the sample below 0.5db at head and tail. MP3 sources should always be treated with this filter.

--sampleplaytime (-spt) default: 0  
Negative time means the actual playtime reduced by the given value. The value is given in milliseconds [ms].

--ringback (-rb) default: 0  
Render and mix a ringback signal into the original sample. Possible values are   
1=USA  
2=UK  
3=Germany  
4=France  
5=Fiji

--sampleoffset (-sof) default: 0  
The value is given in milliseconds [ms]. If this value is lower than the actual sample playback length, the sample is cropped to fit.

--samplelooptime (-slt) default: 0  
Ignore the sample playback duration and simply loop it until this value is reached. The value is given in milliseconds [ms]. If this value is lower than the actual sample playback length, the sample is cropped to fit.

--samplefadeintime (-sfi) default: 0  
Use this option in conjunction with the playtime option, set playtime to values like 27000 and fadetime to 3000 to receive a total playback duration of 30000 milliseconds [ms].

--samplefadeouttime (-sfo) default: 0  
Use this option in conjunction with the playtime option, set playtime to values like 27000 and fadetime to 3000 to receive a total playback duration of 30000 milliseconds [ms].

--backgroundcolor (-col) default: 0  
no help available

--framewidth (-frx) default: 1  
no help available

--frameheight (-fry) default: 1  
no help available

--software (-sft)   
Rendering software name.

FORMAT SPECIFIC PARAMETERS

### Yamaha Synthetic Music Application Format (SMAF/MMF)

Yamaha's SMAF format exists in at least four versions; MA1, MA2, MA3, MA5 and MA7. It allows the encapsulations of various audio formats, graphics and even the usage of a speech synthesizer.

The generated SMAF compatibility mostly depends on the sample format you want to encode - but also on the chosen encoding. Streaming samples e.g. are not supported by MA2. Additionally, there is a special attribute within SMAF files, the PRO-tag - once that is included in a SMAF file, only MA5 and MA7 support it.

#### Internal Sequence Format introduction

A “normal” SMAF file is composed of meta-data, sequence-data and sample data. The internal sequence data format is called compact MIDI, which in some cases looks just like standard MIDI.

#### Standard MIDI reminder

MIDI is a very high level format that does not really describe/propose the way things like voice limitations are handled when e.g. a playback instrument runs out of available sound-generators. MIDI has several ways of addressing instruments and sounds. One of those ways is the so called MIDI-channel (1-16). MIDI-channels do not impose any limitation on the polyphony – each and every single one of them is polyphonic without limits. A MIDI sequence may additionally be partitioned by the use of logical tracks. A midi-track also does not impose any limitations on the amount of channels or voices that are played simultaneously.

#### Single Event timing vs. Dual Event Timing

There are two common approaches of storing the timing of a musical sequence. Those describe the duration of the notes and the pauses in-between.

Dual Event: Each note consists of two events. The first event is the start of a note (note-on) and the second one is the end of the note (note-off). Standard MIDI uses this as it reflects the un-interpreted communication of electronic instruments. That is; when I press a key down it sends a signal over the wire and when I release that key it sends another signal.

Single Event: Each note consists of one event. It describes the start-time, note-value, volume and duration. Compact MIDI uses this.

As a result of this change to single-event timing, the average data amount needed for storing a specific sequence is considerably smaller without losing any attribute. Thus this format was called compact MIDI by Yamaha.

#### Internal Sequence Format continued

The SMAF compact midi format itself had been undergone several iterations to allow new SMAF chip generations to be used in their full beauty while keeping to as much hardware compatibility as possible. The SMAF file format however is interpreted mostly by the Yamaha chip itself and not a software sequence/synthesizer. It works on an interesting hybrid level between high-level sequencing (MIDI) and low level sample trigger (MOD). The SMAF MA2 format for example offers up to four tracks of sequence data. Each of them is devided into four channels. A channel within this format is directly linked to one hardware voice. If becomes obvious that Yamaha simply multiplied the amount of usable tracks from 1 to 4 when upgrading from the 4 voice MA1 to the 16 voice MA2.

The process of converting polyphonic MIDI content into SMAF’s semi polyphonic compact MIDI is a pretty demanding task as it involves an accurate and complete timing simulation of the Yamaha SMAF LSIs (hardware chips). This is the main reason on why all polyphonic conversion technology for that format today comes from Yamaha directly. In the early days, there were a few tools attempting to cover that task without using Yamaha’s conversion API – they utterly failed in compatibility and further development was ceased altogether. RetroCode does not really convert any MIDI to SMAF, hence these limitations were no issue while developing.

RetroCode does not rely on any software from Yamaha when handling SMAF files. The biggest disadvandtage of using their APIs is that those libraries are only available on certain platforms (mostly windows). RetroSMAF is, to our knowledge today the only portable free sample codec covering all current sample aware hardware platforms (MA2-MA7).

The SMAF format allows various codecs for the use with sample data. Within mobile phones however, only the ADPCM variant is commonly supported.

When converting sample data into SMAF files, what RetroCode™ actually does is the following;

* Encode PCM sample data into the Yamaha ADPCM format
* Render a short compact MIDI sequence that is containing a note-on event triggering of the sample
* Encode META data
* Encode Checksums

#### Valid input sample format/s

#### 16Bit, Mono or Stereo, 4000Hz to 44100Hz

#### Streaming samples

The SMAF format offers two modes of storing sample data; the traditional audio track storage and the streaming sample mode. Streaming samples have been introduced with the SMAF MA3 format. Streaming samples cannot be used with SMAF MA2 only compatible devices. Streaming samples have weaker limitations on the sample size, hence when available, this encoding should be chosen.

Non-streaming samples have an important format limitation when it comes to encoding the sample rate; only the following values can be used; 4000Hz, 8000Hz, 11025Hz, 22050Hz, 44100Hz. Streaming samples on the other hand allow the encoding of any sample rate from 4000Hz up to 44100Hz (in steps of 1Hz).

#### Sample format and SMAF compatibility

Mono

|  |  |
| --- | --- |
| MA2, MA3, MA5, MA7 | below or equal 8kHz |
| MA3, MA5, MA7 | below or equal 16kHz |
| MA5, MA7 | below or equal 24kHz |
| MA7 | above 24kHz |

Stereo

|  |  |
| --- | --- |
| MA5, MA7 | below or equal 12kHz |
| MA7 | above 12kHz |

#### SMAF sample format overview

Some encode-able sample rates are given in this table along with the resulting SMAF generation compatibility.

|  |  |  |
| --- | --- | --- |
| **Frequency [Hz]** | **Mono** | **Stereo** |
| 4.000 | MA2 | MA5 |
| 8.000 | MA2 | MA5 |
| 11.025 | MA3 | MA5 |
| 12.000 | MA3 | MA5 |
| 16.000 | MA3 | MA7 |
| 22.050 | MA5 | MA7 |
| 24.000 | MA5 | MA7 |
| 32.000 | MA7 | MA7 |
| 44.100 | MA7 | MA7 |

--stream (-ss) default: false  
Samples are stored in streaming format (MA3 and upwards - valid only for 8kHz and below).

--deviceid(-did)  
Device specific messages targeting a Yamaha SMAF chip model. Within a SMAF file, a few system exclusive messages are used - that means it contains device specific data marked by a device identifier. Set implicitely by the attributes of the input sample format when 0=Auto is used. 1\*=MA2 (YMU786), 2=MA3 (YMU762), 3=MA5 (YMU765), 4=MA7 (YMU786). Valid only when the streaming sample format has been selected (stream/ss).

(\*) Do not use for MA2 as streaming samples dont work on MA2 devices.

--save (-bs) default: true  
If this flag is set, the resulting object will not be savable on many SMAF compatible devices. Set this flag to produce preview files.

--edit (-be) default: false  
If this flag is set, the resulting object will not be editable on most SMAF compatible devices. Set this flag generally to true or 1.

--transfer (-bt) default: false  
If this flag is set, the resulting object will not be copyable on many SMAF compatible devices. Set this flag generally to true or 1.

--protag (-p5) default: false  
If this flag is set, the resulting object contain a tag making the SMAF compatible only on MA5 and above.

--title (-tt) default: <FILENAME>  
META: Song/Tone/Object Title

--artist (-at)  
META: Artist

--note (-no)  
META: Note/Message/Remarks

--writer (-wr)  
META: Writer

--category (-ca)  
META: Category

--subcategory (-sc)  
META: SubCategory

--copyright (-cp)  
META: Copyright

--vendor (-ve)  
META: Vendor

--arranger (-ar)  
META: Arranger

--management (-ma)  
META: Management info

--managedby (-mb)  
META: Managed by

--carrier (-cr)  
META: Carrier

--datecreated (-dc)  
META: Date created

--daterevised (-dr)  
META: Date revised

### MacroMedia Flash (SWF)

MacroMedia Flash is a macro-scripting, audio, graphics and video wrapping format. The audio codec is usually MP3.

--bitrate (-br) default: 0

Encoded stream bitrate. Use carefully, some formats only allow specific bitrates. MP3: 8000,16000,24000,32000,40000,48000,56000,64000,80000,96000,112000,128000,160000,192000,224000,256000,320000

--dtx (-dx) default: false  
Enable DTX encoding.

### Wave

The Microsoft Wave format allows encapsulation of many audio formats. The most common are PCM and ADPCM.

#### Valid input sample format/s

#### 16Bit, Mono or Stereo, 4000Hz or 8000Hz or 11025Hz or 12000Hz or 16000Hz or 22050Hz or 24000Hz or 32000Hz or 44100Hz

--reverseorder (-rvo)  
Reverse order header output. Usually, Meta-Data is encoded in front of the actual sample data, when using this switch, that order is reversed.

--goldwave (-glw)  
Goldwave fixed format bug. Some older versions of Goldwave (a Windows Shareware Wave Editor) rendered slightly invalid wave files, for simulating that exact bug, use this flag.

--comp (-cc) default: 0  
Determines the compression variant used.  
0=no compression  
1=IMA ADPCM  
2=MS ADPCM  
3=QCELP  
4=SAGEM ADPCM.

--fact (-fct)  
Include fact chunk.

--title (-tt) default: <FILENAME>  
META: Song/Tone/Object Title

--artist (-at)  
META: Artist

--note (-no)  
META: Note/Message/Remarks

--category (-ca)  
META: Category

--copyright (-cp)  
META: Copyright

--datecreated (-dc)  
META: Date created

### Beatnik Rich Music Format (RMF)

The Beatnik Rich Music format allows encapsulation of various audio formats. RMF is one of the few mobile sample formats that allows non-linear playback of sample audio data. RMF is the only common ringtone format using a (very simple) encryption for most parts of the file. RMF also features a data compression.

A “normal” RMF file is composed of meta-data, sequence-data and sample data. The internal sequence data format is MIDI. So unlike SMAF or CMX, this format actually incorporates true MIDI data.

The RMF format allows various codecs for the use with sample data. Within mobile phones however, only the ADPCM variant is commonly supported.

When converting sample data into RMF files, what RetroCode™ actually does is the following;

* Encode PCM sample data into the Apple IMA ADPCM format
* Render a short MIDI sequence that is containing a note-on event triggering of the sample
* Encode META data
* Compress RMF data
* Encrypt RMF data

#### Playback duration

foobar

#### Volume control

The RMF format supports multiple ways of controlling the sample playback volume:

* Global file volume (--volume/-vo)
* Instrument data volume
* MIDI control volume
* MIDI note velocity

For creating a volume fade down to silence at the end of the playback time (--playtime), you can to supply a duration of that fade (--paraNumFadetime/-ft) or use the default of 4 seconds. RetroCode only inserts a number of MIDI controls changing the playback volume. This volume change does hardly affect the resulting file-size as the sample data is not changed, the changed dynamics are rendered by the software synthesizer within the RMF compatible device.

#### Valid input sample format/s

16Bit, Mono or Stereo, 1Hz to 44100Hz

--volume (-vo) default: 100  
Playback volume in percent. Use a value of 100 for best results in most cases.

--tempo (-tm) default: 100  
Playback tempo percentage. Use this option for finetuning/compensating the device dependant playback speed.

--playtime (-pt) default: 26000  
Ignores the sample playback duration and simply loop it until this value is reached. The value is given in milliseconds [ms]. If this value is lower than the actual sample playback length, the sample is cropped to fit.

--paraNumFadetime (-ft) default: 4000  
Fade out duration. The value is given in milliseconds [ms].

--title (-tt) default: <FILENAME>  
META: Song/Tone/Object Title

--artist (-at)  
META: Artist

--composer (-cm)  
META: Composer

--note (-no)  
META: Note/Message/Remarks

--category (-ca)  
META: Category

--subcategory (-sc)  
META: SubCategory

--copyright (-cp)  
META: Copyright

--publisher (-pu)  
META: Publisher

--licenseuse (-ls)  
META: License usage

--licenseterm (-lt)  
META: License term

--licenseurl (-lu)  
META: License URL

--licenseexp (-le)  
META: License expiration date

--source (-sr)  
META: Original source

--tempodesc (-td)  
META: Tempo description

--index (-id)  
META: Index

### Panasonic MxxFxxMxx (MFM)

Panasonic's MFM format is a straight linear format with few extra attributes. Just like VOX, this format also uses the OKI ADPCM algorithm.

#### Valid input sample format/s

16Bit, Mono, 8000Hz

--title (-tt) default: <FILENAME>  
META: Song/Tone/Object Title

--copyright (-cp)  
META: Copyright

--encoder (-en)  
META: Encoder

### VOX

Dialogic ADPCM usually is used within telephony systems. The used file format is very old, files are sometimes encoded with proprietary or rare codecs - standard however is the Dialog ADPCM codec for VOX files. The Dialogic ADPCM codec is also known as the OKI ADPCM codec.

#### Valid input sample format/s

16Bit, Mono, 8000Hz

--title (-tt) default: <FILENAME>  
META: Song/Tone/Object Title

--copyright (-cp)  
META: Copyright

--encoder (-en)  
META: Encoder

### AMR NB

AMR Narrow Band is a very basic linear sample file format. No metadata or any other attribute within the files is possible.

#### Valid input sample format/s

16Bit, Mono or Stereo, 8000Hz

--bitrate (-br) default: 0

Encoded stream bitrate. Valid bitrates: 4750,5150,5900,6700,7400,7950,10020,12200

--dtx (-dx) default: false  
Enable DTX encoding.

### AMR WB

AMR Wide Band is a very basic linear sample file format. No metadata or any other attribute within the files is possible.

#### Valid input sample format/s

16Bit, Mono or Stereo, 16000Hz

--bitrate (-br) default: 0  
Encoded stream bitrate. Valid bitrates: 6600,8850,12650,14250,15850,18250,19850,23050,23850

--dtx (-dx) default: false  
Enable DTX encoding.

### Qualcomm Compact Multimedia Format (CMF/PMD)

Qualcomm's Compact Media Format (CMF / CMX) allows the encapsulation of various audio and graphics formats. It is played only by CDMA devices, hence very limited in compatibility.

#### Valid input sample format/s

16Bit, Mono, 8000Hz or 16000Hz or 32000Hz

--comp (-cc) default: 0  
Determines the compression variant used.   
0=no compression  
1=IMA ADPCM  
2=MS ADPCM (do not use for this file format)  
3=QCELP  
4=SAGEM ADPCM (do not use for this file format)

--volume (-vo) default: 100  
Playback volume in percent. Use a value of 100 for best results in most cases.

--loopcount (-lc) default: 15  
Number of repetitions (2=twice - 15=infinite)

--title (-tt) default: <FILENAME>  
META: Song/Tone/Object Title

--copyright (-cp)  
META: Copyright

--publisher (-pu)  
META: Publisher

--datecreated (-dc)  
META: Date created

### Advanced Audio Coding (AAC)

Advanced Audio Codec

#### Valid input sample format/s

16Bit, Mono or Stereo, 7350Hz or 8000Hz or 11025Hz or 12000Hz or 16000Hz or 22050Hz or 24000Hz or 32000Hz or 44100Hz or 48000Hz or 64000Hz or 88200Hz or 96000Hz

--bitrate (-br) default: 0  
Encoded stream bitrate. Valid bitrates: min=8000, max is sample frequency dependant

--id3 (-i3) default: false  
Enable ID3 tagging.

--mpeg (-mp) default: 4  
2=MPEG2 encoding  
4=MPEG4 encoding (Do not mix this up with MP4 file encoding, that is a different thing)

--aac (-ac) default: 1  
This attribute allows you to determine the AAC profile used for encoding the audio data.   
0=Main  
1=Low Complexity (LC)  
2=Scalable Sample Rate (SSR)  
3=Long Term Prediction (LTP)  
4=High Efficiency (HE)

--adts (-dt) default: 1  
This attribute allows you to determine if the AAC data should be encoded with or without the ADTS frame headers (as needed within MP4 files).   
0=raw AAC  
1=ADTS AAC

--title (-tt) default: <FILENAME>  
META: Song/Tone/Object Title

--artist (-at)  
META: Artist

--note (-no)  
META: Note/Message/Remarks

--category (-ca)  
META: Category

--copyright (-cp)  
META: Copyright

### MPEG-4 (MP4)

Foobar

#### Valid input sample format/s

16Bit, Mono or Stereo, 7350Hz or 8000Hz or 11025Hz or 12000Hz or 16000Hz or 22050Hz or 4000Hz or 32000Hz or 44100Hz or 48000Hz or 64000Hz or 88200Hz or 96000Hz

--bitrate (-br) default: 0  
Encoded stream bitrate. Use carefully, some formats only allow specific bitrates. AMR-NB: 4750,5150,5900,6700,7400,7950,10020,12200 AMR-WB: 6600,8850,12650,14250,15850,18250,19850,23050,23850 MP3: 8000,16000,24000,32000,40000,48000,56000,64000,80000,96000,112000,128000,160000,192000,224000,256000,320000 AAC: min=8000, max is sample frequency dependant

--id3 (-i3) default: false  
Enable ID3 tagging.

--mpeg (-mp) default: 4  
2=MPEG2 encoding  
4=MPEG4 encoding (Do not mix this up with MP4 file encoding, that is a different thing)

--aac (-ac) default: 1  
This attribute allows you to determine the AAC profile used for encoding the audio data.   
0=Main  
1=Low Complexity (LC)  
2=Scalable Sample Rate (SSR)  
3=Long Term Prediction (LTP)  
4=High Efficiency (HE)

--adts (-dt) default: 1  
This attribute allows you to determine if the AAC data should be encoded with or without the ADTS frame headers (as needed within MP4 files).   
0=raw AAC  
1=ADTS AAC

--title (-tt) default: <FILENAME>  
META: Song/Tone/Object Title

--note (-no)  
META: Note/Message/Remarks

--category (-ca)  
META: Category

--copyright (-cp)  
META: Copyright

### 3GPP File Format (3GP)

foobar

#### Valid input sample format/s

16Bit, Mono or Stereo, 7350Hz or 8000Hz or 11025Hz or 12000Hz or 16000Hz or 22050Hz or 4000Hz or 32000Hz or 44100Hz or 48000Hz or 64000Hz or 88200Hz or 96000Hz

--bitrate (-br) default: 0  
Encoded stream bitrate. Use carefully, some formats only allow specific bitrates. AMR-NB: 4750,5150,5900,6700,7400,7950,10020,12200 AMR-WB: 6600,8850,12650,14250,15850,18250,19850,23050,23850 MP3: 8000,16000,24000,32000,40000,48000,56000,64000,80000,96000,112000,128000,160000,192000,224000,256000,320000 AAC: min=8000, max is sample frequency dependant

--id3 (-i3) default: false  
Enable ID3 tagging.

--mpeg (-mp) default: 4  
2=MPEG2 encoding  
4=MPEG4 encoding (Do not mix this up with MP4 file encoding, that is a different thing)

--aac (-ac) default: 1  
This attribute allows you to determine the AAC profile used for encoding the audio data.   
0=Main  
1=Low Complexity (LC)  
2=Scalable Sample Rate (SSR)  
3=Long Term Prediction (LTP)  
4=High Efficiency (HE)

--adts (-dt) default: 1  
This attribute allows you to determine if the AAC data should be encoded with or without the ADTS frame headers (as needed within MP4 files).   
0=raw AAC  
1=ADTS AAC

--title (-tt) default: <FILENAME>  
META: Song/Tone/Object Title

--note (-no)  
META: Note/Message/Remarks

--category (-ca)  
META: Category

--copyright (-cp)  
META: Copyright

### Qualcomm PureVoice (QCELP/QCP)

Foobar

#### Valid input sample format/s

16Bit, Mono, 8000Hz

### Audio Interchange File Format (AIFF)

Foobar

#### Valid input sample format/s

8Bit or 16Bit, Mono or Stereo, 1Hz to 192000Hz

--title (-tt) default: <FILENAME>  
META: Song/Tone/Object Title

--note (-no)  
META: Note/Message/Remarks

--writer (-wr)  
META: Writer

--copyright (-cp)  
META: Copyright

### Windows Media Audio (WMA)

foobar

--bitrate (-br) default: 0  
Encoded stream bitrate. Valid bitrates: 8000,16000,24000,32000,40000,48000,56000,64000,80000,96000,112000,128000,160000,192000,224000,256000,320000

--title (-tt) default: <FILENAME>  
META: Song/Tone/Object Title

--artist (-at)  
META: Artist

--note (-no)  
META: Note/Message/Remarks

--copyright (-cp)  
META: Copyright

### Real Audio (RA)

foobar

--bitrate (-br) default: 0

Encoded stream bitrate. Valid bitrates: 12000,24000

--title (-tt) default: <FILENAME>  
META: Song/Tone/Object Title

--artist (-at)  
META: Artist

--note (-no)  
META: Note/Message/Remarks

--copyright (-cp)  
META: Copyright

### OGG

foobar

--bitrate (-br) default: 0  
Encoded stream bitrate. Use carefully, some formats only allow specific bitrates.

### AVI

Audio Video Interleaved, currently only supported for decoding.

### uLaw

muLaw usually is used within telephony systems originated from the United States. The file format is very old and does not handle any metadata at all. It also does not really have any limitation on the sample frequency, just those 8.000Hz are the commonly used one for this format.

#### Valid input sample format/s

16Bit, Mono, 8000Hz

### aLaw

aLaw usually is used within telephony systems originated from Europe. The file format is very old and does not handle any metadata at all. It also does not really have any limitation on the sample frequency, just those 8.000Hz are the commonly used one for this format.

#### Valid input sample format/s

16Bit, Mono, 8000Hz

### MPEG-1,2 Level 3 (MP3)

MPEG Format 1, 2 and 2.5 Layer 3.

#### Valid input sample format/s

16Bit, Mono or Stereo, 11025Hz or 16000Hz or 22050Hz or 24000Hz or 32000Hz or 44100Hz or 48000Hz

--bitrate (-br) default: 0  
Encoded stream bitrate. Valid bitrates: 8000,16000,24000,32000,40000,48000,56000,64000,80000,96000,112000,128000,160000,192000,224000,256000,320000.

--id3 (-i3) default: false  
Enable ID3 tagging.

--vbr (-vb) default: false  
Enable variable bitrate encoding (VBR).

--crc (-cr) default: false  
Enable cyclic redundancy check value encoding (CRC).

--jstereo (-js) default: false  
Enable joint stereo encoding for stereo content. It is recommended to use joint stereo mode whenever the encoded bitrate is lower than 192kbps.

--title (-tt) default: <FILENAME>  
META: Song/Tone/Object Title

--subtitle (-st)  
META: Song/Tone/Object SubTitle

--artist (-at)  
META: Artist

--note (-no)  
META: Note/Message/Remarks

--category (-ca)  
META: Category

--copyright (-cp)  
META: Copyright

--encoder (-en)  
META: Encoder

# Frequently Asked Questions

## How do I create a Wave-file compatible with a Siemens S65?

code input.wav output.wav -nm –fct –cc 1 –s 8000 –lpf 4000 –c 1

## How do I create a Wave-file compatible with a Sagem handset?

code input.wav output.wav –nm –cc 4 –s 8000 –lpf 4000 –c 1

# Notes for Developers

## Basic function overview of the RetroCode commandline application

RetroCode identifies the source stream's format using all available codecs, reads and decodes it entirely, keeping all important assets in memory. The codecs are derived from CMobileSampleContent.

Sample data internally is to be 16bit. When the input stream data is lower than that, the first step needed after decoding this sample is scale it towards 16bit.

During the filtering process (filter, normalize, rate convert,...) all resulting data is maintained in that source-sample object. Finally the source object is handed over to an instance of the destination format class which then encodes the data and writes it to the output stream.

Important CMobileSampleContent (codec) methods  
Read - reads, parses and decodes the file

SetSource - attaches the source sample object to the file that is to be writting

Write - encodes and writes the file

For more members of CMobileContent and CMobileSampleContent, check MobileContent.h

Developing new codecs  
For a good and relative clean example of a codec, check out retroApple containing the AIFF-parser.

# CREDITS

**Be aware that some of the used codecs are based on fragmential specifications, released by their original IP holders. Namely Retro's RMF codec is not licensed or in any way approved by Beatnik Software. Please make sure that you read all mentioned documents below to find out if you have to pay additional licensing fees when rendering files using this software.**

## Author

Except where otherwise noted, all code was authored by:  
Till Toenshoff <me@mmsguru.com>

## Contributors

Vinicius Kursancew <viniciusalexandre@gmail.com>  
- suggested some code cleanup to simplify compiling  
- provided the autotools make package  
- provided Packer.cpp / Packer.h  
Thomas ‘Dolby’ Robertson  
- contributed his wrecked Powerbook G4 for big-endian code testing  
Lukas Loesche <lloesche@gmail.com>  
- contributed a 60gb HDD for that wrecked Powerbook’s revival  
Dave Lindsay  
- contributed money to support the ongoing development

## IP Holders

Retro's SMAF codec is based on: "Specification: Synthetic music Mobile Application Format" Ver.3.06, Copyright (c) 1999-2002 by YAMAHA CORPORATION.

Retro's CMF codec is based on: "Internet-Draft: draft-atarius-cmf-00.txt", Copyright (c) 2004 by The Internet Society; "3GPP2 C.S0050-0 Version 1.0 - 3GPP2 File Formats for Multimedia Services", Copyright (c) 2003 by 3GGP2.

Retro's MFM codec is based on: "Dialogic ADPCM Algorithm", Copyright (c) 1988 by Dialogic Corporation.

The Qcelp codec is entirely based on: "RFC 3625 - The QCP File Format and Media Types for Speech Data", Copyright (c) 2003 by The Internet Society; "Qualcomm Speech Codec Library", Copyright (c) 2003 by QUALCOMM, Inc.

The AAC codec is entirely based on: "FAAC", Copyright (c) 2001 by M. Bakker; "FAAD2", Copyright (c) 2003 by M. Bakker.

The MP3 codec is entirely based on: "LAME", Copyright (c) 1999 by A.L. Faber; "MPG123", Copyright (c) 1995-1997 by Michael Hipp, "libmad", Copyright (c) 2000-2004 by Underbit Technologies, Inc.

The AMR-NB codec is entirely based on: "3GPP specification TS 26.101: Mandatory speech processing functions; AMR speech codec frame structure", Copyright (c) 2003 by 3GPP; "TS 26.104: 3GPP AMR Floating-point Speech Codec V5.1.0"), Copyright (c) 2003 by 3GPP.

The AMR-WB codec is entirely based on: "3GPP specification TS 26.201: Speech Codec speech processing functions; AMR Wideband Speech Codec; Frame Structure", Copyright (c) 2003 by 3GPP; "TS 26.204: 3GPP AMR Wideband Floating-point Speech Codec"), Copyright (c) 2003 by 3GPP.

The SWF codec is entirely based on: "Macromedia Flash (SWF) and Flash Video (FLV) File Format Specification Version 8", Copyright (c) 2006 by Adobe Inc.

The ID3 parser is entirely based on: "id3lib", Copyright (c) 1999, 2000 by Scott Thomas Haug, 2002 by Thijmen Klok.

The MP4 encoder is entirely based on: "mpeg4ip", Copyright (c) 2001 by Cisco Systems Inc.

The Butterworth filter is entirely based on: "Sound Processing Kit - A C++ Class Library for Audio Signal Processing", Copyright (c) 1995-1998 Kai Lassfolk.

RetroCode(tm) and all its components were originally assembled in 2005 by Till Toenshoff for inhouse usage at Retro Ringtones LLC.

# Appendix A: GPL License Version 3

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies

of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for

software and other kinds of works.

The licenses for most software and other practical works are designed

to take away your freedom to share and change the works. By contrast,

the GNU General Public License is intended to guarantee your freedom to

share and change all versions of a program--to make sure it remains free

software for all its users. We, the Free Software Foundation, use the

GNU General Public License for most of our software; it applies also to

any other work released this way by its authors. You can apply it to

your programs, too.

When we speak of free software, we are referring to freedom, not

price. Our General Public Licenses are designed to make sure that you

have the freedom to distribute copies of free software (and charge for

them if you wish), that you receive source code or can get it if you

want it, that you can change the software or use pieces of it in new

free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you

these rights or asking you to surrender the rights. Therefore, you have

certain responsibilities if you distribute copies of the software, or if

you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether

gratis or for a fee, you must pass on to the recipients the same

freedoms that you received. You must make sure that they, too, receive

or can get the source code. And you must show them these terms so they

know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1) assert copyright on the software, and (2) offer you this License

giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains

that there is no warranty for this free software. For both users' and

authors' sake, the GPL requires that modified versions be marked as

changed, so that their problems will not be attributed erroneously to

authors of previous versions.

Some devices are designed to deny users access to install or run

modified versions of the software inside them, although the manufacturer

can do so. This is fundamentally incompatible with the aim of

protecting users' freedom to change the software. The systematic

pattern of such abuse occurs in the area of products for individuals to

use, which is precisely where it is most unacceptable. Therefore, we

have designed this version of the GPL to prohibit the practice for those

products. If such problems arise substantially in other domains, we

stand ready to extend this provision to those domains in future versions

of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents.

States should not allow patents to restrict development and use of

software on general-purpose computers, but in those that do, we wish to

avoid the special danger that patents applied to a free program could

make it effectively proprietary. To prevent this, the GPL assures that

patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and

modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of

works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this

License. Each licensee is addressed as "you". "Licensees" and

"recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work

in a fashion requiring copyright permission, other than the making of an

exact copy. The resulting work is called a "modified version" of the

earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based

on the Program.

To "propagate" a work means to do anything with it that, without

permission, would make you directly or secondarily liable for

infringement under applicable copyright law, except executing it on a

computer or modifying a private copy. Propagation includes copying,

distribution (with or without modification), making available to the

public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other

parties to make or receive copies. Mere interaction with a user through

a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices"

to the extent that it includes a convenient and prominently visible

feature that (1) displays an appropriate copyright notice, and (2)

tells the user that there is no warranty for the work (except to the

extent that warranties are provided), that licensees may convey the

work under this License, and how to view a copy of this License. If

the interface presents a list of user commands or options, such as a

menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work

for making modifications to it. "Object code" means any non-source

form of a work.

A "Standard Interface" means an interface that either is an official

standard defined by a recognized standards body, or, in the case of

interfaces specified for a particular programming language, one that

is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other

than the work as a whole, that (a) is included in the normal form of

packaging a Major Component, but which is not part of that Major

Component, and (b) serves only to enable use of the work with that

Major Component, or to implement a Standard Interface for which an

implementation is available to the public in source code form. A

"Major Component", in this context, means a major essential component

(kernel, window system, and so on) of the specific operating system

(if any) on which the executable work runs, or a compiler used to

produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all

the source code needed to generate, install, and (for an executable

work) run the object code and to modify the work, including scripts to

control those activities. However, it does not include the work's

System Libraries, or general-purpose tools or generally available free

programs which are used unmodified in performing those activities but

which are not part of the work. For example, Corresponding Source

includes interface definition files associated with source files for

the work, and the source code for shared libraries and dynamically

linked subprograms that the work is specifically designed to require,

such as by intimate data communication or control flow between those

subprograms and other parts of the work.

The Corresponding Source need not include anything that users

can regenerate automatically from other parts of the Corresponding

Source.

The Corresponding Source for a work in source code form is that

same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of

copyright on the Program, and are irrevocable provided the stated

conditions are met. This License explicitly affirms your unlimited

permission to run the unmodified Program. The output from running a

covered work is covered by this License only if the output, given its

content, constitutes a covered work. This License acknowledges your

rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not

convey, without conditions so long as your license otherwise remains

in force. You may convey covered works to others for the sole purpose

of having them make modifications exclusively for you, or provide you

with facilities for running those works, provided that you comply with

the terms of this License in conveying all material for which you do

not control copyright. Those thus making or running the covered works

for you must do so exclusively on your behalf, under your direction

and control, on terms that prohibit them from making any copies of

your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under

the conditions stated below. Sublicensing is not allowed; section 10

makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological

measure under any applicable law fulfilling obligations under article

11 of the WIPO copyright treaty adopted on 20 December 1996, or

similar laws prohibiting or restricting circumvention of such

measures.

When you convey a covered work, you waive any legal power to forbid

circumvention of technological measures to the extent such circumvention

is effected by exercising rights under this License with respect to

the covered work, and you disclaim any intention to limit operation or

modification of the work as a means of enforcing, against the work's

users, your or third parties' legal rights to forbid circumvention of

technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you

receive it, in any medium, provided that you conspicuously and

appropriately publish on each copy an appropriate copyright notice;

keep intact all notices stating that this License and any

non-permissive terms added in accord with section 7 apply to the code;

keep intact all notices of the absence of any warranty; and give all

recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey,

and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to

produce it from the Program, in the form of source code under the

terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified

it, and giving a relevant date.

b) The work must carry prominent notices stating that it is

released under this License and any conditions added under section

7. This requirement modifies the requirement in section 4 to

"keep intact all notices".

c) You must license the entire work, as a whole, under this

License to anyone who comes into possession of a copy. This

License will therefore apply, along with any applicable section 7

additional terms, to the whole of the work, and all its parts,

regardless of how they are packaged. This License gives no

permission to license the work in any other way, but it does not

invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display

Appropriate Legal Notices; however, if the Program has interactive

interfaces that do not display Appropriate Legal Notices, your

work need not make them do so.

A compilation of a covered work with other separate and independent

works, which are not by their nature extensions of the covered work,

and which are not combined with it such as to form a larger program,

in or on a volume of a storage or distribution medium, is called an

"aggregate" if the compilation and its resulting copyright are not

used to limit the access or legal rights of the compilation's users

beyond what the individual works permit. Inclusion of a covered work

in an aggregate does not cause this License to apply to the other

parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms

of sections 4 and 5, provided that you also convey the

machine-readable Corresponding Source under the terms of this License,

in one of these ways:

a) Convey the object code in, or embodied in, a physical product

(including a physical distribution medium), accompanied by the

Corresponding Source fixed on a durable physical medium

customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product

(including a physical distribution medium), accompanied by a

written offer, valid for at least three years and valid for as

long as you offer spare parts or customer support for that product

model, to give anyone who possesses the object code either (1) a

copy of the Corresponding Source for all the software in the

product that is covered by this License, on a durable physical

medium customarily used for software interchange, for a price no

more than your reasonable cost of physically performing this

conveying of source, or (2) access to copy the

Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the

written offer to provide the Corresponding Source. This

alternative is allowed only occasionally and noncommercially, and

only if you received the object code with such an offer, in accord

with subsection 6b.

d) Convey the object code by offering access from a designated

place (gratis or for a charge), and offer equivalent access to the

Corresponding Source in the same way through the same place at no

further charge. You need not require recipients to copy the

Corresponding Source along with the object code. If the place to

copy the object code is a network server, the Corresponding Source

may be on a different server (operated by you or a third party)

that supports equivalent copying facilities, provided you maintain

clear directions next to the object code saying where to find the

Corresponding Source. Regardless of what server hosts the

Corresponding Source, you remain obligated to ensure that it is

available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided

you inform other peers where the object code and Corresponding

Source of the work are being offered to the general public at no

charge under subsection 6d.

A separable portion of the object code, whose source code is excluded

from the Corresponding Source as a System Library, need not be

included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any

tangible personal property which is normally used for personal, family,

or household purposes, or (2) anything designed or sold for incorporation

into a dwelling. In determining whether a product is a consumer product,

doubtful cases shall be resolved in favor of coverage. For a particular

product received by a particular user, "normally used" refers to a

typical or common use of that class of product, regardless of the status

of the particular user or of the way in which the particular user

actually uses, or expects or is expected to use, the product. A product

is a consumer product regardless of whether the product has substantial

commercial, industrial or non-consumer uses, unless such uses represent

the only significant mode of use of the product.

"Installation Information" for a User Product means any methods,

procedures, authorization keys, or other information required to install

and execute modified versions of a covered work in that User Product from

a modified version of its Corresponding Source. The information must

suffice to ensure that the continued functioning of the modified object

code is in no case prevented or interfered with solely because

modification has been made.

If you convey an object code work under this section in, or with, or

specifically for use in, a User Product, and the conveying occurs as

part of a transaction in which the right of possession and use of the

User Product is transferred to the recipient in perpetuity or for a

fixed term (regardless of how the transaction is characterized), the

Corresponding Source conveyed under this section must be accompanied

by the Installation Information. But this requirement does not apply

if neither you nor any third party retains the ability to install

modified object code on the User Product (for example, the work has

been installed in ROM).

The requirement to provide Installation Information does not include a

requirement to continue to provide support service, warranty, or updates

for a work that has been modified or installed by the recipient, or for

the User Product in which it has been modified or installed. Access to a

network may be denied when the modification itself materially and

adversely affects the operation of the network or violates the rules and

protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided,

in accord with this section must be in a format that is publicly

documented (and with an implementation available to the public in

source code form), and must require no special password or key for

unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this

License by making exceptions from one or more of its conditions.

Additional permissions that are applicable to the entire Program shall

be treated as though they were included in this License, to the extent

that they are valid under applicable law. If additional permissions

apply only to part of the Program, that part may be used separately

under those permissions, but the entire Program remains governed by

this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option

remove any additional permissions from that copy, or from any part of

it. (Additional permissions may be written to require their own

removal in certain cases when you modify the work.) You may place

additional permissions on material, added by you to a covered work,

for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you

add to a covered work, you may (if authorized by the copyright holders of

that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the

terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or

author attributions in that material or in the Appropriate Legal

Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or

requiring that modified versions of such material be marked in

reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or

authors of the material; or

e) Declining to grant rights under trademark law for use of some

trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that

material by anyone who conveys the material (or modified versions of

it) with contractual assumptions of liability to the recipient, for

any liability that these contractual assumptions directly impose on

those licensors and authors.

All other non-permissive additional terms are considered "further

restrictions" within the meaning of section 10. If the Program as you

received it, or any part of it, contains a notice stating that it is

governed by this License along with a term that is a further

restriction, you may remove that term. If a license document contains

a further restriction but permits relicensing or conveying under this

License, you may add to a covered work material governed by the terms

of that license document, provided that the further restriction does

not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you

must place, in the relevant source files, a statement of the

additional terms that apply to those files, or a notice indicating

where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the

form of a separately written license, or stated as exceptions;

the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly

provided under this License. Any attempt otherwise to propagate or

modify it is void, and will automatically terminate your rights under

this License (including any patent licenses granted under the third

paragraph of section 11).

However, if you cease all violation of this License, then your

license from a particular copyright holder is reinstated (a)

provisionally, unless and until the copyright holder explicitly and

finally terminates your license, and (b) permanently, if the copyright

holder fails to notify you of the violation by some reasonable means

prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is

reinstated permanently if the copyright holder notifies you of the

violation by some reasonable means, this is the first time you have

received notice of violation of this License (for any work) from that

copyright holder, and you cure the violation prior to 30 days after

your receipt of the notice.

Termination of your rights under this section does not terminate the

licenses of parties who have received copies or rights from you under

this License. If your rights have been terminated and not permanently

reinstated, you do not qualify to receive new licenses for the same

material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or

run a copy of the Program. Ancillary propagation of a covered work

occurring solely as a consequence of using peer-to-peer transmission

to receive a copy likewise does not require acceptance. However,

nothing other than this License grants you permission to propagate or

modify any covered work. These actions infringe copyright if you do

not accept this License. Therefore, by modifying or propagating a

covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically

receives a license from the original licensors, to run, modify and

propagate that work, subject to this License. You are not responsible

for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an

organization, or substantially all assets of one, or subdividing an

organization, or merging organizations. If propagation of a covered

work results from an entity transaction, each party to that

transaction who receives a copy of the work also receives whatever

licenses to the work the party's predecessor in interest had or could

give under the previous paragraph, plus a right to possession of the

Corresponding Source of the work from the predecessor in interest, if

the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the

rights granted or affirmed under this License. For example, you may

not impose a license fee, royalty, or other charge for exercise of

rights granted under this License, and you may not initiate litigation

(including a cross-claim or counterclaim in a lawsuit) alleging that

any patent claim is infringed by making, using, selling, offering for

sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this

License of the Program or a work on which the Program is based. The

work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims

owned or controlled by the contributor, whether already acquired or

hereafter acquired, that would be infringed by some manner, permitted

by this License, of making, using, or selling its contributor version,

but do not include claims that would be infringed only as a

consequence of further modification of the contributor version. For

purposes of this definition, "control" includes the right to grant

patent sublicenses in a manner consistent with the requirements of

this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free

patent license under the contributor's essential patent claims, to

make, use, sell, offer for sale, import and otherwise run, modify and

propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express

agreement or commitment, however denominated, not to enforce a patent

(such as an express permission to practice a patent or covenant not to

sue for patent infringement). To "grant" such a patent license to a

party means to make such an agreement or commitment not to enforce a

patent against the party.

If you convey a covered work, knowingly relying on a patent license,

and the Corresponding Source of the work is not available for anyone

to copy, free of charge and under the terms of this License, through a

publicly available network server or other readily accessible means,

then you must either (1) cause the Corresponding Source to be so

available, or (2) arrange to deprive yourself of the benefit of the

patent license for this particular work, or (3) arrange, in a manner

consistent with the requirements of this License, to extend the patent

license to downstream recipients. "Knowingly relying" means you have

actual knowledge that, but for the patent license, your conveying the

covered work in a country, or your recipient's use of the covered work

in a country, would infringe one or more identifiable patents in that

country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or

arrangement, you convey, or propagate by procuring conveyance of, a

covered work, and grant a patent license to some of the parties

receiving the covered work authorizing them to use, propagate, modify

or convey a specific copy of the covered work, then the patent license

you grant is automatically extended to all recipients of the covered

work and works based on it.

A patent license is "discriminatory" if it does not include within

the scope of its coverage, prohibits the exercise of, or is

conditioned on the non-exercise of one or more of the rights that are

specifically granted under this License. You may not convey a covered

work if you are a party to an arrangement with a third party that is

in the business of distributing software, under which you make payment

to the third party based on the extent of your activity of conveying

the work, and under which the third party grants, to any of the

parties who would receive the covered work from you, a discriminatory

patent license (a) in connection with copies of the covered work

conveyed by you (or copies made from those copies), or (b) primarily

for and in connection with specific products or compilations that

contain the covered work, unless you entered into that arrangement,

or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting

any implied license or other defenses to infringement that may

otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or

otherwise) that contradict the conditions of this License, they do not

excuse you from the conditions of this License. If you cannot convey a

covered work so as to satisfy simultaneously your obligations under this

License and any other pertinent obligations, then as a consequence you may

not convey it at all. For example, if you agree to terms that obligate you

to collect a royalty for further conveying from those to whom you convey

the Program, the only way you could satisfy both those terms and this

License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have

permission to link or combine any covered work with a work licensed

under version 3 of the GNU Affero General Public License into a single

combined work, and to convey the resulting work. The terms of this

License will continue to apply to the part which is the covered work,

but the special requirements of the GNU Affero General Public License,

section 13, concerning interaction through a network will apply to the

combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of

the GNU General Public License from time to time. Such new versions will

be similar in spirit to the present version, but may differ in detail to

address new problems or concerns.

Each version is given a distinguishing version number. If the

Program specifies that a certain numbered version of the GNU General

Public License "or any later version" applies to it, you have the

option of following the terms and conditions either of that numbered

version or of any later version published by the Free Software

Foundation. If the Program does not specify a version number of the

GNU General Public License, you may choose any version ever published

by the Free Software Foundation.

If the Program specifies that a proxy can decide which future

versions of the GNU General Public License can be used, that proxy's

public statement of acceptance of a version permanently authorizes you

to choose that version for the Program.

Later license versions may give you additional or different

permissions. However, no additional obligations are imposed on any

author or copyright holder as a result of your choosing to follow a

later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY

APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT

HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY

OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,

THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM

IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF

ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING

WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS

THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY

GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE

USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF

DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD

PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),

EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF

SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided

above cannot be given local legal effect according to their terms,

reviewing courts shall apply local law that most closely approximates

an absolute waiver of all civil liability in connection with the

Program, unless a warranty or assumption of liability accompanies a

copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest

possible use to the public, the best way to achieve this is to make it

free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest

to attach them to the start of each source file to most effectively

state the exclusion of warranty; and each file should have at least

the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify

it under the terms of the GNU General Public License as published by

the Free Software Foundation, either version 3 of the License, or

(at your option) any later version.

This program is distributed in the hope that it will be useful,

but WITHOUT ANY WARRANTY; without even the implied warranty of

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

GNU General Public License for more details.

You should have received a copy of the GNU General Public License

along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short

notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>

This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it

under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate

parts of the General Public License. Of course, your program's commands

might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school,

if any, to sign a "copyright disclaimer" for the program, if necessary.

For more information on this, and how to apply and follow the GNU GPL, see

<http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program

into proprietary programs. If your program is a subroutine library, you

may consider it more useful to permit linking proprietary applications with

the library. If this is what you want to do, use the GNU Lesser General

Public License instead of this License. But first, please read

<http://www.gnu.org/philosophy/why-not-lgpl.html>.