**FFmpeg开源项目编译说明**

**----搭建编译环境、用VS2013编译和调试**

**一、准备工作**

1、本机环境：win7 sp1，74位，vs2013

2、ffmpeg官网上有源代码和现成的静态库和动态库可以调用。如果需要定制ffmpeg的模块，或者跟踪调试和分析研究，则需要进行编译和调试。这里采用的Ffmpeg版本为当前最新版本：2.6.2。

3、ffmpeg本身是linux下的开源项目。它在linux、windows系统中都可以编译。在windows系统，尽量选择VS2013编译工具，是因为VS2013支持大部分C99的特性，基本不需要改动代码，也不需要使用C99转C89的工具。

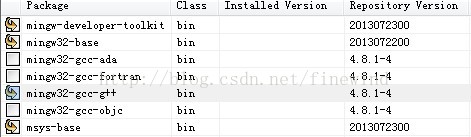
4、ffmpeg是一个开源的多媒体库，使用非常广泛。在linux下编译ffmpeg非常简单，而在windows下编译就不是那么容易了。一般在windows下使用MinGW的gcc toolchain进行编译，这样的话，因为ffmpeg的导出函数均是C风格，因而gcc编译的lib可以被vs链接。但是gcc的debug符号与vs的debug符号（\*.pdb）是无法兼容的，因此我们用vs来开发基于ffmpeg的程序时无法深入ffmpeg内部进行debug

**二、搭建编译环境**

MinGW和yasm是绕不过去的，因为需要使用MinGW来生成config.h，而ffmpeg的汇编语法和VS的不一样，因此需要yasm。具体步骤如下：

1、 下载FFMPEG源码，下载地址：<http://ffmpeg.zeranoe.com/builds/>；（这里也可以下载：静态库static、动态库shared、开发库dev）

2、 下载MinGW安装器，下载地址：<http://www.mingw.org/>；下载完成后安装，安装完成后点运行，标记上以下几项：



然后在Installation菜单下点击Apply Changes（mingw32-gcc-g++也可不选择，因为我们用vs2013编译！）；

**注意：**运行下载的MinGW安装管理器，安装好MinGW，里面已经包含msys。假设安装好后MinGW路径为：C:/MinGW ，则msys路径应为：C:/MinGW/msys 。要将C:\MinGW\bin和C:\MinGW\msys\1.0\bin加到系统path环境变量中（注意：win7的path里面的目录分割符是采用反斜杠的“\”）。

3、 下载yasm.exe，下载地址： <http://yasm.tortall.net/> ；下载后的文件如下图所示：（根据计算机的32位和64位选择，本机为用win64位的yasm-1.2.0-win64.exe）



下载后改名为yasm.exe，再复制到C:/MinGW/msys/1.0/bin目录下；

4、 复制C:/MinGW/msys/1.0/msys.bat 到同目录下，改名叫做msys\_vs2013.bat（这样做是为了保留原来的文件！）。用编辑器打开C:/MinGW/msys/1.0/msys\_vs2013.bat，在此文件的最前面(@echo off之后)添加一行如下内容：

call "C:\Program Files (x86)\Microsoft Visual Studio 12.0\VC\bin\vcvars32.bat" (要与vs2013的实际安装路径一致)；

5、 重命名 C:/MinGW/msys/1.0/bin/link.exe 为link\_renamed.exe (依实际安装选择路径)，这一步是防止这个link.exe与vc的link.exe发生冲突，编译完成后可修改回来。

**三、编译ffmpeg**

1、双击C:/MinGW/msys/1.0/msys\_vs2013.bat，运行shell（linux的shell）：转到FFMPEG源代码根目录下，可运行 ./configure --help查看编译配置选项（用./configure –help>>lisq.txt，在lisq.txt中查看方便一些），可以使用的命令及选项为：

静态库：./configure  --enable-static   --prefix=./vs2013\_build  --enable-debug --toolchain=msvc   
动态库：./configure  --enable-shared  --prefix=./vs2013\_build  --enable-debug --toolchain=msvc

等待配置完成返回（大约两分钟）；

其他常用选项：--enable-avresample

2、 输入 make编译；(提示：make clean是清理项目文件) （大约七八分钟）

3、 输入make install安装。（大约一分钟）

如果静态库的configure配置，完成后，生成编译好的ffmpe库，有4个目录，bin、include、lib、share 。生成的头文件（\*.h）及库（\*.a，这是带有调试信息的静态库，可以改为\*.lib）已经在ffmpeg源代码下的vs2013\_build目录下，使用这个库，即可在VS下编译，且可以调试单步进入FFMPEG函数的内部，跟踪代码的执行情况。

如果是动态库配置，再编译和安装，则bin目录里面有\*.lib和\*.dll，在lib目录里面有\*.def导出文件。开发阶段只需要工程中包含include和lib，运行阶段需要对应dll（注意两部分版本要一致）.

**四、我的实例：验证VS2013编译和调试ffmpeg内部库函数**

在 vs2013\_build/share/ffmpeg/examples 目录下有若干个示例。

把 avio\_reading.c 添加到我们的test\_ffmpeg工程（vs2013），配置include与lib路径、依赖库。在链接库中加入文件（静态库调用ffmpeg）：

ws2\_32.lib;libavcodec.a;libavdevice.a;libavfilter.a;libavformat.a;libavutil.a;libswresample.a;libswscale.a

编译，链接，运行，OK！---》》可以加断点和跟踪调试，进入ffmpeg函数内部，实际是跟踪到了ffmpeg源代码目录下面对应的那些\*.c文件了！！

**Ffmpeg各模块的功能**

FFmpeg是一套可以用来记录、转换数字音频、视频，并能将其转化为流的开源计算机程序。它包括了领先的音/视频编码库libavcodec等。

**libavformat**：用于各种音视频[封装格式](http://baike.baidu.com/view/1942911.htm)的生成和解析，包括获取解码所需信息以生成解码上下文结构

和读取音视频帧等功能；

**libavcodec**：用于各种类型声音/[图像](http://baike.baidu.com/view/42116.htm)编解码；

**libavutil**：包含一些公共的工具函数；

**libswscale**：用于视频场景比例缩放、色彩映射转换；

**libpostproc**：用于后期效果处理；

**ffmpeg**：该项目提供的一个工具，可用于格式转换、解码或[电视卡](http://baike.baidu.com/view/44687.htm)即时编码等；

**ffsever**：一个 HTTP 多媒体即时广播串流服务器；

**ffplay**：是一个简单的播放器，使用ffmpeg 库解析和解码，通过SDL显示；

# [ffmpeg编译选项汇总](http://www.cnblogs.com/wainiwann/p/4204230.html)

===========================================================

在控制台输入如下命令可看到 ffmpeg 详细的编译选项。

[root@localhost ffmpeg]# ./configure --help

Usage: configure [options]

Options: [defaults in brackets after descriptions]

# Help options:

--help print this message

--list-decoders show all available decoders

--list-encoders show all available encoders

--list-hwaccels show all available hardware accelerators

--list-demuxers show all available demuxers

--list-muxers show all available muxers

--list-parsers show all available parsers

--list-protocols show all available protocols

--list-bsfs show all available bitstream filters

--list-indevs show all available input devices

--list-outdevs show all available output devices

--list-filters show all available filters

# Standard options:

--logfile=FILE log tests and output to FILE [config.log]

--disable-logging do not log configure debug information

--fatal-warnings fail if any configure warning is generated

--prefix=PREFIX install in PREFIX []

--bindir=DIR install binaries in DIR [PREFIX/bin]

--datadir=DIR install data files in DIR [PREFIX/share/ffmpeg]

--docdir=DIR install documentation in DIR [PREFIX/share/doc/ffmpeg]

--libdir=DIR install libs in DIR [PREFIX/lib]

--shlibdir=DIR install shared libs in DIR [LIBDIR]

--incdir=DIR install includes in DIR [PREFIX/include]

--mandir=DIR install man page in DIR [PREFIX/share/man]

--enable-rpath use rpath to allow installing libraries in paths

not part of the dynamic linker search path

use rpath when linking programs [USE WITH CARE]

# Licensing options:

--enable-gpl allow use of GPL code, the resulting libs

and binaries will be under GPL [no]

--enable-version3 upgrade (L)GPL to version 3 [no]

--enable-nonfree allow use of nonfree code, the resulting libs

and binaries will be unredistributable [no]

# Configuration options:

--disable-static do not build static libraries [no]

--enable-shared build shared libraries [no]

--enable-small optimize for size instead of speed

--disable-runtime-cpudetect disable detecting cpu capabilities at runtime (smaller binary)

--enable-gray enable full grayscale support (slower color)

--disable-swscale-alpha disable alpha channel support in swscale

--disable-all disable building components, libraries and programs

--enable-incompatible-libav-abi enable incompatible Libav fork ABI [no]

--enable-raise-major increase major version numbers in sonames [no]

# Program options:

--disable-programs do not build command line programs

--disable-ffmpeg disable ffmpeg build

--disable-ffplay disable ffplay build

--disable-ffprobe disable ffprobe build

--disable-ffserver disable ffserver build

# Documentation options:

--disable-doc do not build documentation

--disable-htmlpages do not build HTML documentation pages

--disable-manpages do not build man documentation pages

--disable-podpages do not build POD documentation pages

--disable-txtpages do not build text documentation pages

# Component options:

--disable-avdevice disable libavdevice build

--disable-avcodec disable libavcodec build

--disable-avformat disable libavformat build

--disable-avutil disable libavutil build

--disable-swresample disable libswresample build

--disable-swscale disable libswscale build

--disable-postproc disable libpostproc build

--disable-avfilter disable libavfilter build

--enable-avresample enable libavresample build [no]

--disable-pthreads disable pthreads [autodetect]

--disable-w32threads disable Win32 threads [autodetect]

--disable-os2threads disable OS/2 threads [autodetect]

--disable-network disable network support [no]

--disable-dct disable DCT code

--disable-dwt disable DWT code

--disable-error-resilience disable error resilience code

--disable-lsp disable LSP code

--disable-lzo disable LZO decoder code

--disable-mdct disable MDCT code

--disable-rdft disable RDFT code

--disable-fft disable FFT code

--disable-faan disable floating point AAN (I)DCT code

--disable-pixelutils disable pixel utils in libavutil

# Hardware accelerators:

--disable-dxva2 disable DXVA2 code [autodetect]

--disable-vaapi disable VAAPI code [autodetect]

--disable-vda disable VDA code [autodetect]

--disable-vdpau disable VDPAU code [autodetect]

# Individual component options:

--disable-everything disable all components listed below

--disable-encoder=NAME disable encoder NAME

--enable-encoder=NAME enable encoder NAME

--disable-encoders disable all encoders

--disable-decoder=NAME disable decoder NAME

--enable-decoder=NAME enable decoder NAME

--disable-decoders disable all decoders

--disable-hwaccel=NAME disable hwaccel NAME

--enable-hwaccel=NAME enable hwaccel NAME

--disable-hwaccels disable all hwaccels

--disable-muxer=NAME disable muxer NAME

--enable-muxer=NAME enable muxer NAME

--disable-muxers disable all muxers

--disable-demuxer=NAME disable demuxer NAME

--enable-demuxer=NAME enable demuxer NAME

--disable-demuxers disable all demuxers

--enable-parser=NAME enable parser NAME

--disable-parser=NAME disable parser NAME

--disable-parsers disable all parsers

--enable-bsf=NAME enable bitstream filter NAME

--disable-bsf=NAME disable bitstream filter NAME

--disable-bsfs disable all bitstream filters

--enable-protocol=NAME enable protocol NAME

--disable-protocol=NAME disable protocol NAME

--disable-protocols disable all protocols

--enable-indev=NAME enable input device NAME

--disable-indev=NAME disable input device NAME

--disable-indevs disable input devices

--enable-outdev=NAME enable output device NAME

--disable-outdev=NAME disable output device NAME

--disable-outdevs disable output devices

--disable-devices disable all devices

--enable-filter=NAME enable filter NAME

--disable-filter=NAME disable filter NAME

--disable-filters disable all filters

# External library support:

--enable-avisynth enable reading of AviSynth script files [no]

--disable-bzlib disable bzlib [autodetect]

--enable-fontconfig enable fontconfig, useful for drawtext filter [no]

--enable-frei0r enable frei0r video filtering [no]

--enable-gnutls enable gnutls, needed for https support

if openssl is not used [no]

--disable-iconv disable iconv [autodetect]

--enable-ladspa enable LADSPA audio filtering [no]

--enable-libaacplus enable AAC+ encoding via libaacplus [no]

--enable-libass enable libass subtitles rendering,

needed for subtitles and ass filter [no]

--enable-libbluray enable BluRay reading using libbluray [no]

--enable-libbs2b enable bs2b DSP library [no]

--enable-libcaca enable textual display using libcaca [no]

--enable-libcelt enable CELT decoding via libcelt [no]

--enable-libcdio enable audio CD grabbing with libcdio [no]

--enable-libdc1394 enable IIDC-1394 grabbing using libdc1394

and libraw1394 [no]

--enable-libfaac enable AAC encoding via libfaac [no]

--enable-libfdk-aac enable AAC de/encoding via libfdk-aac [no]

--enable-libflite enable flite (voice synthesis) support via libflite [no]

--enable-libfreetype enable libfreetype, needed for drawtext filter [no]

--enable-libfribidi enable libfribidi, improves drawtext filter [no]

--enable-libgme enable Game Music Emu via libgme [no]

--enable-libgsm enable GSM de/encoding via libgsm [no]

--enable-libiec61883 enable iec61883 via libiec61883 [no]

--enable-libilbc enable iLBC de/encoding via libilbc [no]

--enable-libmfx enable HW acceleration through libmfx

--enable-libmodplug enable ModPlug via libmodplug [no]

--enable-libmp3lame enable MP3 encoding via libmp3lame [no]

--enable-libnut enable NUT (de)muxing via libnut,

native (de)muxer exists [no]

--enable-libopencore-amrnb enable AMR-NB de/encoding via libopencore-amrnb [no]

--enable-libopencore-amrwb enable AMR-WB decoding via libopencore-amrwb [no]

--enable-libopencv enable video filtering via libopencv [no]

--enable-libopenh264 enable H.264 encoding via OpenH264 [no]

--enable-libopenjpeg enable JPEG 2000 de/encoding via OpenJPEG [no]

--enable-libopus enable Opus de/encoding via libopus [no]

--enable-libpulse enable Pulseaudio input via libpulse [no]

--enable-libquvi enable quvi input via libquvi [no]

--enable-librtmp enable RTMP[E] support via librtmp [no]

--enable-libschroedinger enable Dirac de/encoding via libschroedinger [no]

--enable-libshine enable fixed-point MP3 encoding via libshine [no]

--enable-libsmbclient enable Samba protocol via libsmbclient [no]

--enable-libsoxr enable Include libsoxr resampling [no]

--enable-libspeex enable Speex de/encoding via libspeex [no]

--enable-libssh enable SFTP protocol via libssh [no]

--enable-libstagefright-h264 enable H.264 decoding via libstagefright [no]

--enable-libtheora enable Theora encoding via libtheora [no]

--enable-libtwolame enable MP2 encoding via libtwolame [no]

--enable-libutvideo enable Ut Video encoding and decoding via libutvideo [no]

--enable-libv4l2 enable libv4l2/v4l-utils [no]

--enable-libvidstab enable video stabilization using vid.stab [no]

--enable-libvo-aacenc enable AAC encoding via libvo-aacenc [no]

--enable-libvo-amrwbenc enable AMR-WB encoding via libvo-amrwbenc [no]

--enable-libvorbis enable Vorbis en/decoding via libvorbis,

native implementation exists [no]

--enable-libvpx enable VP8 and VP9 de/encoding via libvpx [no]

--enable-libwavpack enable wavpack encoding via libwavpack [no]

--enable-libwebp enable WebP encoding via libwebp [no]

--enable-libx264 enable H.264 encoding via x264 [no]

--enable-libx265 enable HEVC encoding via x265 [no]

--enable-libxavs enable AVS encoding via xavs [no]

--enable-libxcb enable X11 grabbing using XCB [autodetect]

--enable-libxcb-shm enable X11 grabbing shm communication [autodetect]

--enable-libxcb-xfixes enable X11 grabbing mouse rendering [autodetect]

--enable-libxcb-shape enable X11 grabbing shape rendering [autodetect]

--enable-libxvid enable Xvid encoding via xvidcore,

native MPEG-4/Xvid encoder exists [no]

--enable-libzmq enable message passing via libzmq [no]

--enable-libzvbi enable teletext support via libzvbi [no]

--disable-lzma disable lzma [autodetect]

--enable-decklink enable Blackmagick DeckLink I/O support [no]

--enable-nvenc enable NVIDIA NVENC support [no]

--enable-openal enable OpenAL 1.1 capture support [no]

--enable-opencl enable OpenCL code

--enable-opengl enable OpenGL rendering [no]

--enable-openssl enable openssl, needed for https support

if gnutls is not used [no]

--disable-sdl disable sdl [autodetect]

--enable-x11grab enable X11 grabbing (legacy) [no]

--disable-xlib disable xlib [autodetect]

--disable-zlib disable zlib [autodetect]

# Toolchain options:

--arch=ARCH select architecture []

--cpu=CPU select the minimum required CPU (affects

instruction selection, may crash on older CPUs)

--cross-prefix=PREFIX use PREFIX for compilation tools []

--progs-suffix=SUFFIX program name suffix []

--enable-cross-compile assume a cross-compiler is used

--sysroot=PATH root of cross-build tree

--sysinclude=PATH location of cross-build system headers

--target-os=OS compiler targets OS []

--target-exec=CMD command to run executables on target

--target-path=DIR path to view of build directory on target

--target-samples=DIR path to samples directory on target

--tempprefix=PATH force fixed dir/prefix instead of mktemp for checks

--toolchain=NAME set tool defaults according to NAME

--nm=NM use nm tool NM [nm -g]

--ar=AR use archive tool AR [ar]

--as=AS use assembler AS []

--windres=WINDRES use windows resource compiler WINDRES [windres]

--yasmexe=EXE use yasm-compatible assembler EXE [yasm]

--cc=CC use C compiler CC [gcc]

--cxx=CXX use C compiler CXX [g++]

--dep-cc=DEPCC use dependency generator DEPCC [gcc]

--ld=LD use linker LD []

--pkg-config=PKGCONFIG use pkg-config tool PKGCONFIG [pkg-config]

--pkg-config-flags=FLAGS pass additional flags to pkgconf []

--ranlib=RANLIB use ranlib RANLIB [ranlib -D]

--doxygen=DOXYGEN use DOXYGEN to generate API doc [doxygen]

--host-cc=HOSTCC use host C compiler HOSTCC

--host-cflags=HCFLAGS use HCFLAGS when compiling for host

--host-cppflags=HCPPFLAGS use HCPPFLAGS when compiling for host

--host-ld=HOSTLD use host linker HOSTLD

--host-ldflags=HLDFLAGS use HLDFLAGS when linking for host

--host-libs=HLIBS use libs HLIBS when linking for host

--host-os=OS compiler host OS []

--extra-cflags=ECFLAGS add ECFLAGS to CFLAGS []

--extra-cxxflags=ECFLAGS add ECFLAGS to CXXFLAGS []

--extra-ldflags=ELDFLAGS add ELDFLAGS to LDFLAGS []

--extra-ldexeflags=ELDFLAGS add ELDFLAGS to LDEXEFLAGS []

--extra-libs=ELIBS add ELIBS []

--extra-version=STRING version string suffix []

--optflags=OPTFLAGS override optimization-related compiler flags

--build-suffix=SUFFIX library name suffix []

--enable-pic build position-independent code

--enable-thumb compile for Thumb instruction set

--enable-lto use link-time optimization

# Advanced options (experts only):

--malloc-prefix=PREFIX prefix malloc and related names with PREFIX

--disable-symver disable symbol versioning

--enable-hardcoded-tables use hardcoded tables instead of runtime generation

--disable-safe-bitstream-reader

disable buffer boundary checking in bitreaders

(faster, but may crash)

--enable-memalign-hack emulate memalign, interferes with memory debuggers

--sws-max-filter-size=N the max filter size swscale uses [256]

# Optimization options (experts only):

--disable-asm disable all assembly optimizations

--disable-altivec disable AltiVec optimizations

--disable-amd3dnow disable 3DNow! optimizations

--disable-amd3dnowext disable 3DNow! extended optimizations

--disable-mmx disable MMX optimizations

--disable-mmxext disable MMXEXT optimizations

--disable-sse disable SSE optimizations

--disable-sse2 disable SSE2 optimizations

--disable-sse3 disable SSE3 optimizations

--disable-ssse3 disable SSSE3 optimizations

--disable-sse4 disable SSE4 optimizations

--disable-sse42 disable SSE4.2 optimizations

--disable-avx disable AVX optimizations

--disable-xop disable XOP optimizations

--disable-fma3 disable FMA3 optimizations

--disable-fma4 disable FMA4 optimizations

--disable-avx2 disable AVX2 optimizations

--disable-armv5te disable armv5te optimizations

--disable-armv6 disable armv6 optimizations

--disable-armv6t2 disable armv6t2 optimizations

--disable-vfp disable VFP optimizations

--disable-neon disable NEON optimizations

--disable-inline-asm disable use of inline assembly

--disable-yasm disable use of nasm/yasm assembly

--disable-mipsdspr1 disable MIPS DSP ASE R1 optimizations

--disable-mipsdspr2 disable MIPS DSP ASE R2 optimizations

--disable-mipsfpu disable floating point MIPS optimizations

--disable-fast-unaligned consider unaligned accesses slow

# Developer options (useful when working on FFmpeg itself):

--disable-debug disable debugging symbols

--enable-debug=LEVEL set the debug level []

--disable-optimizations disable compiler optimizations

--enable-extra-warnings enable more compiler warnings

--disable-stripping disable stripping of executables and shared libraries

--assert-level=level 0(default), 1 or 2, amount of assertion testing,

2 causes a slowdown at runtime.

--enable-memory-poisoning fill heap uninitialized allocated space with arbitrary data

--valgrind=VALGRIND run "make fate" tests through valgrind to detect memory

leaks and errors, using the specified valgrind binary.

Cannot be combined with --target-exec

--enable-ftrapv Trap arithmetic overflows

--samples=PATH location of test samples for FATE, if not set use

$FATE\_SAMPLES at make invocation time.

--enable-neon-clobber-test check NEON registers for clobbering (should be

used only for debugging purposes)

--enable-xmm-clobber-test check XMM registers for clobbering (Win64-only;

should be used only for debugging purposes)

--enable-random randomly enable/disable components

--disable-random

--enable-random=LIST randomly enable/disable specific components or

--disable-random=LIST component groups. LIST is a comma-separated list

of NAME[:PROB] entries where NAME is a component

(group) and PROB the probability associated with

NAME (default 0.5).

--random-seed=VALUE seed value for --enable/disable-random

NOTE: Object files are built at the place where configure is launched.

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