

[首页](#) [资讯](#) [精华](#) [论坛](#) [问答](#) [博客](#) [专栏](#) [群组](#) [更多 ▼](#)

[您还未登录！](#) [登录](#) [注册](#)

## xhc66

- [博客](#)
- [微博](#)
- [相册](#)
- [收藏](#)
- [留言](#)
- [关于我](#)



### VLC简介及使用说明

博客分类：

- [VLC、QuickTimeWeb插件相关](#)

[LinuxWindows网络应用应用服务器Git](#)

#### 一、简介

VLC的全名是Video Lan Client，是一个开源的、跨平台的视频播放器。VLC支持大量的音视频传输、封装和编码格式，完整的功能特性列表可以在这里获得

<http://www.videolan.org/vlc/features.html>，下面给出一个简要的不完整的列表：

操作系统：Windows、WinCE、Linux、MacOSX、BEOS、BSD

访问形式：文件、DVD/VCD/CD、http、ftp、mms、TCP、UDP、RTP、IP组播、IPv6、rtsp

编码格式：MPEG\*、DIVX、WMV、MOV、3GP、FLV、H.263、H.264、FLAC

视频字幕：DVD、DVB、Text、Vobsub

视频输出：DirectX、X11、XVideo、SDL、FrameBuffer、ASCII

控制界面：WxWidgets、QT4、Web、Telnet、Command line

浏览器插件：ActiveX、Mozilla (firefox)

实际上为了更清晰一点，我们可以反过来说说VLC不支持哪些常见的。首先是RealVideo（Real的Audio部分支持），因为Real的Video加码器存在版权的问题。实际上VLC 0.9.0已经加入了RealVideo的支持，但是需要额外的解码器（类似MPlayer）。另外，VLC不支持3GP的音频格式AMR。

VLC原先是几个法国的大学生做的项目，后来他们把VLC作为了一个开源的项目，吸引了来自世界各国的很多优秀程序员来共同编写和维护VLC，才逐渐变成了现在这个样子。

至于为什么叫VideoLan Client，是因为以前还有一个VideoLan Server的项目（简称VLS），而目前VLS的功能已经合并到VLC中来，所以VLC不仅仅是一个视频播放器，它也可以作为小型的视频服务器，更可以一边播放一边转码，把视频流发送到网络上。VLC最为突出的就是网络流的播放功能，例如MPEG2的UDP TS流的播放和转发，几乎是无可替代的。

对普通用户来说，VLC还有一个好处是不影响Windows中的解码器。VLC通常不影响也不依赖于系统中自带的解码器（除了realvideo和quicktime的类型），很绿色很环保；更不用担心流氓软件、广告插件之类的恶心的玩意儿。

从程序结构来看，VLC的可扩展性是相当优秀的。VLC绝大部分用高效的C代码来编写（少量的C++和汇编），但是实现了完全动态的模块化，所有功能包括程序框架本身都是module，

可以在运行时载入，这使得VLC可以轻易的扩展多种功能并且容易维护。它的架构有一点类似于DirectShow的技术。

VLC也很注重版权方面的问题，你可以放心的自由的使用而不需要为版权的问题担心——VLC只包括免费的、自由的库。VLC基于GPL，因此也可以用于商业应用，只需要保留GPL，以及公开源代码，如果你修改了VLC的话。

下面是VLC相关的一些链接

VLC官方网站：<http://www.videolan.org/>

VLC下载页面：<http://www.videolan.org/vlc/>

VLC下载目录（源码和安装包）：<http://download.videolan.org/pub/videolan/vlc/>

VLC Nightly Builds：<http://nightlies.videolan.org/>

VLC 开发Wiki：[http://wiki.videolan.org/Developers\\_Corner](http://wiki.videolan.org/Developers_Corner)

VLC Win32第三方库预编译包下载目录：<http://download.videolan.org/pub/testing/win32/>

VLC 官方论坛：<http://forum.videolan.org/>

VLC 邮件列表：<http://www.videolan.org/developers/lists.html>

## 二、Windows平台编译

第一步，要创建编译VLC的环境。VLC在Windows下可以用Msys+MingW 或者Cygwin的方式来编译，二者大同小异，这里主要介绍Cygwin。

Cygwin是一个在Windows下面模拟Linux环境的东西。它提供了很多库和应用程序，让你可以像在Linux上来使用Windows。你需要在<http://www.cygwin.com/> 下载一个cygwin的安装程序setup.exe。然后选择一个cygwin的原来下载所需要的库，国内的话到 <ftp://ftp.sjtu.edu.cn/cygwin/>和 <http://www.cygwin.cn/pub> 相对来说会快一点。第一次安装，除了默认的库以外，我们还需要加入一些库来支持VLC的编译，包括

Archive （目录）

unzip

zip

Devel

autoconf

automake

binutils

cvs

expat

gcc

gcc-core

gcc-g++

gcc-mingw

gcc-mingw-core

gcc-mingw-g++

gdb （如果你需要调试的话）

gettext

gettext-devel

git （不一定需要）

libiconv

libtool

make

mingw-runtime

nasm

patchutils

pkg-config

```

subversion
Editor
vim (有了它方便点)
Libs
libgcrypt
Web //这两个可以不要, 如果你不想编译第三方库
curl (optional: for building extras/contrib)
wget (optional: for building extras/contrib)

```

然后是下载所需要版本的VLC的源代码。对于Release版本, 可以从这里下载：  
<http://download.videolan.org/pub/videolan/vlc/>, 如果你需要下载实时最新的库, 就要从VLC的源码库上取下来了。现在VLC改用git了, 使用命令: `git clone git://git.videolan.org/vlc.git`

VLC还需要很多第三方的库, 你可以取得源码来编译, 这部分可以参考Linux上的VLC编译的文章。在Windows上VLC做了一个编译好的win32第三方库的包, 可以从<http://download.videolan.org/pub/testing/win32/> 下载。包需要解压到根目录 '/', 事实上这些库都位于 '/usr/win32-branch' 里边。

在cygwin中进入vlc的主目录, 运行bootstrap, './bootstrap'。在等待它结束之前, 我们先来看一下如何配置VLC的编译。运行 './configure -h >cfg\_opt.txt' 就可以把配置的选项信息都输出到cfg\_opt.txt中, 然后慢慢来研究。里面大多是启用、禁用某些特性和功能模块, 或者配置某些模块需要的库的路径等信息。为了方便可以写一个简单的脚本来做这件事, 当然这个不是必需的。

```

PATH=/usr/win32-branch/bin:$PATH \
PKG_CONFIG_PATH=/usr/win32-branch/lib/pkgconfig \
CPPFLAGS="-I/usr/win32-branch/include -I/usr/win32-branch/include/ebml" \
LDFLAGS=-L/usr/win32-branch/lib \
CC="gcc -mno-cygwin" CXX="g++ -mno-cygwin" \
./configure \
    -host=i686-pc-mingw32 \
    -disable-gtk \
    -enable-nls -enable-sdl --with-sdl-config-path=/usr/win32-branch/bin \
    -enable-ffmpeg --with-ffmpeg-mp3lame --with-ffmpeg-faac \
    -with-ffmpeg-zlib -enable-faad -enable-flac -enable-theora \
    --with-wx-config-path=/usr/win32-branch/bin \
    --with-freetype-config-path=/usr/win32-branch/bin \
    --with-ribidi-config-path=/usr/win32-branch/bin \
    -enable-live555 --with-live555-tree=/usr/win32-branch/live.com \
    -enable-caca --with-caca-config-path=/usr/win32-branch/bin \
    --with-xml2-config-path=/usr/win32-branch/bin \
    --with-dvdnav-config-path=/usr/win32-branch/bin \
    -disable-cddax -disable-vcdx -enable-goom \
    -enable-twolame -enable-dvdrread \
    -enable-release -enable-dca \
    -enable-mkv \
    -enable-quicktime -enable-mozilla \
    --with-mozilla-sdk-path=/usr/win32-branch/gecko-sdk \
    -enable-mostly-builtin \
    && make

```

例如把这个文件保存为‘mybuild’，等bootstrap运行结束之后，我们只需要运行一下 mybuild就会开始配置和编译了。这通常需要挺长的时间，如果运气好，在漫长的等待之后make运行成功，vlc就编译好了。直接双击主目录里边的vlc.exe应该就可以运行了。

为了方便可以把它打包，最简单的是‘make package-win32-base’，它将创建一个子目录来存放所有运行所需的东西，这个目录就是一个绿色版的VLC啦。

当然如果你运气没那么棒，可能会碰到各种各样的错误，例如在bootstrap或者configure中出错，通常是因为没安装某个库或者库的版本不合适；也可能碰到编译的错误，尝试去google一下。

本文参考 <http://wiki.videolan.org/Win32CompileCygwinNew>

### 三、Linux平台编译

先下载源代码，把源代码解压之后也是先运行一下bootstrap，看看缺什么工具没有，例如gcc、make、libtool、automake、autoconfig等是否有合适的版本。如果不合适的话就用你喜欢的方式去升级或者安装。

下面最麻烦的一步来了。VLC在Linux上没有给出和Windows上类似的第三方库的预编译包，你必须去自己获得并编译这些库。这些库的数量取决于你的配置选项。VLC给出了一个自动化的解决方案，关注一下主目录\extras\contrib，里边有一些工具来帮助你自动下载、patch和编译所有可能用到的第三方库。先运行一下那个目录下的bootstrap，如果缺少\*\*\*.mak的话可以到VLC的代码库中找一下，是一个系统相关的文件，然后简单的一句make就可以了。如果你对 these 库比较熟悉的话最好先手动筛选一下，有一些可能不是你所需要的，而下载他们可能需要很长的时间。

在漫长的等待之后，如果一切顺利（我几乎不相信会这样），这一步就算完成了。可能遇到的问题有：下载实在太慢，可以用其他下载工具把库的代码包下下来放到\extras\contrib\src里边；如果下载一半断掉，可以把那个不完整的文件删掉重新再运行make。

这一步做完之后和windows上的编译就几乎一样了。写一个配置脚本然后运行它。例如一个配置脚本：

```
./configure \  
-enable-release \  
-disable-skins2 \  
-disable-wxwidgets \  
-enable-mozilla \  
-with-mozilla-sdk-path=./gecko-sdk \  
-disable-sout \  
-disable-httpd \  
-enable-live555 \  
-disable-dvdnav \  
-disable-libcdio \  
-disable-libcddb \  
-disable-cdda \  
-disable-vcd \  
-disable-dvdrread \  
-disable-smb \  
-disable-cmml \  
-disable-alsa \  
-disable-opengl \  
-disable-png \  
-disable-screen \  

```

```
-disable-mkv \  
-disable-mod \  
-disable-mpc \  
-disable-libtar \  
-disable-speex \  
-disable-visual \  
-disable-daap \  
-disable-bonjour \  
-disable-gnutls \  
-enable-faad \  
-enable-mostly-builtin
```

配置完之后再运行make就可以了。

参考：<http://wiki.videolan.org/UnixCompile>

#### 四、使用

VLC的功能很强大，它不仅仅是一个视频播放器，也可作为小型的视频服务器，更可以一边播放一边转码，把视频流发送到网络上。

最简单的，从界面打开一个文件播放，也可以在命令行下使用，如C:\Program Files\VideoLAN\VLC>vlc.exe test.ts

获取内置的帮助，会写到vlc-help.txt：C:\Program Files\VideoLAN\VLC>vlc.exe -h

获取更详细的帮助，有大量的选项介绍：C:\Program Files\VideoLAN\VLC>vlc.exe -H

在线使用帮助

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

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

搜索并显示可用的模块列表：C:\Program Files\VideoLAN\VLC>vlc.exe -l

Windows下，默认情况，VLC的配置文件在 %APPDATA%\vlc\vlcrc，%APPDATA%在XP下通常是 C:\Documents and Settings\用户名\Application Data

Linux下，在用户home目录的/.vlc/中

vlc的选项完全可以通过修改vlcrc来设置，界面只是略微方便一点

重置所有选项到默认状态：C:\Program Files\VideoLAN\VLC>vlc.exe --reset-config

VLC从vlcrc中指定的plugin-path，以及当前目录的modules和plugins目录来递归查找plugin，VLC的大部分功能都是通过plugin来实现的。VLC默认有大量的动态插件，为了加快启动速度，vlc会在%APPDATA%\vlc\cache中缓存plugin的列表，选项plugins-cache=0可以禁止缓存plugin

打开一个UDP组播流，组播组 239.255.1.1，端口 4567，默认端口1234，对于rtp协议，VLC会自动识别，写udp还是rtp都没问题

C:\Program Files\VideoLAN\VLC>vlc.exe udp://@239.255.1.1:4567

在本地UDP端口 888 接收流，“@”表示绑定一个地址，而不是连接该地址

C:\Program Files\VideoLAN\VLC>vlc.exe udp://@888

串流输出，就是在播放的时候，以某种方式向外传送视频，在打开界面的串流/保存 中设置会比较方便。

例如，循环播放test.ts，以rtp方式传送到224.1.1.1端口1234,同时显示视频

vlc.exe test.ts --loop :sout=#duplicate{dst=std{access=rtp,mux=ts,dst=224.1.1.1:1234},dst=display}

例如，接收UDP端口888，数据全部保存到C:\dump.ts

```
vlc.exe udp://@888 :demux=dump :demuxdump-file="C:\dump.ts"
```

关于Windows下视频输出模块

**Direct3D**：效果比DirectX差一点，但是方便截图，也可以在图像上实现alpha

**DirectX**：效果最好，利用DirectX的显示加速

**OpenGL**：在不同的硬件上表现不太一样

**WinGDI**：最慢的一种，不需要显卡加速

**caca**：用彩色的Assic字符来显示

临时启用某个视频输出，可以这样：C:\Program Files\VideoLAN\VLC-0.8.6e>vlc test.ts -vout=caca

**ActiveX控件**

官方发布的VLC自带ActiveX控件 axvlc.dll，注册之后可以方便的在应用程序和网页中使用VLC，注册的办法是

```
C:\Program Files\VideoLAN\VLC-0.8.6e>regsvr32 axvlc.dll
```

ActiveX VLC的使用方法可以参考源代码中ActiveX目录的README.TXT和test.html，ActiveX控件的接口有第一版和第二版，第一版简单，功能少，已经不再维护建议用第二版本，功能多一点

**Mozilla Plugin**

你还可以在Windows和Linux的Firefox中使用VLC。Windows下可以在安装VLC的时候选上Mozilla plugin，事实上它做的就是HKLM\_Software\_MozillaPlugin键下添加一个VLC的子键。Linux下可以把libvlcplugin.so（或者叫npvlc.so）和插件目录放到Firefox的plugins目录，来使之生效。

同样这适用于基于Xulrunner的应用程序，事实上像 Miro、SongBird等xul应用都是用的这个插件。与ActiveX类似的，VLC的Mozilla Plugin也有两套接口，建议用新的第二版。

**注册表**

在Windows下VLC只使用很少量的注册表的信息，最重要的一条是HKLM\_Software\_VLC键下的InstallDir项，VLC的Mozilla Plugin和ActiveX控件通过这一项来定位其插件的目录。Telnet、Http控制对VLC来说，控制界面都是Interface类的模块，你可以使用各种控制模块。Windows下默认使用的是wxwidgets图形界面，还可以使用http、telnet等界面，来远程控制VLC，如果用VLC在服务器上专门作视频流转发一类的事情，这种远程界面可以帮上大忙。

**视频过滤器**

0.8系列中有多达13种视频过滤器，为视频添加各种效果和特殊处理。

**Skin2界面**

一个漂亮一点的，可以换肤的界面。

**其他的**

播放DVD、VCD光盘，打开DirectShow设备，播放当前屏幕...

**一些问题**

目前的VLC无法播放Real视频，如rm、rmvb。其实从0.9版本的VLC开始可以依赖其他解码器播放。但就目前测试的情况来看，顺序播放没问题，但是拖动则效果很差。RTSP的播放，拖动有问题，在获取播放位置时有缺陷。例如和达尔文服务器的配合。

## 五、视频播放的基本原理

几乎所有的视频播放器，如VLC、MPlayer、Xine，包括DirectShow，在播放视频的原理和架构上都是非常相似的，理解这个对理解VLC的源码会有事半功倍的效果。

大致的来说，播放一个视频分为4个步骤：

1. **access** 访问，或者理解为接收、获取、得到
2. **demux** 解复用，就是把通常合在一起的音频和视频分离(还有可能的字幕)
3. **decode** 解码，包括音频和视频的解码
4. **output** 输出，也分为音频和视频的输出（**aout**和**vout**）

拿播放一个UDP组播的MPEG TS流来说吧，**access**部分负责从网络接收组播流，放到VLC的内存缓冲区中，**access**模块关注IP协议，如是否IPv6、组播地址、组播协议、端口等信息；如果检测出来是RTP协议（RTP协议在UDP头部简单得加上了固定12个字节的信息），还要分析RTP头部信息。这部分可以参看VLC源码 `/modules/access/udp.c`。在同目录下还可以看到大量的**access**模块，如**file**、**http**、**dvd**、**ftp**、**smb**、**tcp**、**dshow**、**mms**、**v4l**...等等

而**demux**部分首先要解析TS流的信息。TS格式是MPEG2协议的一部分，概括地说，TS通常是固定188字节的一个**packet**，一个TS流可以包含多个**program**（节目），一个**program**又可以包含多个视频、音频、和文字信息的ES流；每个ES流会有不同的PID标示。而又为了可以分析这些ES流，TS有一些固定的PID用来间隔发送**program**和es流信息的表格：**PAT**和**PMT**表。关于TS格式的详细信息可以去google一下。VLC专门做了一个独立的库**libdvbpsi**来解析和编码TS流，而调用它的代码可以参见VLC源码 `/modules/demux/ts.c`。

其实之所以需要**demux**，是因为音视频在制作的时候实际上都是独立编码的，得到的是分开的数据，为了传输方便必须要用某种方式合起来，这就有了各种封装格式也就有了**demux**。**demux**分解出来的音频和视频流分别送往音频解码器和视频解码器。因为原始的音视频都是占用大量空间，而且冗余度较高的数据，通常在制作的时候就会进行某种压缩。这就是我们熟知的音视频编码格式，包括MPEG1（VCD）、MPEG2（DVD）、MPEG4、H.264、rmvb等等。音视频解码器的作用就是把这些压缩了的数据还原成原始的音视频数据。VLC解码MPEG2使用了一个独立的库**libmpeg2**，调用它的源文件是 `/modules/codecs/libmpeg2.c`。VLC关于编解码的模块都放在**/modules/codecs**目录下，其中包括著名的庞大的**ffmpeg**。

解码器，例如视频解码器输出的是一张一张的类似位图格式的图像，但是要让人从屏幕看得到，还需要一个视频输出的模块。当然可以像一个Win32窗口程序那样直接把图像画到窗口DC上——VLC的一个输出模块**WinGDI**就是这么干的，但是通常这太慢了，而且消耗大量的CPU。在Windows下比较好的办法是用DirectX的接口，会自动调用显卡的加速功能。

这样的功能分解使得模块化更容易一点，每个模块住需要专注于自己的事；从整体来说功能强大而且灵活。

但是事情总是不会那么简单。就拿**access**来说，媒体的访问是分层的，如RTSP就涉及到IPv4、TCP、UDP、RTCP、RTSP等多个层次的协议。有些视频格式包括了传输、封装格式和编辑码格式如MPEG系列，有些封装格式是独立的容器，但是很多人会误解它是编解码格式，如**mkv**、**avi**这些。

音频和视频在**demux**之后就是独立的，但是需要有一套机制把它们同步起来。同时我们需要有一套机制来控制速度、暂停、停止、跳进，获取各种媒体信息，这些都是很复杂而又很重要的事情。

另外也许需要在某个地方插入一些修改，来实现某种效果。如音频的EQ，视频的亮度调整之类的，VLC专门设计了**access\_filter**、**audio\_filter**和**video\_filter**类型的模块来做这一类事情。

VLC比较独特的地方是集成了原来的VLS的功能，这依赖于VLC中**stream\_output**类型的模块，它们可以把正在播放的视频以某种方式重新转码和发送出去，如**http**、**UDP**、文件等等。

MPlayer的结构与此是类似的，如**/stream**目录对应的是**access**的功能，**/mpdemux**对应的**demux**功能，**/libmpcodecs**是解码器，**/libvo**和**/libao2**分别是视频和音频的输出。

DirectShow也是类似的，不过分类更多一些更复杂一点。DirectShow里面的模块叫做“**filter**”，**filter**之间通过“**pin**”来连接。**access**的模块对应于DirectShow中的**SourceFilter**，这一类**Filter**只有输出**pin**没有输入**pin**。**demux**模块对应于**splitter filter**，这种**filter**有一个输入**pin**，多个输出**pin**。解码模块是一类**transform filter**，有一个输入**pin**、一个输出**pin**，输出模块对应于**rendering filter**，有一个输

入pin，没有输出pin。当然transform filter不一定是解码器，也可能是某种其他的处理。  
另外给出一个VLC的API Document，参见：<http://rogerfd.cn/doc/vlcapi.htm>

## 六、精简

VLC默认包含了大量的功能，超过2百个插件；Windows下安装包大小接近10MB，安装之后超过35MB。有时候如果我们只需要部分的功能，应该如何精简它呢？

精简VLC的第一步是重新配置编译选项，将不需要的功能去掉。一个精简配置脚本如下：

```
PATH=/usr/win32-branch/bin:$PATH\  
PKG_CONFIG_PATH=/usr/win32-branch/lib/pkgconfig\  
CPPFLAGS="-I/usr/win32-branch/include -I/usr/win32-branch/include/ebml"\  
LDFLAGS=-L/usr/win32-branch/lib\  
CC="gcc -mno-cygwin" CXX="g++ -mno-cygwin"\  
./configure \  
-with-freetype-config-path=/usr/win32-branch/bin \  
-with-fribidi-config-path=/usr/win32-branch/bin \  
-disable-fimpeg \  
-disable-live555 \  
-disable-cddax \  
-disable-vcdx \  
-disable-skins2 \  
-disable-wxwidgets \  
-disable-mozilla \  
-disable-httpd \  
-disable-dvdnav \  
-disable-libcdio \  
-disable-libcddb \  
-disable-cdda \  
-disable-sdl \  
-disable-sdl-image \  
-disable-vcd \  
-disable-dvdread \  
-disable-smb \  
-disable-cmml \  
-disable-alsa \  
-disable-openssl \  
-disable-png \  
-disable-screen \  
-disable-mkv \  
-disable-mod \  
-disable-mpc \  
-disable-libtar \  
-disable-speex \  
-disable-visual \  
-disable-daap \  
-disable-bonjour \  
-disable-gnutls \  
-disable-vorbis \  
-disable-real \  
-disable-xml
```



```
-disable-x264 \
-enable-sout \
-enable-activex \
-enable-release
```

从这个脚本可见，主要是禁用了大量的功能。运行 `./configure -h` 可以显示每一个配置项的意义；如果还不清楚可以再去搜一下选项对应的库的位置。`ffmpeg`是一个庞大的编解码库，但是MPEG2的解码可以利用另一个库`libmpeg2`，所以在这里不需要它。

注意make完之后要用`make package-win32-base`来打一下包，打包时会用`strip`工具来压缩每一个可执行文件。

但是这样还没有结束。配置项并不是很全面的，而且还会有一些小问题，例如`xml`禁用了还会有。。。于是还需要手动的清理一下`plugins`目录里面的插件，挑出没有用处的，这也需要对VLC模块的熟悉。或者使用Roger的笨方法，一个一个试，对功能没有影响的就拿掉。

SimpleTV0.4所使用的plugin如下所示：

<code>liba52sys_plugin.dll</code>	<code>liba52</code> 用于AC3格式的音频解码，CCTV高清频道需要
<code>liba52tofloat32_plugin.dll</code>	
<code>liba52tospdif_plugin.dll</code>	
<code>liba52_plugin.dll</code>	
<code>libaccess_file_plugin.dll</code>	文件访问，播放文件需要
<code>libaccess_http_plugin.dll</code>	HTTP访问模块
<code>libaccess_output_file_plugin.dll</code>	sout的文件模块，用于录制
<code>libaccess_udp_plugin.dll</code>	UDP访问模块
<code>libaout_directx_plugin.dll</code>	DirectX的声音输出，默认的声音输出模块
<code>libdeinterlace_plugin.dll</code>	解交错的vout-filter模块
<code>libdirect3d_plugin.dll</code>	Direct3D视频输出模块，在上面可以半透明。。。。
<code>libfixed32tofloat32_plugin.dll</code>	一些数据转换模块
<code>libfixed32tos16_plugin.dll</code>	
<code>libfloat32tos16_plugin.dll</code>	
<code>libfloat32tos8_plugin.dll</code>	
<code>libfloat32tou16_plugin.dll</code>	
<code>libfloat32tou8_plugin.dll</code>	
<code>libfloat32_mixer_plugin.dll</code>	
<code>libglwin32_plugin.dll</code>	OpenGL视频输出模块
<code>libi420_ymga_plugin.dll</code>	
<code>libi422_yuy2_plugin.dll</code>	
<code>libipv4_plugin.dll</code>	IPv4网络模块
<code>libipv6_plugin.dll</code>	IPv6网络模块
<code>liblibmpeg2_plugin.dll</code>	<code>libmpeg2</code> ，解码全靠它
<code>liblogger_plugin.dll</code>	日志模块，调试必备
<code>libmemcpy_plugin.dll</code>	好像是利用各种CPU指令加速内存拷贝的
<code>libmpeg_audio_plugin.dll</code>	MPEG音频的解码模块
<code>libmpgatofixed32_plugin.dll</code>	
<code>libmpga_plugin.dll</code>	
<code>libmux_dummy_plugin.dll</code>	
<code>libpacketizer_mpegvideo_plugin.dll</code>	
<code>libs16tofixed32_plugin.dll</code>	
<code>libs16tofloat32swab_plugin.dll</code>	
<code>libs16tofloat32_plugin.dll</code>	
<code>libs8tofloat32_plugin.dll</code>	

libsimple\_channel\_mixer\_plugin.dll  
 libstream\_out\_display\_plugin.dll     sout的模块 。 。 。  
 libstream\_out\_duplicate\_plugin.dll  
 libstream\_out\_standard\_plugin.dll  
 libtrivial\_channel\_mixer\_plugin.dll  
 libtrivial\_mixer\_plugin.dll  
 libtrivial\_resampler\_plugin.dll  
 libts\_plugin.dll                    TS流的解复用模块  
 libu8tofixed32\_plugin.dll  
 libu8tofloat32\_plugin.dll  
 libugly\_resampler\_plugin.dll  
 libvout\_directx\_plugin.dll         DirectX视频输出模块  
 libwingdi\_plugin.dll               GDI视频输出模块

作为ActiveX控件的VLC，只需要这些模块和axvlc.dll就足够了。当然如果使用了libvlc.dll还要带上。

## VLC 0.9.2 正式发布 && VLC播放Rm Rmvb的插件

新的VLC加入了RealVideo的demux和codec module，已经可以播放rm、rmvb格式的real视频，但是需要下载以下两个附件：

pncrt.dll 拷贝到 C:\Program Files\VideoLAN\VLC\  
 drv43260.dll 拷贝到 C:\Program Files\VideoLAN\VLC\plugins\

这两个文件也可以从MPlayer的win32 codec中拷贝出来。从测试的情况来看，顺序播放没有什么问题，但是跳转的时候比较慢，图像不太稳定

这里有对VLC新特性的介绍 [http://wiki.videolan.org/What\\_is\\_cool\\_in\\_0.9](http://wiki.videolan.org/What_is_cool_in_0.9)

Release Note： <http://www.videolan.org/developers/vlc/NEWS>

下载目录：<http://download.videolan.org/pub/vlc/0.9.2/win32/>

下载链接（exe）：<http://download.videolan.org/pub/vlc/0.9.2/win32/vlc-0.9.2-win32.exe>

## 七、将VLC嵌入自己的应用

总的来说把VLC内嵌入自己的应用有4种途径：

- 直接调用VLC进程
- VLC的plugin for Mozilla
- VLC的ActiveX插件
- 调用libvlc

当然，理论上也可以把VLC的源码直接扣出一部分来放到自己的程序中，不过这种方法难度太大没多大意义。

先说第一种办法，别怀疑，这是非常有用的一种解决方案，特别是对于转码、流转发等应用，通常直接调用VLC就可以了。VLC有HTTP和Telnet的Interface模块（就是和用户交互的界面），也很方便和远程交互。而且跨平台应用也非常方便，只需要安装不同平台的VLC就行了。直接调用VLC的话最重要的就是参数，可以从 `vlc.exe -H` 中获取比较详细的命令行参数介绍。对于转码、转发的，命令行参数会比较复杂，可以用GUI先试一下，然后把界面上的Generated stream output string拷下来。

VLC的Mozilla plugin除了用于Firefox的web页之外，还可以用于任何XUL-base的应用，这种应用通常使用xulrunner来启动。XUL是一种用XML来写界面的语言，有点像HTML而且也兼容大部分HTML标记，Firefox就是用XUL来写的。VLC的Mozilla插件通常用Javascript来控制，详细的接口文档可以参考这里：<http://www.videolan.org/doc/play-howto/en/ch04.html#id310965> 这个接口有比较老的第一版和比较新的第二版，建议用新的，功能多一些。用VLC的Mozilla插件的应用并不少，例如比较出名的Miro、Songbird。因为这个好处是显而易见的：跨平台、可控性强、使用简单；限制也很明显：专为Mozilla设计，应用的范围很窄。

VLC的ActiveX插件只能用于Windows平台，当然这是因为ActiveX只是微软的东西。但是不得不说这个还是很好用的。无论是C++、C#、VB还是网页，都可以轻松地将这个ActiveX利用起来。关于这个的VLC官方文档在这里[http://wiki.videolan.org/ActiveX\\_Controls](http://wiki.videolan.org/ActiveX_Controls) 开发、使用之前别忘了先要注册ActiveX控件，方法是运行 `regsvr32 axvlc.dll`。Roger 之前写的 SimpleTV 就是调用了VLC的ActiveX控件，所以这也可以作为一个示例的代码，可以主要看看怎么调用这个控件来做串流的代码：

```
//Create Option for the plugin
bool bIPv6 = false;
if (strstr (szMRL, "..."))
{
    bIPv6 = true;
}

tagVARIANT opt,var,ip6;
CString sout, ipv6 = "ipv6=1";
SAFEARRAY *psa;
long index;
SAFEARRAYBOUND bound[1] = {0};
bound[0].cElements = bIPv6 ? 2 : 1;

if (m_cfg.bDisplayWhenRecording)
{
    sout.Format("sout=#duplicate {dst=display,dst=std {access=file,mux=ts,dst=\\\"%s\\\"}}",
        strName);
}
else
{
    sout.Format("sout=#duplicate {dst=std {access=file,mux=ts,dst=\\\"%s\\\"}}", strName);
}

var.vt = VT_BSTR;
var.bstrVal = sout.AllocSysString();
psa = SafeArrayCreate (VT_VARIANT, 1, bound);
index = 0;
SafeArrayPutElement (psa, &index, &var);
if (bIPv6)
{
    ip6.vt = VT_BSTR;
    ip6.bstrVal = ipv6.AllocSysString();
    index = 1;
    SafeArrayPutElement (psa, &index, &ip6);
}
opt.vt = VT_ARRAY | VT_BSTR;
opt.parray = psa;

tagVARIANT nul;
nul.vt = VT_NULL;
m_vlc.GetPlaylist ().add (szMRL, nul, opt);
m_vlc.GetPlaylist ().next ();
```

```
m_vlc.GetPlaylist().play();
```

很多人调用VLC的ActiveX控件都会遇到一个问提就是，播放没有反应。大部分的原因都是因为plugins的问题。VLC作为exe运行时递归搜索当前目录下的modules和plugins目录来寻找plugin，但是当以ActiveX控件运行时，默认只会检查注册表的HKLM\_Software\_VideoLan\_VLC\_InstallDir，并添上plugins作为模块搜索路径，如果这个路径没有设置或者设置不对，VLC将找不到大部分的模块，导致大部分功能缺失而无法运行。如果需要修改这一特性可以修改\activex\plugin.cpp（0.8.6i）中的VLCPlugin::getVLC函数。在SimpleTV中，Roger修改的一个片段是

```
char p_pluginpath[MAX_PATH+1];
int iTmp = 0;
/* 得到SimpleTV.exe的全路径 */
DWORD dwLen = GetModuleFileNameA(NULL, p_pluginpath, MAX_PATH);
for (iTmp = strlen(p_pluginpath) - 1; iTmp >= 0; -- iTmp)
{
    if (p_pluginpath[iTmp] == '\\')
    {
        p_pluginpath[iTmp] = 0;
        break;
    }
}
/* 将模块路径设为SimpleTV下的 player\modules */
strcat(p_pluginpath, "\\player\modules");
ppsz_argv[ppsz_argc++] = "-plugin-path";
ppsz_argv[ppsz_argc++] = p_pluginpath;
```

还有一点是配置文件vlcrc的路径，如果需要修改的话，可以修改src/misc/configuration.c中的GetDir函数，在Windows下这个目录默认返回%APPDATA%，以上两步做好的话其实就可以做出一个Portable的绿色版VLC了。

和其他ActiveX的播放器，例如WMP、QuickTime、RealPlayer相比，VLC的优点也很明显：自带所需解码器、开源、功能强大。

当然ActiveX方式和Mozilla插件的方式也是有局限性的，这样的方式功能还不够强大，例如没有截屏的接口...如果需要更多完整的功能，可以考虑直接调用libvlc的接口，

VLC官方提供了一些文档和例子 <http://wiki.videolan.org/Libvlc>。这里还有一个用C#调用libvlc的例子<http://jeremiah.blog.51cto.com/539865/116981>

在Windows下libvlc可能默认是不编译的（0.8.6i），可以在configure中开启这一选项 `-enable-shared-libvlc`

完整的VLC体积还是不小的，如果只需要VLC一部分功能的话，可以最后做一下精简。





[龙将新服,注册送V1](#)

龙将新服开启,注册就送V1,先得先得.



[上海电信云主机5M带..](#)

用户可自行安装软件 管理PTF IIS

分享到： 

[Flex 4中组件背景设置（填充方式）group为 ...](#) | [IReport的子报表设置](#)

- 2010-10-29 10:01
- 浏览 1920
- [评论\(0\)](#)
- 分类:[Web前端](#)
- [相关推荐](#)

评论

发表评论



[您还没有登录,请您登录后再发表评论](#)



xhc66

- 浏览:9950 次
- 性别:
- 来自: 北京
- 我现在离线

最近访客

[更多访客>>](#)



[确实比较男](#)



[likecolin1](#)



[wc593237554](#)



[圣诞王子](#)

文章分类

- [全部博客 \(14\)](#)
- [IReport+Jasperreport \(3\)](#)
- [VLC 、QuickTimeWeb插件相关 \(1\)](#)
- [Flex 4 \(3\)](#)
- [WebService \(1\)](#)
- [JS\(Java Script\) \(4\)](#)

- [Java相关 \(2\)](#)

#### 社区版块

- [我的资讯](#) (0)
- [我的论坛](#) (0)
- [我的问答](#) (0)

#### 存档分类

- [2012-02](#) (1)
- [2011-06](#) (1)
- [2011-04](#) (1)
- [更多存档...](#)

#### 最新评论

- [hanyun19880918](#) :  
[IReport的子报表设置](#)

---

声明：ITeye文章版权属于作者，受法律保护。没有作者书面许可不得转载。若作者同意转载，必须以超链接形式标明文章原始出处和作者。

© 2003-2011 ITeye.com. All rights reserved. [ 京ICP证110151号 京公网安备110105010620 ]