# 各种音视频编解码详解

## [编解码学习笔记（一）：基本概念](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://javasogo.iteye.com%2Fblog%2F1015213)

媒体业务是网络的主要业务之间。尤其移动互联网业务的兴起，在运营商和应用开发商中，媒体业务份量极重，其中媒体的编解码服务涉及需求分析、应用开发、释放license收费等等。最近因为项目的关系，需要理清媒体的codec，比较搞的是，在豆丁网上看运营商的规范标准，同一运营商同样的业务在不同文档中不同的要求，而且有些要求就我看来应当是历史的延续，也就是现在已经很少采用了。所以豆丁上看不出所以然，从 wiki上查。中文的wiki信息量有限，很短，而wiki的英文内容内多，删减版也减肥得太过。我在网上还看到一个山寨的中文wiki，长得很像，红色的，叫“天下维客”。wiki的中文还是很不错的，但是阅读后建议再阅读英文。

　　我对媒体codec做了一些整理和总结，资料来源于wiki，小部分来源于网络博客的收集。网友资料我们将给出来源。如果资料已经转手几趟就没办法，雁过留声，我们只能给出某个轨迹

### ****基本概念****

#### **编解码**

　　编解码器（codec）指的是一个能够对一个信号或者一个数据流进行变换的设备或者程序。这里指的变换既包括将 信号或者数据流进行编码（通常是为了传输、存储或者加密）或者提取得到一个编码流的操作，也包括为了观察或者处理从这个编码流中恢复适合观察或操作的形式的操作。编解码器经常用在视频会议和流媒体等应用中。

#### **容器**

　　很多多媒体数据流需要同时包含音频数据和视频数据，这时通常会加入一些用于音频和视频数据同步的元数据，例如字幕。这三种数据流可能会被不同的程序，进程或者硬件处理，但是当它们传输或者存储的时候，这三种数据通常是被封装在一起的。通常这种封装是通过视频文件格 式来实现的，例如常见的\*.mpg, \*.avi, \*.mov, \*.mp4, \*.rm, \*.ogg or \*.tta. 这些格式中有些只能使用某些编解码器，而更多可以以容器的方式使用各种编解码器。

　　FourCC全称Four-Character Codes，是由4个字符（4 bytes）组成，是一种独立标示视频数据流格式的四字节，在wav、avi档案之中会有一段FourCC来描述这个AVI档案，是利用何种codec来 编码的。因此wav、avi大量存在等于“IDP3”的FourCC。

　　视频是现在电脑中多媒体系统中的重要一环。为了适应储存视频的需要，人们设定了不同的视频文件格式来把视频和音频放在一个文件中，以方便同时回放。视频档实际上都是一个容器里面包裹着不同的轨道，使用的容器的格式关系到视频档的可扩展性。

### ****参数介绍****

#### **采样率**

　　采样率（也称为采样速度或者采样频率）定义了每秒从连续信号中提取并组成离散信号的采样个数，它用赫兹（Hz）来表示。采样频率的倒数叫作采样周期或采样时间，它是采样之间的时间间隔。注意不要将采样率与比特率（bit rate，亦称“位速率”）相混淆。

　　采样定理表明采样频率必须大于被采样信号带宽的两倍，另外一种等同的说法是奈奎斯特频率必须大于被采样信号的带宽。如果信号的带宽是 100Hz，那么为了避免混叠现象采样频率必须大于200Hz。换句话说就是采样频率必须至少是信号中最大频率分量频率的两倍，否则就不能从信号采样中恢复原始信号。

　　对于语音采样：

* 8,000 Hz - 电话所用采样率, 对于人的说话已经足够
* 11,025 Hz
* 22,050 Hz - 无线电广播所用采样率
* 32,000 Hz - miniDV 数码视频 camcorder、DAT (LP mode)所用采样率
* 44,100 Hz - 音频 CD, 也常用于 MPEG-1 音频（VCD, SVCD, MP3）所用采样率
* 47,250 Hz - Nippon Columbia (Denon)开发的世界上第一个商用 PCM 录音机所用采样率
* 48,000 Hz - miniDV、数字电视、DVD、DAT、电影和专业音频所用的数字声音所用采样率
* 50,000 Hz - 二十世纪七十年代后期出现的 3M 和 Soundstream 开发的第一款商用数字录音机所用采样率
* 50,400 Hz - 三菱 X-80 数字录音机所用所用采样率
* 96,000 或者 192,000 Hz - DVD-Audio、一些 LPCM DVD 音轨、Blu-ray Disc（蓝光盘）音轨、和 HD-DVD （高清晰度 DVD）音轨所用所用采样率
* 2.8224 MHz - SACD、 索尼 和 飞利浦 联合开发的称为 Direct Stream Digital 的 1 位 sigma-delta modulation 过程所用采样率。

　　在模拟视频中，采样率定义为帧频和场频，而不是概念上的像素时钟。图像采样频率是传感器积分周期的循环速度。由于积分周期远远小于重复所需时间，采样频率可能与采样时间的倒数不同。

* 50 Hz - PAL 视频
* 60 / 1.001 Hz - NTSC 视频

　　当模拟视频转换为数字视频的时候，出现另外一种不同的采样过程，这次是使用像素频率。一些常见的像素采样率有：

* 13.5 MHz - CCIR 601、D1 video

#### **分辨率**

　　分辨率，泛指量测或显示系统对细节的分辨能力。此概念可以用时间、空间等领域的量测。日常用语中之分辨率多用于图像的清晰度。分辨率越高代表图像品质越好，越能表现出更多的细节。但相对的，因为纪录的信息越多，文件也就会越大。目前个人电脑里的图像，可以使用图像 处理软件，调整图像的大小、编修照片等。例如 photoshop，或是photoimpact等软件。

**图像分辨率** ：

　　用以描述图像细节分辨能力，同样适用于数字图像、胶卷图像、及其他类型图像。常用'线每毫米'、 '线每英吋'等来衡量。通常，“分辨率”被表示成每一个方向上的像素数量，比如640x480等。而在某些情况下，它也可以同时表示成“每英吋像素” （pixels per inch，ppi）以及图形的长度和宽度。比如72ppi，和8x6英吋。

**视频分辨率** ：

各种电视规格分辨率比较视频的画面大小称为“分辨率”。数位视频以像素为度量单位，而类比视频以水平扫瞄线数量为度量单位。标清电视频号分辨率为 720/704/640x480i60（NTSC）或768/720x576i50（PAL/SECAM）。新的高清电视（HDTV）分辨率可达 1920x1080p60，即每条水平扫瞄线有1920个像素，每个画面有1080条扫瞄线，以每秒钟60张画面的速度播放。

#### **画面更新率fps**

　　Frame rate中文常译为“画面更新率”或“帧率”，是指视频格式每秒钟播放的静态画面数量。典型的画面更新率由早期的每秒6或8张（frame persecond，简称fps），至现今的每秒120张不等。PAL (欧洲，亚洲，澳洲等地的电视广播格式) 与 SECAM (法国，俄国，部分非洲等地的电视广播格式) 规定其更新率为25fps，而NTSC (美国，加拿大，日本等地的电视广播格式) 则规定其更新率为29.97 fps。电影胶卷则是以稍慢的24fps在拍摄，这使得各国电视广播在播映电影时需要一些复杂的转换手续（参考Telecine转换）。要达成最基本的视觉暂留效果大约需要10fps的速度。

### **压缩方法**

#### **有损压缩和无损压缩**

　　在视频压缩中有损（Lossy ）和无损（Lossless）的概念与静态图像中基本类似。无损压缩也即压缩前和解压缩后的数据完全一致。多数的无损压缩都采用RLE行程编码算法。有损压缩意味着解压缩后的数据与压缩前的数据不一致。在压缩的过程中要丢失一些人眼和人耳所不敏感的图像或音频信息，而且丢失的信息不可恢复。几乎所有高压缩的算法都采用有损压缩，这样才能达到低数据率的目标。丢失的数据率与压缩比有关，压缩比越小，丢失的数据越多，解压缩后的效果一般越差。此外，某些有损压缩算法采用多次重复压缩的方式，这样还会引起额外的数据丢失。

* 无损格式，例如WAV，PCM，TTA，FLAC，AU，APE，TAK，WavPack(WV)
* 有损格式，例如MP3，Windows Media Audio（WMA），Ogg Vorbis（OGG），AAC

#### **帧内压缩和帧间压缩**

　　帧内（Intraframe）压缩也称为空间压缩 （Spatial compression）。当压缩一帧图像时，仅考虑本帧的数据而不考虑相邻帧之间的冗余信息，这实际上与静态图像压缩类似。帧内一般采用有损压缩算法，由于帧内压缩时各个帧之间没有相互关系，所以压缩后的视频数据仍可以以帧为单位进行编辑。帧内压缩一般达不到很高的压缩。

　　采用帧间（Interframe）压缩是基于许多视频或 动画的连续前后两帧具有很大的相关性，或者说前后两帧信息变化很小的特点。也即连续的视频其相邻帧之间具有冗余信息，根据这一特性，压缩相邻帧之间的冗余量就可以进一步提高压缩量，减小压缩比。帧间压缩也称为时间压缩（Temporalcompression），它通过比较时间轴上不同帧之间的数据进行压缩。帧间压缩一般是无损的。帧差值（Frame differencing）算法是一种典型的时间压缩法，它通过比较本帧与相邻帧之间的差异，仅记录本帧与其相邻帧的差值，这样可以大大减少数据量。

#### **对称编码和不对称编码**

　　对称性（symmetric）是压缩编码的一个关键特征。对称意味着压缩和解压缩占用相同的计算处理能力和时间，对称算法适合于实时压缩和传送视频，如视频会议应用就以采用对称的压缩编码算法为好。而在电子出版和其它多媒体应用中，一般是把视频预先压缩处理好，尔后再播放，因此可以采用不对称（asymmetric）编码。不对称或非对称意味着压缩时需要花费大量的处理能力和时间，而解压缩时则能较好地实时回放，也即以不同的速度进行压缩和解压缩。一般地说，压缩一段视频的时间比回放（解压缩）该视频的时间 要多得多。例如，压缩一段三分钟的视频片断可能需要10多分钟的时间，而该片断实时回放时间只有三分钟。

除wiki外的资料来源：http://tech.lmtw.com/csyy/Using/200411/3142.html

#### **资料压缩**

港台将information翻译为资料 资料压缩是透过去除资料中的冗余资讯而达成。就视讯资料而言，资料中的冗余资讯可以分成四类：

时间上的冗余资讯（temporal redundancy）  
　　在视讯资料中，相邻的帧（frame）与帧之间通常有很强的关连性，这样的关连性即为时间上的冗余资讯。这即是上一节中的帧间压缩。

空间上的冗余资讯（spatial redundancy）  
　　在同一张帧之中，相邻的像素之 间通常有很强的关连性，这样的关连性即为空间上的冗余资讯。这即是上一节中的帧内压缩。

统计上的冗余资讯（statistical redundancy）  
　　统计上的冗余资讯指的是欲编码的符号（symbol）的机率分布是不均匀（non-uniform）的。

感知上的冗余资讯（perceptual redundancy）  
　　感知上的冗余资讯是指在人在观看视讯时，人眼无法察觉的资讯。

#### **视讯压缩**

视讯压缩（英文：Video compression） 是指运用资料压缩技术将数位视讯资料中的冗余资讯去除，降低表示原始视讯所需的资料量，以便视讯资料的传输与储存。实际上，原始视讯资料的资料量往往过大，例如未经压缩的电视品质视讯资料的位元率高达216Mbps，绝大多数的应用无法处理如此庞大的资料量，因此视讯压缩是必要的。目前最新的视讯编码标准为ITU-T视讯编码专家组（VCEG）和ISO／IEC动态图像专家组（MPEG）联合组成的联合视讯组（JVT，Joint Video Team）所提出的H.264/AVC。

## 编解码学习笔记（二）：codec类型

　　一个典型的视讯编码器：在进行当前信号编码时，编码器首先会产生对当前信号做预测的信号，称作预测信号（predicted signal），预测的方式可以是时间上的预测（interprediction），亦即使用先前帧的信号做预测，或是空间上的预测 （intra prediction），亦即使用同一张帧之中相邻像素的信号做预测。得到预测信号后，编码器会将当前信号与预测信号相减得到残余信号（residual signal），并只对残余信号进行编码，如此一来，可以去除一部份时间上或是空间上的冗余资讯。接着，编码器并不会直接对残余信号进行编码，而是先将残余信号经过变换（通常为离散余弦变换）然后量化以 进一步去除空间上和感知上的冗余资讯。量化后得到的量化系数会再透过熵编码，去除统计上的冗余资讯。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **视讯编码标准发展** | | | | |
| **年份** | **标准** | **制定组织** | **解除版权保护 （DRM-free ）** | **主要应用** |
| **1984** | [H.120](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DH.120%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) | [ITU-T](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FITU-T) | 是 |  |
| **1990** | [H.261](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FH.261) | [ITU-T](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FITU-T) | 是 | [视讯会议](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2F%25E8%25A6%2596%25E8%25A8%258A%25E6%259C%2583%25E8%25AD%25B0)、[视讯通话](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3D%25E8%25A6%2596%25E8%25A8%258A%25E9%2580%259A%25E8%25A9%25B1%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) |
| **1993** | [MPEG-1 第二部份](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DMPEG-1_%25E7%25AC%25AC%25E4%25BA%258C%25E9%2583%25A8%25E4%25BB%25BD%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) | [ISO](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FISO) ／[IEC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FIEC) | 是 | 影音光碟（[VCD](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FVCD) ） |
| **1995** | [H.262/MPEG-2 第二部份](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DH.262%2FMPEG-2_%25E7%25AC%25AC%25E4%25BA%258C%25E9%2583%25A8%25E4%25BB%25BD%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) | [ISO](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FISO)／ [IEC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FIEC)、[ITU-T](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FITU-T) | 否 | [DVD影碟](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DDVD%25E5%25BD%25B1%25E7%25A2%259F%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1)（[DVD-Video](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FDVD-Video) ）、[蓝光](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2F%25E8%2597%258D%25E5%2585%2589)（[Blu-Ray](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DBlu-Ray%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) ）影碟、数位视讯广播（[DVB](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FDVB) ）、[SVCD](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FSVCD) |
| **1996** | [H.263](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FH.263) [[6]](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2F%25E8%25A7%2586%25E9%25A2%2591%25E5%258E%258B%25E7%25BC%25A9%23_note-H263-5) | [ITU-T](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FITU-T) |  | [视讯会议](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2F%25E8%25A6%2596%25E8%25A8%258A%25E6%259C%2583%25E8%25AD%25B0)、[视讯通话](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3D%25E8%25A6%2596%25E8%25A8%258A%25E9%2580%259A%25E8%25A9%25B1%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1)、[3G](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2F3G) 手机视讯（[3GP](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2F3GP) ） |
| **1999** | [MPEG-4 第二部份](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DMPEG-4_%25E7%25AC%25AC%25E4%25BA%258C%25E9%2583%25A8%25E4%25BB%25BD%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) | [ISO](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FISO) ／[IEC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FIEC) | 否 |  |
| **2003** | [H.264/MPEG-4 AVC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FH.264%2FMPEG-4_AVC) [[1]](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2F%25E8%25A7%2586%25E9%25A2%2591%25E5%258E%258B%25E7%25BC%25A9%23_note-H264_AVC-0) | [ISO](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FISO)／ [IEC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FIEC)、[ITU-T](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FITU-T) | 否 | [蓝光](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2F%25E8%2597%258D%25E5%2585%2589)（[Blu-Ray](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DBlu-Ray%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) ）影碟、数位视讯广播（[DVB](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FDVB)）、[iPod](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FIPod) 视讯、[高画质DVD](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3D%25E9%25AB%2598%25E7%2595%25AB%25E8%25B3%25AADVD%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1)（[HD DVD](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FHD_DVD) ） |

常见的编解码见下表，在以后会分类论述：

|  |  |  |
| --- | --- | --- |
| 视频codec | [ISO](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2F%25E5%259C%258B%25E9%259A%259B%25E6%25A8%2599%25E6%25BA%2596%25E5%258C%2596%25E7%25B5%2584%25E7%25B9%2594)/[IEC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2F%25E5%259B%25BD%25E9%2599%2585%25E7%2594%25B5%25E5%25B7%25A5%25E5%25A7%2594%25E5%2591%2598%25E4%25BC%259A) | [MJPEG](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FMotion_JPEG) **·** [Motion JPEG 2000](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FJPEG_2000%23Motion_JPEG_2000) **·** [MPEG-1](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FMPEG-1) **·** [MPEG-2](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FMPEG-2) ([Part 2](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DH.262%2FMPEG-2_Part_2%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) )**·** [MPEG-4](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FMPEG-4) ([Part 2/ASP](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DMPEG-4_Part_2%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [Part 10/AVC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FH.264%2FMPEG-4_AVC) )**·** [HVC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DHigh-performance_Video_Coding%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) |
| [ITU-T](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2F%25E5%259B%25BD%25E9%2599%2585%25E7%2594%25B5%25E4%25BF%25A1%25E8%2581%2594%25E7%259B%259F%25E8%25BF%259C%25E7%25A8%258B%25E9%2580%259A%25E4%25BF%25A1%25E6%25A0%2587%25E5%2587%2586%25E5%258C%2596%25E7%25BB%2584) | [H.120](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DH.120%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [H.261](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FH.261) **·** [H.262](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DH.262%2FMPEG-2_Part_2%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [H.263](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FH.263) **·** [H.264](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FH.264%2FMPEG-4_AVC) **·** [H.265](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FH.265) |
| 其它 | [AMV](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAMV_video_format) **·** [AVS](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAudio_Video_Standard) **·** [Bink](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FBink_Video) **·** [CineForm](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FCineForm) **·** [Cinepak](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FCinepak) **·** [Dirac](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FDirac_%2528codec%2529) **·** [DV](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FDV) **·** [Indeo](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FIndeo) **·** [Microsoft Video 1](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMicrosoft_Video_1)**·**[OMS Video](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FOMS_Video) **·** [Pixlet](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FPixlet) **·** [RealVideo](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FRealVideo) **·** [RTVideo](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FRTVideo) **·** [SheerVideo](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FSheerVideo) **·** [Smacker](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FSmacker_video) **·** [Sorenson Video & Sorenson Spark](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FSorenson_codec)**·** [Theora](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FTheora)**·** [VC-1](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FVC-1)**·** [VP3](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FVP3)**·** [VP6](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FVP6)**·** [VP7](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FVP7)**·** [VP8](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FVP8)**·** [WMV](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FWindows_Media_Video) |
| 音频 codec | [ISO](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2F%25E5%259C%258B%25E9%259A%259B%25E6%25A8%2599%25E6%25BA%2596%25E5%258C%2596%25E7%25B5%2584%25E7%25B9%2594)/[IEC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2F%25E5%259B%25BD%25E9%2599%2585%25E7%2594%25B5%25E5%25B7%25A5%25E5%25A7%2594%25E5%2591%2598%25E4%25BC%259A)[MPEG](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FMPEG) | [MPEG-1 Layer III (MP3)](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FMP3) **·** [MPEG-1 Layer II](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FMPEG-1_Audio_Layer_II) **·** [MPEG-1 Layer I](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DMPEG-1_Audio_Layer_I%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [AAC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FAdvanced_Audio_Coding) **·** [HE-AAC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DHigh-Efficiency_Advanced_Audio_Coding%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·**[MPEG-4 ALS](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DAudio_Lossless_Coding%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [MPEG-4 SLS](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DMPEG-4_SLS%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [MPEG-4 DST](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FDirect_Stream_Transfer%23DST) |
| [ITU-T](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2F%25E5%259B%25BD%25E9%2599%2585%25E7%2594%25B5%25E4%25BF%25A1%25E8%2581%2594%25E7%259B%259F%25E8%25BF%259C%25E7%25A8%258B%25E9%2580%259A%25E4%25BF%25A1%25E6%25A0%2587%25E5%2587%2586%25E5%258C%2596%25E7%25BB%2584) | [G.711](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FG.711) **·** [G.718](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DG.718%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [G.719](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DG.719%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [G.722](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FG.722) **·** [G.722.1](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DG.722.1%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [G.722.2](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DG.722.2%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [G.723](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FG.723) **·** [G.723.1](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DG.723.1%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [G.726](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FG.726) **·**[G.728](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DG.728%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [G.729](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FG.729) **·** [G.729.1](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DG.729.1%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) |
| 其它 | [AC-3](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FDolby_Digital) **·** [AMR](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAdaptive_Multi-Rate_audio_codec) **·** [AMR-WB](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAdaptive_Multi-Rate_Wideband) **·** [AMR-WB+](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FExtended_Adaptive_Multi-Rate_-_Wideband) **·** [Apple Lossless](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FApple_Lossless) **·** [ATRAC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAdaptive_Transform_Acoustic_Coding) **·** [DRA](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FDynamic_Resolution_Adaptation) **·** [DTS](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FDTS_%2528sound_system%2529) **·**[FLAC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FFree_Lossless_Audio_Codec) **·** [GSM-HR](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FHalf_Rate) **·** [GSM-FR](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FFull_Rate) **·** [GSM-EFR](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FEnhanced_Full_Rate) **·** [iLBC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FInternet_Low_Bit_Rate_Codec) **·** [Monkey's Audio](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMonkey%2527s_Audio) **·** [TTA](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FTTA_%2528codec%2529) (True Audio)**·** [MT9](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMT9) **·** [μ-law](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2F%25CE%259C-law_algorithm) **·**[Musepack](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMusepack) **·** [Nellymoser](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FNellymoser_Asao_Codec) **·** [OptimFROG](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FOptimFROG) **·** [OSQ](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FOriginal_Sound_Quality) **·** [RealAudio](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FRealAudio)**·** [RTAudio](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FRTAudio) **·** [SD2](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FDigidesign%23Sound_Designer_File_Formats) **·** [SHN](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FShorten) **·**[SILK](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FSILK) **·** [Siren](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FSiren_Codec) **·** [Speex](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FSpeex) **·** [TwinVQ](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FTwinVQ) **·** [Vorbis](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FVorbis) **·** [WavPack](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FWavPack) **·**[WMA](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FWindows_Media_Audio) |
| 图像压缩 | [ISO](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2F%25E5%259C%258B%25E9%259A%259B%25E6%25A8%2599%25E6%25BA%2596%25E5%258C%2596%25E7%25B5%2584%25E7%25B9%2594)/[IEC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2F%25E5%259B%25BD%25E9%2599%2585%25E7%2594%25B5%25E5%25B7%25A5%25E5%25A7%2594%25E5%2591%2598%25E4%25BC%259A)/[ITU-T](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2F%25E5%259B%25BD%25E9%2599%2585%25E7%2594%25B5%25E4%25BF%25A1%25E8%2581%2594%25E7%259B%259F%25E8%25BF%259C%25E7%25A8%258B%25E9%2580%259A%25E4%25BF%25A1%25E6%25A0%2587%25E5%2587%2586%25E5%258C%2596%25E7%25BB%2584) | [JPEG](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FJPEG) **·** [JPEG 2000](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FJPEG_2000) **·** [JPEG XR](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DJPEG_XR%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [lossless JPEG](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DLossless_JPEG%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [JBIG](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FJBIG) **·** [JBIG2](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DJBIG2%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [PNG](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FPortable_Network_Graphics) **·** [WBMP](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DWireless_Application_Protocol_Bitmap_Format%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) |
| Others | [APNG](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FAPNG) **·** [BMP](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FBMP) **·** [DjVu](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FDjVu) **·** [EXR](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DOpenEXR%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [GIF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FGraphics_Interchange_Format) **·** [ICER](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DICER%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [ILBM](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DILBM%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [MNG](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FMultiple-image_Network_Graphics) **·** [PCX](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FPCX) **·** [PGF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DProgressive_Graphics_File%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [TGA](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DTruevision_TGA%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·**[TIFF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FTagged_Image_File_Format) |
| 媒体容器 | 通用 | [3GP](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2F3GP) **·** [ASF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DAdvanced_Systems_Format%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [AVI](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FAudio_Video_Interleave) **·** [Bink](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DBink_Video%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [BXF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DBroadcast_eXchange_Format%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [DMF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FDivX%23DivX.E6.A0.BC.E5.BC.8F) **·** [DPX](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DDigital_Picture_Exchange%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [EVO](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DEnhanced_VOB%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [FLV](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FFlash_Video) **·** [GXF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DGeneral_Exchange_Format%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [M2TS](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FM2TS) **·**[Matroska](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FMatroska) **·** [MPEG-PS](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DMPEG_program_stream%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [MPEG-TS](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DMPEG_transport_stream%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [MP4](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FMPEG-4_Part_14) **·** [MXF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DMXF%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [Ogg](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FOgg) **·** [QuickTime](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FQuickTime) **·**[RealMedia](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FRealMedia)**·** [RIFF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FResource_Interchange_File_Format) **·** [Smacker](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DSmacker_video%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) **·** [VOB](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DVOB%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1) |
| 只用于音频 | [AIFF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FAIFF)**·**[AU](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fw%2Findex.php%3Ftitle%3DAu%25E6%2596%2587%25E4%25BB%25B6%25E6%25A0%25BC%25E5%25BC%258F%26amp%3Baction%3Dedit%26amp%3Bredlink%3D1)**·**[WAV](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FWAV) |

上面的表格，查看某个具体的codec，可以在中文的wiki中查找，但是英文的wiki咨询更为丰富，见下表

|  |  |
| --- | --- |
| [**Multimedia**](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMultimedia)[**compression**](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FData_compression)**formats** | |
| [Video compression](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FVideo_compression) | |  |  | | --- | --- | | [ISO](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FInternational_Organization_for_Standardization)/[IEC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FInternational_Electrotechnical_Commission) | MJPEG·Motion JPEG 2000·MPEG-1·[MPEG-2](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMPEG-2)([Part 2](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FH.262%2FMPEG-2_Part_2))·[MPEG-4](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMPEG-4)([Part 2/ASP](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMPEG-4_Part_2) [Part 10/AVC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FH.264%2FMPEG-4_AVC))·[HEVC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FHigh_Efficiency_Video_Coding) | | [ITU-T](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FITU-T) | [H.120](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FH.120)·[H.261](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FH.261)·[H.262](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FH.262%2FMPEG-2_Part_2)·[H.263](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FH.263)·[H.264](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FH.264%2FMPEG-4_AVC)·[HEVC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FHigh_Efficiency_Video_Coding) | | Others | [AMV](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAMV_video_format)·[AVS](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAudio_Video_Standard)·[Bink](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FBink_Video)·[CineForm](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FCineForm)·[Cinepak](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FCinepak)·[Dirac](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FDirac_%2528codec%2529)·[DV](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FDV)·[Indeo](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FIndeo)·[Microsoft Video 1](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMicrosoft_Video_1)·[OMS Video](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FOMS_Video)·[Pixlet](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FPixlet)·[RealVideo](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FRealVideo)·[RTVideo](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FRTVideo)·[SheerVideo](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FSheerVideo)·[Smacker](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FSmacker_video)·[Sorenson Video & Sorenson Spark](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FSorenson_codec)·[Theora](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FTheora)·[VC-1](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FVC-1)·[VP3](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FVP3)·[VP6](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FVP6)·[VP7](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FVP7)·[VP8](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FVP8)·[WMV](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FWindows_Media_Video) | |
| [Audio compression](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAudio_compression_%2528data%2529) | |  |  | | --- | --- | | [ISO](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FInternational_Organization_for_Standardization)/[IEC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FInternational_Electrotechnical_Commission) | [MPEG-1 Layer III (MP3)](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMP3)·[MPEG-1 Layer II](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMPEG-1_Audio_Layer_II)·[MPEG-1 Layer I](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMPEG-1_Audio_Layer_I)·[AAC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAdvanced_Audio_Coding)·[HE-AAC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FHigh-Efficiency_Advanced_Audio_Coding)·[MPEG-4 ALS](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAudio_Lossless_Coding)·[MPEG-4 SLS](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMPEG-4_SLS)·[MPEG-4 DST](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FDirect_Stream_Transfer%23DST)·[MPEG-4 HVXC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FHarmonic_Vector_Excitation_Coding)·[MPEG-4 CELP](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMPEG-4_Part_3%23Subparts) | | [ITU-T](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FITU-T) | [G.711](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FG.711) · [G.718](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FG.718) · [G.719](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FG.719) · [G.722](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FG.722) · [G.722.1](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FG.722.1) · [G.722.2](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FG.722.2) · [G.723](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FG.723) · [G.723.1](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FG.723.1) ·[G.726](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FG.726) · [G.728](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FG.728) · [G.729](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FG.729) · [G.729.1](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FG.729.1) | | Others | [AC-3](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FDolby_Digital)·[AMR](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAdaptive_Multi-Rate_audio_codec)·[AMR-WB](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAdaptive_Multi-Rate_Wideband)·[AMR-WB+](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FExtended_Adaptive_Multi-Rate_-_Wideband)·[Apple Lossless](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FApple_Lossless)·[ATRAC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAdaptive_Transform_Acoustic_Coding)·[DRA](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FDynamic_Resolution_Adaptation)·[DTS](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FDTS_%2528sound_system%2529)·[FLAC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FFree_Lossless_Audio_Codec)·[GSM-HR](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FHalf_Rate)·[GSM-FR](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FFull_Rate)·[GSM-EFR](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FEnhanced_Full_Rate)·[iLBC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FInternet_Low_Bit_Rate_Codec)·[Monkey's Audio](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMonkey%2527s_Audio)·[TTA](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FTTA_%2528codec%2529)(True Audio)·[MT9](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMT9)·[μ-law](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2F%25CE%259C-law_algorithm)·[Musepack](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMusepack)·[Nellymoser](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FNellymoser_Asao_Codec)·[OptimFROG](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FOptimFROG)·[OSQ](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FOriginal_Sound_Quality)·[RealAudio](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FRealAudio)· [RTAudio](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FRTAudio)·[SD2](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FDigidesign%23Sound_Designer_File_Formats)·[SHN](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FShorten)·[SILK](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FSILK)·[Siren](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FSiren_Codec)·[Speex](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FSpeex)·[TwinVQ](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FTwinVQ)·[Vorbis](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FVorbis)·[WavPack](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FWavPack)·[WMA](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FWindows_Media_Audio) | |
| [Image compression](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FImage_compression) | |  |  | | --- | --- | | [ISO](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FInternational_Organization_for_Standardization)/[IEC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FInternational_Electrotechnical_Commission)/[ITU-T](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FITU-T) | [JPEG](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FJPEG)·[JPEG 2000](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FJPEG_2000)·[JPEG XR](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FJPEG_XR)·[lossless JPEG](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FLossless_JPEG)·[JBIG](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FJBIG)·[JBIG2](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FJBIG2)·[PNG](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FPortable_Network_Graphics)·[WBMP](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FWireless_Application_Protocol_Bitmap_Format) | | Others | [APNG](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAPNG)·[BMP](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FBMP_file_format)·[DjVu](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FDjVu)·[EXR](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FOpenEXR)·[GIF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FGraphics_Interchange_Format)·[ICER](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FICER)·[ILBM](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FILBM)·[MNG](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMultiple-image_Network_Graphics)·[PCX](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FPCX)·[PGF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FProgressive_Graphics_File)·[TGA](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FTruevision_TGA)·[QTVR](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FQuickTime_VR)·[TIFF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FTagged_Image_File_Format) | |
| [Media containers](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FContainer_format_%2528digital%2529) | |  |  | | --- | --- | | ISO/IEC | [MPEG-PS](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMPEG_program_stream)·[MPEG-TS](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMPEG_transport_stream)·[MPEG-4 Part 12](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMPEG-4_Part_12)/[JPEG 2000 Part 12](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FJPEG_2000%23Motion_JPEG_2000)·[MPEG-4 Part 14](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMPEG-4_Part_14) | | ITU-T | [H.222.0](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMPEG-2%23Systems) | | Others | [3GP and 3G2](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2F3GP_and_3G2)·[ASF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAdvanced_Systems_Format)·[AVI](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAudio_Video_Interleave)·[Bink](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FBink_Video)·[DivX Media Format](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FDivX_Media_Format)·[DPX](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FDigital_Picture_Exchange)·[EVO](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FEnhanced_VOB)·[Flash Video](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FFlash_Video)·[GXF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FGeneral_Exchange_Format)·[M2TS](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2F.m2ts)·[Matroska](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMatroska)·[MXF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMaterial_Exchange_Format)·[Ogg](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FOgg)·[QuickTime](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FQuickTime)·[RealMedia](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FRealMedia)·[REDCODE RAW](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FREDCODE)·[RIFF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FResource_Interchange_File_Format)·[Smacker](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FSmacker_video)·[MOD and TOD](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMOD_and_TOD_%2528video_format%2529)·[VOB](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FVOB)·[WebM](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FWebM) | | Audio only | [AIFF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAudio_Interchange_File_Format)·[AU](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAu_file_format)·[WAV](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FWAV) | |

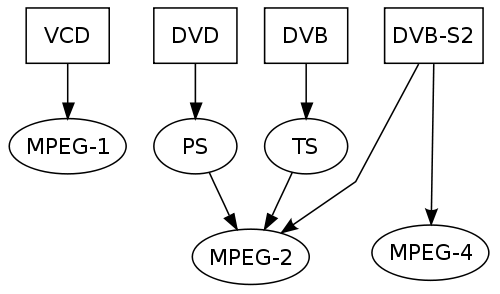
## [编解码学习笔记（三）：Mpeg系列——Mpeg 1和Mpeg 2](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://javasogo.iteye.com%2Fblog%2F1015211)

### **MPEG视频编码标准**

MPEG是Moving Picture Experts Group的简称。这个名字本来的含义是指一个研究视频和音频编码标准的小组。现在我们所说的MPEG泛指又该小组制定的一系列视频编码标准。该小组于 1988年组成，至今已经制定了MPEG-1、MPEG-2、MPEG-3、MPEG-4、MPEG-7等多个标准，MPEG-21正在制定中。

MPEG到目前为止已经制定并正在制定以下和视频相关的标准：

* MPEG-1： 第一个官方的视訊音訊压缩标准，随后在Video CD中被采用，其中的音訊压缩的第三级（MPEG-1 Layer 3）简称MP3， 成为比较流行的音訊压缩格式。
* MPEG-2： 广播质量的视訊、音訊和传输协议。被用于無線數位電視-ATSC、DVB以及ISDB、数字卫星电视（例如DirecTV）、 数字有线电视信号，以及DVD视频光盘技术中。
* MPEG-3： 原本目标是为高解析度电视（HDTV）设计，随后發現MPEG-2已足夠HDTV應用，故 MPEG-3的研發便中止。
* MPEG- 4：2003 年发布的视訊压缩标准，主要是扩展MPEG-1、MPEG-2等標準以支援視訊／音訊物件（video/audio "objects"）的編碼、3D內容、低位元率編碼（low bitrate encoding）和數位版權管理（Digital Rights Management），其中第10部分由ISO/IEC和ITU-T联合发布，称为H.264/MPEG-4 Part 10。参见H.264。
* MPEG-7：MPEG-7并不是一个视訊压缩标准，它是一个多媒体内容的描述标准。
* MPEG-21：MPEG-21是一个正在制定中的标准，它的目标是为未来多媒体的应用提供一个完整的平台。



媒体codec在于MPEG-1，MPEG-2，MPEG-4，如上图所示。

图中名称解释：在图中VCD，DVD地球人都知道，何为DVB？

DVB：数字视频广播（DVB, Digital VideoBroadcasting），是由“DVB Project”维护的一系列为国际所承认的数字电视公开标准。DVB系统传输方式有如下几种：

* 卫星电视（DVB-S 及 DVB-S2）
* 有线电视（DVB-C）
* 无线电视（DVB-T）
* 手持地面无线（DVB-H）

这些标准定义了传输系统的物理层与数据链路层。设备通过同步并行接口（synchronous parallel interface, SPI），同步串行接口（synchronous serial interface, SSI），或异步串行接口（asynchronousserial interface, ASI）与物理层交互。数据以MPEG-2传输流的方式传输，并要求符合更严格的限制（DVB-MPEG）。对移动终端即时压缩传输数据的标准（DVB- H）目前正处于测试之中。

这些传输方式的主要区别在于使用的调制方式，因为不同它们应用的频率带宽的要求不同。利用高频载波的DVB-S使用QPSK调制方式，利用低频载波的DVB-C使用QAM-64 调制方式，而利用VHF 及 UHF载波的DVB-T使用COFDM调制方式。

除音频与视频传输外，DVB也定义了带回传信道（DVB-RC）的数据通信标准（DVB-DATA）。

DVB的codec，视频为：MPEG-2，MPEG-4 AVC；音频为：MP3，AC-3，AAC，HE-AAC。

### **MPEG-1**

　　MPEG-1作为ISO/IEC11172正式发布。

　　MPEG-1较早的视频编码，质量比较差，主要用于 CD-ROM 存储视频，国内最为大家熟悉的就是 VCD（Video CD），他的视频编码就是采用 MPEG-1。它是为CD光盘介质定制的视频和音频压缩格式。一张70分钟的CD光盘传输速率大约在1.4Mbps。而 MPEG-1采用了块方式的运动补偿、离散余弦变换（DCT）、量化等技术，并为1.2Mbps传输速率进行了优化。MPEG-1 随后被Video CD采用作为内核技术。MPEG-1的输出质量大约和传统录像机VCR，信号质量相当，这也许是Video CD在发达国家未获成功的原因。

　　MPEG-1音频分三层，就是MPEG-1 Layer I, II, III，其中第三层协议也就是MPEG- 1 Layer 3，简称MP3。MP3目前已经成为广泛流传的音频压缩技术。

#### **MPEG-1有下面几个部分：**

* 第一部分（Part 1）：系统；
* 第二部分（Part 2）：视频；
* 第三部分（Part 3）：音频；定义level1，level2，level3，并在MPEG-2中定义了扩展。
* 第四部分（Part 4）：一次性测试；
* 第五部分（Part 5）：参考软件；

#### **MPEG-1的缺点：**

* 1个音频压缩系统限于两个通道（立体声）
* 没有为隔行扫描视频提供标准化支持，且压缩率差
* 只有一个标准化的“profile” （约束参数比特流）， 不适应 更高分辨率的视频。MPEG - 1可以支持4k的视频，但难以提供更高分辨率的视频编码并且标识硬件的支持能力。
* 支持只有一个颜色空间，4:2:0。

### **MPEG-2**

#### **MPEG-2内容介绍**

　　MPEG-2作为ISO/IEC 13818正式发布，通常用来为广播信号提供视频和音频编码，包括卫星电视、有线电视等。MPEG-2经过少量修改后，也成为DVD产品的内核技术。

#### **MPEG-2有11部分，具体如下：**

##### 第一部（Part 1）：系统-描述视频和音频的同步和多路技术

　　正式名称是 ISO/IEC 13818-1或 ITU-T中的H.222.0

　　MPEG-2的系统描述部分（第1部分）定义了传输流，它用来一套在非可靠介质上传输数字视频信号和音频信号的机制，主要用在广播电视领域。

　　定义了两个不同但相关的容器格式，MPEG transport stream和MPEG program stream，也就是图中的TS和PS。MPEG传输流（TS）为携带可损数字视频和音频，媒体流的开始和结束可以不标识出来，就像广播或者磁带，其中的 例子包括ATSC，DVB，SBTVD 和HDV。MPEG-2系统还定义了MPEG节目流（PS），它为以文件为基础的媒体设计一个容器格式，用于硬盘驱动器，光盘 和闪存。

　　 MPEG-2 PS（节目流）是为在存储介质保存视频信息而开发的。 MPEG-2 TS（传输流）是为在网络传输视频信息而开发的。目前，MPEG-2 TS最广泛地应用是DVB系统。TS流与PS流的区别在于TS流的包结构是固定度的，而PS流的包结构是可变长度。 PS包与TS包在结构上的这种差异，导致了它们对传输误码具有不同的抵抗能力，因而应用的环境也有所不同。TS码流由于采用了固定长度的包结构，当传输误 码破坏了某一TS包的同步信息时，接收机可在固定的位置检测它后面包中的同步信息，从而恢复同步，避免了信息丢失。而PS包由于长度是变化的，一旦某一 PS包的同步信息丢失，接收机无法确定下一包的同步位置，就会造成失步，导致严重的信息丢失。因此，在信道环境较为恶劣，传输误码较高时，一般采用TS码流；而在信道环境较好，传输误码较低时，一般采用PS码流。由于 TS码流具有较强的抵抗传输误码的能力，因此目前在传输媒体中进行传输的MPEG-2码流基本上都采用了TS码流的包格。

##### 第二部（Part 2）：视频-视频压缩

　　正式名称是 ISO/IEC 13818-2或 ITU-T H.262。

　　提供隔行扫描和非隔行扫描视频信号的压缩编解码器。

　　MPEG-2的第二部分即视频部分和MPEG-1类似，但是它提供对隔行扫描视频显示模式的支持（隔行扫描广泛应用在广播电视领域）。MPEG-2视频并没有对低位速率（小于1Mbps）进行优化，在 3Mbit/s及以上位速率情况下，MPEG-2明显优于MPEG-1。 MPEG-2向后兼容，也即是说，所有符合标准的MPEG-2解码器也能够正常播放MPEG-1视频流。

　　MPEG-2技术也应用在了HDTV传输系统中。MPEG-2 不光运用于 DVD-Video ，现在大部 分 HDTV（高清电视）也采用 MPEG-2 编码，分辨率达到了 1920x1080。由于 MPEG-2 的普及，本来为 HDTV 准备 的 MPEG-3 最终宣告放弃。

　　MPEG-2视频通常包含多个GOP（GroupOf Pictures），每一个GOP包含多个帧（frame）。帧的帧类（frame type）通常包括I-帧（I-frame）、P-帧（P-frame）和B-帧（B-frame）。其中I-帧采用帧内编码，P-帧采用前向估计，B- 帧采用双向估计。一般来说输入视频格式是25（CCIR标准）或者29.97（FCC）帧／秒。

　　MPEG-2支持隔行扫描和逐行扫描。在逐行扫描模式下，编码的基本单元是帧。在隔行扫描模式下，基本编码可以是帧，也可以是场（field）。

　　原始输入图像首先被转换到YCbCr颜色空间。其中Y是亮度，Cb和Cr是两个色度通道。 Cb指蓝色色 度，Cr指红色色度。对于每一通道，首先采用块分区，然后形成“宏块”（macroblocks），宏块构成了编码的基本单元。每一个宏块再分 区成8x8的小块。色度通道分区成小块的数目取决于初始参数设置。例如，在常用的4:2:0格式下，每个色度宏块只采样出一个小块，所以三个通道宏块能够 分区成的小块数目是4+1+1=6个。

　　对于I-帧，整幅图像直接进入编码过程。对于P-帧和B-帧，首先做运动补偿。通常来说，由于相邻帧之间的相关 性很强，宏块可以在前帧和后帧中对应相近的位置找到相似的区域匹配的比较好，这个偏移量作为运动向量被记录下来，运动估计重构的区域的误差被送到编码器中编码。

　　对于每一个8×8小块，离散余弦变换把图像从空间域转换到频域。得到的变换系数被量化并重新组织排列顺序，从而增加长零的可能性。之后做游程编码（run-length code）。最后作哈夫曼编码（Huffman Encoding）。

　　I帧编码是为了减少空间域冗余，P帧和B帧是为了减少时间域冗余。

　　GOP是由固定模式的一系列I帧、P帧、B帧组成。常用的结构由15个帧组成，具有以下形式 IBBPBBPBBPBBPBB。GOP中各个帧的比例的选取和带宽、图像的质量要求有一定关系。例如因为B帧的压缩时间可能是I帧的三倍，所以对于计算 能力不强的某些实时系统，可能需要减少B帧的比例。

　　MPEG-2输出的比特流可以是匀速或者变速的。最大比特率，例如在DVD应用上，可达10.4 Mbit/s。如果要使用固定比特率，量化尺度就需要不断的调节以产生匀速的比特流。但是，提高量化尺度可能带来可视的失真效果。比如马赛克现象。

##### 第三部（Part 3）：音频-音频压缩

　　MPEG-2的第三部分定义了音频压缩标准。MPEG-2 BC（Backwards compatible），后向兼容MPEG-1音频。该部分改进了MPEG-1的音频压缩，支持两通道以上的音频，可高达5.1多声道。MPEG-2音频 压缩部分也保持了向后兼容的特点（也称为MPEG - 2 BC），允许的MPEG - 1音频解码器解码两个主立体声组件。还定义音频MPEG-1 Layer I,II ,III额外的比特率和采样频率。

　　例如mp2，是MPEG-1 Audio level 2，标准有：ISO/IEC 11172-3, ISO/IEC 13818-3。MPEG-1Layer II 定义在 ISO/IEC 11172-3，也就是MPEG-1的第三部分，在 ISO/IEC 13818-3，也就是MPEG-2的第3部分定义扩展。

##### 第四部（Part 4）：测试规范

　　描述测试程序。

##### 第五部（Part 5）：仿真软件

　　描述软件仿真系统。

##### 第六部（Part 6）：DSM-CC(Digital Storage Media Commandand Control)扩展

　　描述DSM-CC（数字存储媒体命令及控制）扩展。

##### 第七部（Part 7）：Advanced Audio Coding (AAC)

　　MPEG-2的第七部分定义了不能向后兼容的音频压缩（也成为MPEG-2 NBC）。也成为MPEG-2 NBC（not-backwards compatible MPEG-1Audio）。该部分提供了更强的音频功能。通常我们所说的MPEG-2 AAC指的就是这一部分。AAC即Advanced Audio Coding。 AAC是比以前的MPEG音频标准的效率，并在某个程度上没有它的前任MPEG-1 Layer3（MP3）复杂，它没有复杂的混合滤波器（hybrid filter bank）。它支持从1到48个通道，采样率从8-96千赫，多渠道，多语种和多节目（multiprogram）能力。AAC也在MPEG-4标准的第 3部分描述。

##### 第八 部（Part 8）：

　　 已取消。

##### 第九部（Part 9）：实时接口扩展

　　实时接口扩展。

##### 第十部（Part 10）：DSM-CC一致性扩展

　　DSM-CC一致性扩展。

##### 第十一部（Part 11） ：IP

　　知识产权管理（IPMP）。XML定义在ISO/IEC23001-3。MPEG-2内核技术大约涉及640个专利，这些专利主要集中在20间公司和一间大学。

#### MPEG- 2 profile和level

**MPEG-2提供广泛的应用， 对于大部分的应用，即不现实的也过于昂贵，去支持整个标准，通常只支持子集，因此标准定义了profile和level来表示这些子集。profile定 义特性相关，例如压缩算法，色度格式等。level定义性能相关，例如最大比特率，最大帧大小等。一个应用程序应当通过profile和level来表示他的能力。profile和level的组合构成MPEG-2视频编码标准在某种特定应用下的子集。对某一输入格式的图像，采用特定集合的压缩编码工具，产生规定速率范围内的编码码流 。**例如一台DVD播放机可以说，它支持最多的主要profile和主要level（通常写为MP@ML）。

MPEG-2主要的profile：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **名称** | **英文** | **中文** | **图像编码类型** | **色度格式**YCbCr | **长宽比** | **伸缩模式** |
| SP | Simple Profile | 简单类 | I帧、P帧 | 4:2:0 | 4:3或16:9 |  |
| MP | Main Profile | 主类 | I帧、P帧、B帧 | 4:2:0 | 4:3 或16:9 |  |
| SNR | SNR Scalable profile | 信噪比分层类 | I帧、P帧、B帧 | 4:2:0 | 4:3 或16:9 | 信噪比可伸缩 |
| Spatial | Spatially scalable profile | 空间可分层类 | I帧、P帧、B帧 | 4:2:0 | 4:3 或16:9 | 信噪比或空间可伸缩 |
| 442P | 4:2:2 Profile |  | I帧、P帧、B帧 | 4:2:2 |  |  |
| HP | High profile | 高类 | I帧、P帧、B帧 | 4:2:0或  4:2:2 | 4:3 或16:9 | 信噪比或空间可伸缩 |

MPEG-2主要的level：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **英文** | **帧频** | **最大长×最大宽** | **每秒最大亮度样本 （约为高×宽×帧频率）** | **最大比特率 （Mbit/s）** |
| LL | Low Level | 23.976, 24, 25, 29.97, 30 | 352×288 | 3,041,280 | 4 |
| ML | Main Level | 23.976, 24, 25, 29.97, 30 | 720×576 | 10,368,000，例外为：HP中4:2:0为14,475,600，4:2:2为11,059,200 | 15 |
| H-14 | High-1440 level | 23.976, 24, 25, 29.97, 30, 50, 59.94, 60 | 1440×1152 | 47,001,600，例外为：HP中4:2:0为62,668,800 | 60 |
| HL | High level | 23.976, 24, 25, 29.97, 30, 50, 59.94, 60 | 1920×1152 | 62,668,800，例外为：HP中4:2:0为83,558,400 | 80 |

组合例子

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Profile @ Level** | **Resolution (px)** | **Framerate max. (Hz)** | **Sampling** | **Bitrate (Mbit/s)** | **Example Application** |
| **SP@LL** | 176 × 144 | 15 | 4:2:0 | 0.096 | Wireless handsets |
| **SP@ML** | 352 × 288 | 15 | 4:2:0 | 0.384 | PDAs |
| 320 × 240 | 24 |
| **MP@LL** | 352 × 288 | 30 | 4:2:0 | 4 | Set-top boxes (STB) |
| **MP@ML** | 720 × 480 | 30 | 4:2:0 | 15 (DVD: 9.8) | DVD, SD-DVB |
| 720 × 576 | 25 |
| **MP@H-14** | 1440 × 1080 | 30 | 4:2:0 | 60 (HDV: 25) | HDV |
| 1280 × 720 | 30 |
| **MP@HL** | 1920 × 1080 | 30 | 4:2:0 | 80 | ATSC 1080i, 720p60, HD-DVB (HDTV).  (Bitrate for terrestrial transmission is limited to 19.39Mbit/s) |
| 1280 × 720 | 60 |
| **422P@LL** |  |  | 4:2:2 |  |  |
| **422P@ML** | 720 × 480 | 30 | 4:2:2 | 50 | Sony IMX using I-frame only, Broadcast "contribution" video (I&P only) |
| 720 × 576 | 25 |
| **422P@H-14** | 1440 × 1080 | 30 | 4:2:2 | 80 | Potential future MPEG-2-based HD products from Sony and Panasonic |
| 1280 × 720 | 60 |
| **422P@HL** | 1920 × 1080 | 30 | 4:2:2 | 300 | Potential future MPEG-2-based HD products from Panasonic |
| 1280 × 720 | 60 |

#### MPEG- 2在DVD上的应用

DVD中采用了 MPEG-2标准并引入如下技术参数限制：  
\* 分辨率  
o 720 x 480, 704 x 480, 352 x 480, 352 x 240 像素（NTSC制式）  
o 720 x 576, 704 x 576, 352 x 576, 352 x 288 像素（PAL制式）  
\* 纵横比  
o 4:3  
o 16:9  
\* 帧率（帧播放速度）  
o 59.94 场／秒，23.976帧／秒，29.97帧／秒（NTSC）  
o 50 场／秒，25帧／秒（PAL）  
\* 视频+音频 比特率  
o 平均最大缓冲区 9.8 Mbit/s  
o 峰值 15 Mbit/s  
o 最小值 300 Kbit/s  
\* YUV 4:2:0  
\* 字幕支持  
\* 内嵌字幕支持（NTSC only）  
\* 音频  
o LPCM编码：48kHz或96kHz；16或24-bit；最多可达6声道  
o MPEG Layer 2 (MP2)：48 kHz，可达5.1声道  
o 杜比数字-Dolby Digital（DD，也称为AC-3）：48 kHz，32-448kbit/s，可达5.1声道  
o 数字家庭影院系统-Digital Theater Systems (DTS)：754 kbit/s或1510 kbit/s  
o NTSC制式DVD必须包含至少一道LPCM或Dolby Digital  
o PAL制式DVD必须包含至少一道MPEG Layer 2、LPCM或者Dolby Digital  
\* GOP结构  
o 必须为GOP提供串行的头信息  
o GOP最大可含帧数目：18 (NTSC) / 15 (PAL)

#### MPEG- 2在DVB上的应用

DVB-MPEG相关技术参数：  
\* 必须符合以下一种分辨率：  
o 720 × 480 像素，24/1.001，24，30/1.001或30帧／秒  
o 640 × 480 像素，24/1.001，24，30/1.001或30帧／秒  
o 544 × 480 像素，24/1.001，24，30/1.001或30帧／秒  
o 480 × 480 像素，24/1.001，24，30/1.001或30帧／秒  
o 352 × 480 像素，24/1.001，24，30/1.001或30帧／秒  
o 352 × 240 像素，24/1.001，24，30/1.001或30帧／秒  
o 720 × 576 像素，25帧／秒  
o 544 × 576 像素，25帧／秒  
o 480 × 576 像素，25帧／秒  
o 352 × 576 像素，25帧／秒  
o 352 × 288 像素，25帧／秒

#### MPEG- 2和NTSC

必须符合以下一种分辨率：  
o 1920 × 1080 像素，最多60帧／秒（1080i）  
o 1280 × 720 像素，最多60帧／秒（720p）  
o 720 × 576 像素，最多50帧／秒，25帧／秒（576i，576p）  
o 720 × 480 像素，最多60帧／秒，30帧／秒（480i，480p）  
o 640 × 480 像素，最多60帧／秒  
注：1080i按 1920×1088像素编码，但是最后8行在显示时抛弃。

#### 对YCbCr的补充资料

YCbCr不是一种绝对色彩空间，是YUV压缩和偏移的版本。右图为UV色版。

Y（Luma，Luminance）视讯，也就是灰阶值。UV 视作表示彩度的 C（Chrominance或Chroma）。主要的采样（subsample）格式有YCbCr 4:2:0、YCbCr 4:2:2、YCbCr 4:1:1和 YCbCr 4:4:4。YUV的表示法称为 A:B:C 表示法：

\* 4:4:4 表示完全取样。  
\* 4:2:2 表示 2:1 的水平取样，没有垂直下采样。  
\* 4:2:0 表示 2:1 的水平取样，2:1 的垂直下采样。  
\* 4:1:1 表示 4:1 的水平取样，没有垂直下采样。

最常用Y:UV记录的比重通常 1:1 或2:1，DVD-Video 是以 YUV 4:2:0 的方式记录，也就是我们俗称的I420，YUV4:2:0 并不是说只有U（即 Cb）, V（即 Cr）一定为 0，而是指U：V互相援引，时见时隐，也就是说对于每一个行，只有一个U或者V份量，如果一行是4:2:0的话，下一行就是4:0:2，再下一行是 4:2:0...以此类推。

以上来自wiki资料的整理。

## [编解码学习笔记（四）：Mpeg系列——Mpeg 4](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://javasogo.iteye.com%2Fblog%2F1015210)

在上次对MPEG-2的学习整理中，有一个疑惑，双声道理解，就是左右立体声，但是5.1声道是什么？我们经常看到杜比5.1声道的说法。“0.1”声道具体指什么？今天去wiki查了一下，相关内容也整理入我们的学习笔记。

本文档资料来源:

* wiki
* [http://baike.baidu.com/view/190268.htm](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://baike.baidu.com%2Fview%2F190268.htm)
* [http://baike.baidu.com/view/25047.htm](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://baike.baidu.com%2Fview%2F25047.htm)

### 5.1声道

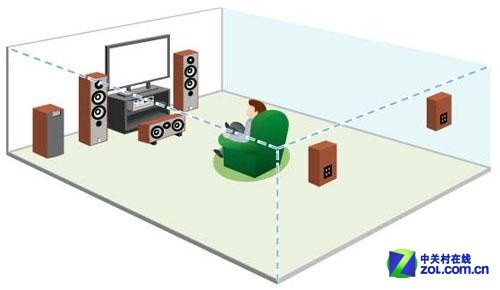
　　使用杜比数字技术下，最标准常用的是5.1声道设置，但杜比数字容许一系列不同声道的选择。全部可供选择的声道如下列所示:

* 单声道（中央）
* 双声道立体声（左、右），选择性地交叉应用杜比环回
* 三声道立体声（左、中、右）
* 双声道立体声加单环回（左、右、环回）
* 三声道立体声加单环回（左、中、右、环回）
* 四声道环回立体声（左前、右前、左后、右后）
* 五声道环回立体声（左前、中、右前、左后、右后）

　　以上所有这些设置可选择性地使用低频效果和杜比数字EX矩阵编码中加入附加后环绕声道。杜比编码技术是向下兼容 的，很多杜比播放器／解码器均备有向下混音作用是发布不同声道至可供使用的扬声器。这包括一些功能例如声音数据通过前扬声器播放（如适用），和当中央扬声器不适用时发布中央频道至左或右扬声器。或当用户只有2.0喇叭时，杜比解码器能把多声道信号混音编码为 2.0立体声。

　　在5.1, 7.1 或其他等文字中，'.1'指的是低频LFE声道。

　　其实5.1声道就是使用5个喇叭和1个超低音扬声器来实现一种身临其境的音乐播放方式，它是由杜比公司开发的，所以叫做“杜比5.1声道”。在5.1声道系统里采用左(L)、中(C)、右(R)、左后(LS)、右后(RS)五个方向输出声音，使人产生犹如身临音乐厅的感觉。五个声道相互独立，其中“.1” 声道，则是一个专门设计的超低音声道。正是因为前后左右都有喇叭，所以就会产生被音乐包围的真实感。如图所示。



### MPEG-4

#### 总体介绍

　　MPEG-4是一套用于音频、视频信息的压缩编码标准， 由国际标准化组织（ISO）和国际电工委员会（IEC）下属的“動態影像专家组”（Moving Picture Experts Group，即MPEG） 制定，第一版在1998年10月通過，第二版在1999年12月通過。MPEG-4格式的主要用途在於網上流媒体、光碟、語音傳送（視訊電話），以及電視廣播。MPEG-4作为ISO/IEC14496正式发布。ISO/IEC 14496-Coding of audio-visual object （AV对象编码）。

　　为了应对网络传输等环境，传统的 MPEG-1/2 已经不能适应，所以促使了 MPEG-4 的诞生。 与 MPEG-1和MPEG-2相比，MPEG-4的特点是其更适于交互AV服务以及远程监控。MPEG-4是第一个使你由被动变为主动(不再只是观看，允许 你加入其中，即有交互性)的动态图像标准，它的另一个特点是其综合性。从根源上说，MPEG-4试图将自然物体与人造物体相溶合 (视觉效果意义上的)。MPEG-4的设计目标还有更广的适应性和更灵活的可扩展性。 MPEG-4 采用了一系列新技术，来满足在低带宽下传输较高视频质量的需求。DivX，XviD，MS MPEG4 都是采用的MPEG-4 视频编码，除了在 DVDRip 上面的应用，3GPP现在也接纳了 MPEG-4 作为视频编码方案。

　　最初MPEG-4的主要目的是用于低比特率下的视频通信，但是作为一个多媒体的编码标准，它的范围最后得到了扩展。在技术方面MPEG-4允许不同的软件/硬件开发商创建多媒体对象来提供更好的适应性、灵活性，为数字电视，动态图像，互联网等业务提供更好的质量。

　　MPEG-4提供范围从每秒几k比特到每秒数十兆比特的，它具有下面功能：

* 改善MPEG-2的编码效率
  + MPEG-4基于更高的编码效率。同已有的或即将形成的其它标准相比，在相同的比特率下，它基于更高的视觉听觉质量，这就 使得在低带宽的信道上传送视频、音频成为可能。同时MPEG-4还能对同时发生的数据流进行编码。一个场景的多视角或多声道数据流可以高效、同步地合成为 最终数据流。这可用于虚拟三维游戏、三维电影、飞行仿真练习等。
* 提供混合媒体数据（视频，音频，语音）的编码能力
* 差错容忍使得内容稳定传输。
  + 当在传输有误码或丢包现象时，MPEG4受到的影响很小，并且能迅速恢复。
* 提供受众视听场景的互动能力，MPEG-4终端用户提供不同的对象支持各种互动要求。
  + MPEG-4提供了基于内容的多媒体数据访问工具，如索引、超级链接、上传、下载、删除等。利用这些工具，用户可以方便地 从多媒体数据库中有选择地获取自己所需的与对象有关的内容，并提供了内容的操作和位流编辑功能，可应用于交互式家庭购物，淡入淡出的数字化效果等。 MPEG-4提供了高效的自然或合成的多媒体数据编码方法。它可以把自然场景或对象组合起来成为合成的多媒体数据。
* MPEG-4对传输数据网是透明的，它可以兼容各种网络。
  + MPEG-4提供了易出错环境的鲁棒性，来保证其在许多无线和有线网络以及存储介质中的应用，此外，MPEG-4还支持基于内容的的可分级性，即把内容、质量、复杂性分成许多小块来满足不同用户的不同需求，支持具有不同带宽，不同存储容量的传输信道和接收端。
  + 这些特点无疑会加速多媒体应用的发展，从中受益的应用领域有：因特网多媒体应用；广播电视；交互式视频游戏；实时可视通 信；交互式存储媒体应用；演播室技术及电视后期制作；采用面部动画技术的虚拟会议；多媒体邮件；移动通信条件下的多媒体应用；远程视频监控；通过ATM网 络等进行的远程数据库业务等。

#### MPEG-4视频编码核心思想

　　在MPEG-4制定之前，MPEG-1、MPEG-2、H.261、H.263都是采用第一代压缩编码技术，着 眼于图像信号的统计特性来设计编码器，属于波形编码的范畴。第一代压缩编码方案把视频序列按时间先后分为一系列帧，每一帧图像又分成宏块以进行运动补偿和编码，这种编码方案存在以下缺陷：

* 将图像固定地分成相同大小的块，在高压缩比的情况下会出现严重的块效应，即马赛克效应；
* 不能对图像内容进行访问、编辑和回放等操作；
* 未充分利用人类视觉系统（HVS，Human Visual System）的特性。

　　MPEG-4则代表了基于模型/对象的第二代压缩编码技术，它充分利用了人眼视觉特性，抓住了图像信息传输的本质，从轮廓、纹理思路出发，支持基于视觉内容的交互功能，这适应了多媒体信息的应用由播放型转向基于内容的访问、检索及操作的发展趋势。

　　AV对象（AVO，AudioVisual Object）是MPEG-4为支持基于内容编码而提出的重要概念。对象是指在一个场景中能够访问和操纵的实体，对象的划分可根据其独特的纹理、运动、形状、模型和高层语义为依据。在MPEG-4中所见的视音频已不再是过去MPEG-1、MPEG-2中图像帧的概念，而是一个个视听场景（AV场景），这些 不同的AV场景由不同的AV对象组成。AV对象是听觉、视觉、或者视听内容的表示单元，其基本单位是原始AV对象，它可以是自然的或合成的声音、图像。原 始AV对象具有高效编码、高效存储与传输以及可交互操作的特性，它又可进一步组成复合AV对象。因此MPEG-4标准的基本内容就是对AV对象进行高效编 码、组织、存储与传输。AV对象的提出，使多媒体通信具有高度交互及高效编码的能力，AV对象编码就是MPEG-4的核心编码技术。

　　MPEG-4实现基于内容交互的首要任务就是把视频/图像分割成不同对象或者把运动对象从背景中分离出来，然后针对不同对象采用相应编码方法，以实现高效压缩。因此视频对象提取即视频对象分割，是MPEG-4视频编码的关键技术，也是新一代视频编码的研究热点和难点。

　　MPEG-4不仅可提供高压缩率，同时也可实现更好的多媒体内容互动性及全方位的存取性，它采用开放的编码系统，可随时加入新的编码算法模块，同时也可根据不同应用需求现场配置解码器，以支持多种多媒体应用。

### MPEG-4各部分

　　MPEG-4由一系列的子标准组成，被称为部，包括以下的部分。对于媒体编解码，重点关注Part2,Part 3, Part 10。

#### 第一部（ISO/IEC 14496-1）：系统

　　描述视訊和音訊的同步以及混合方式（Multiplexing，简写为MUX）。定义了 MP4 容器格式, 支持类似 DVD 菜单这样的直观和互动特性等。

#### 第二部（ISO/IEC 14496-2）：视频

　　定义了一个对各种视觉信息（包括视訊、静止纹理、计算机合成图形等等）的编解码器。对视訊部分来说，众多”Profiles”中很常用的一种是Advanced SimpleProfile (ASP)，例如XviD编码就 属于MPEG-4Part 2。包括 3ivx, DivX4/Project Mayo, DivX 5, Envivio,ffmpeg/ffds, mpegable, Nero Digital, QuickTime, Sorenson, XviD 等常见的视频格式, 需要注意的是 Divx 3.11, MS MPEG-4, RV9/10, VP6,WMV9 并不属于标准的 MPEG-4 标准。

#### 第三部（ISO/IEC 14496-3）：音频

　　定义了一个对各种音訊信号进行编码的编解码器的集合。包括高级音訊编码（Advanced Audio Coding，缩写为AAC） 的若干变形和其他一些音频／语音编码工具。即 AAC 音频标准, 包括 LCAAC, HE AAC 等, 支持 5.1 声道编码, 可以用更低的码率实现更好的效果 (相对于 MP3, OGG 等) 。

#### 第四部（ISO/IEC 14496-4）：一致性

　　定义了对本标准其他的部分进行一致性测试的程序。

#### 第五部（ISO/IEC 14496-5）：参考软件

　　提供了用于演示功能和说明本标准其他部分功能的软件。

#### 第六部（ISO/IEC 14496-6）：多媒体传输集成框架

　　即DMIF：Delivery Multimedia IntegrationFramework

#### 第七部（ISO/IEC 14496-7）：优化的参考软件

　　提供了对实现进行优化的例子（这裡的实现指的是第五部分）。

#### 第八部（ISO/IEC 14496-8）：在IP网络上传输

　　定义了在IP网络上传输MPEG-4内容的方式。

#### 第九部（ISO/IEC 14496-9）：参考硬件

　　提供了用于演示怎样在硬件上实现本标准其他部分功能的硬件设计方案。

#### 第十部（ISO/IEC 14496-10）：进阶视频编码，也即ITU H.264，常写为H.264／AVC

　　或称高级视频编码（Advanced Video Coding，缩写为AVC）：定义了一个视频编解码器（codec），AVC和XviD都属于MPEG-4编码，但由于AVC属于MPEG-4Part 10，在技术特性上比属于MPEG-4 Part2的XviD要先进。另外从技术上讲，它和ITU-T H.264标准是一致的，故全称为MPEG-4 AVC/H.264。

#### 第十一部（ISO/IEC 14496-11）：场景描述和应用引擎

　　可用于多种profile（包括2D和3D版本）的互交互媒体。修订了MPEG-4 Part 1:2001以及Part1的两个修订方案。它定义了应用引擎（交付，生命周期，格式，可下载Java字节代码应用程序的行为），二进制场景格式 （BIFS：Binary Format for Scene），可扩展MPEG-4文本格式（一种使用XML描述MPEG-4多媒体内容的文本格式）系统level表述。也就是MPEG-4 Part21中的BIFS，XMT，MPEG-J。

#### 第十二部（ISO/IEC 14496-12）：基于ISO的媒体文件格式

　　定义了一个存储媒体内容的文件格式。

#### 第十三部（ISO/IEC 14496-13）：IP

　　知识产权管理和保护（IPMP for Intellectual Property Management and Protection）拓展。

#### 第十四部（ISO/IEC 14496-14）：MPEG-4文件格式

　　定义了基于第十二部分的用于存储MPEG-4内容的視訊檔案格式。

#### 第十五部（ISO/IEC 14496-15）：AVC文件格式

　　定义了基于第十二部分的用于存储第十部分的视频内容的文件格式。

#### 第十六部（ISO/IEC 14496-16）：动画框架扩展

　　动画框架扩展（AFX : Animation Framework eXtension）。

#### 第十七部（ISO/IEC 14496-17）：同步文本字幕格式

　　尚未完成－2005年1月达成”最终委员会草案”，FCD: Final Committee Draft。

#### 第十八部（ISO/IEC 14496-18）：字体压缩和流式传输（针对公开字体格式）。

#### 第十九部（ISO/IEC 14496-19）：综合用材质流（Synthesized TextureStream）。

#### 第二十部（ISO/IEC 14496-20）：简单场景表示

　　LASeR for Lightweight Scene Representation，尚未完成－2005年1月达成”最终委员会草案”，FCD for Final Committee Draft。

#### 第二十一部（ISO/IEC 14496-21）：用于描绘（Rendering）的MPEG-J拓展

　　尚未完成－2005年1月达成“委员会草案”，CD for Committee Draft）。

### Profile和Level

　　MPEG-4提供大量的编码方式和丰富的设置。 和MPEG-2一样，应用一般不可能支持MPEG-4全集，通过profile和level来描述子集。这些子集，通过 “profile”来表明解码器要求，为了避免计算的复杂，每个profile都有一个或者多个“level”。profile和level的有效组合使得编码生成器只需实现标准中所需的子集，同时保持与其他MPEG-4设备的互通。（解码支持范围通常比编码支持范围大），检查其他MPEG-4设备是否符合标准，即一致性测试。

　　对于H.264/AVC（也就是MPEG-4 Part 4）提供下面的profile：

Feature support in particular profiles

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Feature** | **CBP** | **BP** | **XP** | **MP** | **HiP** | **Hi10P** | **Hi422P** | **Hi444PP** |
| **B slices** | No | No | Yes | Yes | Yes | Yes | Yes | Yes |
| **SI and SP slices** | No | No | Yes | No | No | No | No | No |
| **Flexible macroblock ordering (FMO)** | No | Yes | Yes | No | No | No | No | No |
| **Arbitrary slice ordering (ASO)** | No | Yes | Yes | No | No | No | No | No |
| **Redundant slices (RS)** | No | Yes | Yes | No | No | No | No | No |
| **Data partitioning** | No | No | Yes | No | No | No | No | No |
| **Interlaced coding (PicAFF, MBAFF)** | No | No | Yes | Yes | Yes | Yes | Yes | Yes |
| **CABAC entropy coding** | No | No | No | Yes | Yes | Yes | Yes | Yes |
| **8×8 vs. 4×4 transform adaptivity** | No | No | No | No | Yes | Yes | Yes | Yes |
| **Quantization scaling matrices** | No | No | No | No | Yes | Yes | Yes | Yes |
| **Separate Cb and Cr QP control** | No | No | No | No | Yes | Yes | Yes | Yes |
| **Monochrome (4:0:0)** | No | No | No | No | Yes | Yes | Yes | Yes |
| **Chroma formats** | 4:2:0 | 4:2:0 | 4:2:0 | 4:2:0 | 4:2:0 | 4:2:0 | 4:2:0/4:2:2 | 4:2:0/4:2:2/4:4:4 |
| **Sample depths (bits)** | 8 | 8 | 8 | 8 | 8 | 8 to 10 | 8 to 10 | 8 to 14 |
| **Separate color plane coding** | No | No | No | No | No | No | No | Yes |
| **Predictive lossless coding** | No | No | No | No | No | No | No | Yes |

　　level用来表明一个profile的解码器的性能要求的范围，例如最大图片分配了，帧频，比特率等等。对于解码器，一个指定的level要求可以对该level以及低于该level的码流进行解码。（A decoder that conforms to agiven level is required to be capable of decoding all bitstreams that areencoded for that level and for all lower levels. 来源：[http://en.wikipedia.org/wiki/H.264/MPEG-4\_AVC](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FH.264%2FMPEG-4_AVC)）

Levels with maximum property values

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Level** | **Max macroblocks** | | **Max video bit rate (VCL)** | | | | **Examples for high resolution @  frame rate  (max stored frames)** |
| **per second** | **per frame** | **BP, XP, MP  (kbit/s)** | **HiP  (kbit/s)** | **Hi10P  (kbit/s)** | **Hi422P, Hi444PP  (kbit/s)** |
| **1** | 1,485 | 99 | 64 | 80 | 192 | 256 | 128×96@30.9 (8)  176×144@15.0 (4) |
| **1b** | 1,485 | 99 | 128 | 160 | 384 | 512 | 128×96@30.9 (8)  176×144@15.0 (4) |
| **1.1** | 3,000 | 396 | 192 | 240 | 576 | 768 | 176×144@30.3 (9)  320×240@10.0 (3)  352×288@7.5 (2) |
| **1.2** | 6,000 | 396 | 384 | 480 | 1,152 | 1,536 | 320×240@20.0 (7)  352×288@15.2 (6) |
| **1.3** | 11,880 | 396 | 768 | 960 | 2,304 | 3,072 | 320×240@36.0 (7)  352×288@30.0 (6) |
| **2** | 11,880 | 396 | 2,000 | 2,500 | 6,000 | 8,000 | 320×240@36.0 (7)  352×288@30.0 (6) |
| **2.1** | 19,800 | 792 | 4,000 | 5,000 | 12,000 | 16,000 | 352×480@30.0 (7)  352×576@25.0 (6) |
| **2.2** | 20,250 | 1,620 | 4,000 | 5,000 | 12,000 | 16,000 | 352×480@30.7(10)  352×576@25.6 (7)  720×480@15.0 (6)  720×576@12.5 (5) |
| **3** | 40,500 | 1,620 | 10,000 | 12,500 | 30,000 | 40,000 | 352×480@61.4 (12)  352×576@51.1 (10)  720×480@30.0 (6)  720×576@25.0 (5) |
| **3.1** | 108,000 | 3,600 | 14,000 | 17,500 | 42,000 | 56,000 | 720×480@80.0 (13)  720×576@66.7 (11)  1280×720@30.0 (5) |
| **3.2** | 216,000 | 5,120 | 20,000 | 25,000 | 60,000 | 80,000 | 1,280×720@60.0 (5)  1,280×1,024@42.2 (4) |
| **4** | 245,760 | 8,192 | 20,000 | 25,000 | 60,000 | 80,000 | 1,280×720@68.3 (9)  1,920×1,080@30.1 (4)  2,048×1,024@30.0 (4) |
| **4.1** | 245,760 | 8,192 | 50,000 | 62,500 | 150,000 | 200,000 | 1,280×720@68.3 (9)  1,920×1,080@30.1 (4)  2,048×1,024@30.0 (4) |
| **4.2** | 522,240 | 8,704 | 50,000 | 62,500 | 150,000 | 200,000 | 1,920×1,080@64.0 (4)  2,048×1,080@60.0 (4) |
| **5** | 589,824 | 22,080 | 135,000 | 168,750 | 405,000 | 540,000 | 1,920×1,080@72.3 (13)  2,048×1,024@72.0 (13)  2,048×1,080@67.8 (12)  2,560×1,920@30.7 (5)  3,680×1,536@26.7 (5) |
| **5.1** | 983,040 | 36,864 | 240,000 | 300,000 | 720,000 | 960,000 | 1,920×1,080@120.5 (16)  4,096×2,048@30.0 (5)  4,096×2,304@26.7 (5 |

## [编解码学习笔记（五）：Mpeg系列——AAC音频](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://javasogo.iteye.com%2Fblog%2F1015209)

### AAC在MPEG2和MPEG4中定义。

扩展名：.m4a, .m4b, .m4p, .m4v, .m4r, .3gp, .mp4, .aac   
互联网媒体类型：audio/aac, audio/aacp, audio/3gpp, audio/3gpp2,audio/mp4, audio/MP4A-LATM, audio/mpeg4-generic  
格式：有损数据压缩   
延伸自：MPEG-2 音频   
标准：ISO/IEC 13818-7（MPEG-2第7部）, ISO/IEC 14496-3（MPEG-4第3部）

AAC（Advanced Audio Coding），中文称为“高级音频编码”，出现于1997年，基于 MPEG-2的音频编码技术。由Fraunhofer IIS、杜比实验室、AT&T、Sony（索尼）等公司共同开发，目的是取代MP3格式。2000年，MPEG-4标准出现后，AAC 重新集成了其特性，加入了SBR技术和PS技术，为了区别于传统的 MPEG-2 AAC 又称为 MPEG-4 AAC。

### AAC格式的主要扩展名有三种：

* AAC —— 使用MPEG-2 Audio Transport Stream( ADTS，参见MPEG-2 )容器，区别于使用MPEG-4容器的MP4/M4A格式，属于传统的AAC编码（FAAC默认的封装，但FAAC亦可输出 MPEG-4 封装的AAC）
* MP4 —— 使用了MPEG-4 Part 14（第14部分）的简化版即3GPP Media Release 6 Basic (3gp6，参见3GP ) 进行封装的AAC编码（Nero AAC 编码器仅能输出MPEG-4封装的AAC）；
* M4A —— 为了区别纯音频MP4文件和包含视频的MP4文件而由苹果(Apple)公司使用的扩展名，Apple iTunes 对纯音频MP4文件采用了".M4A"命名。M4A的本质和音频MP4相同，故音频MP4文件亦可直接更改扩展名为M4A。

### AAC 特性

　　作为一种高压缩比的音频压缩算法，AAC压缩比通常为18：1，也有资料说为20：1，远胜mp3； 在音质方面，由于采用多声道，和使用低复杂性的描述方式，使其比几乎所有的传统编码方式在同规格的情况下更胜一筹。不过直到2006年， 使用这一格式储存音乐的并不多，可以播放该格式的mp3播放器更是少之又少，目前所知仅有苹果iPod、Sony Walkman（NWZ-A、NWZ-S、NWZ-E、NWZ-X系列）、任天堂NDSi。魅 族M8，此外计算机上很多音乐播放软体都支持AAC（前提是安装过AAC解码器），如苹果iTunes。但在移动电话领域，AAC的支持度已很普 遍，Nokia、Sony Ericsson、Motorola等品牌均在其中高端产品中支持 AAC（一开始主要是LC-AAC，随着移动电话性能的发展，HE-AAC的支持也已广泛）。

　　AAC可以支持多达48个音轨，15个低频（LFE）音轨，5.1多声道支持，更高的采样率（最高可达 96kHz，音频CD为44.1kHz）和更高的采样精度（支持8bit、16bit、24bit、32bit，音频CD为 16bit）以及有多种语言的兼容能力，更高的解码效率，一般来说，AAC可以在对比MP3文件缩小30%的前题下提供更好的音质。

* 相对于传统的LC-AAC，High Efficiency AAC（HE-AAC或写为 AAC-HE）又称为 "aacPlus v1" 或 "AAC+" - 结合了 SBR (Spectral Band Replication) and AAC技术；适用于低比特率（64kbps以下）；
* HE-AAC v2，又称为 "aacPlus v2" - 结合了 Parametric Stereo(参数化立体 声，PS)和 HE-AAC 中的SBR技术。

### AAC分为 9 种规格，

因为"AAC"是一个大家族，以适应不同场合的需要，也正是由于 AAC 的规格（Profile）繁多，导致普通电脑用户感觉十分困扰：

1. MPEG-2 AAC LC 低复杂度规格（Low Complexity）
2. MPEG-2 AAC Main 主规格
3. MPEG-2 AAC SSR 可变采样率规格（Scaleable Sample Rate）
4. MPEG-4 AAC LC 低复杂度规格（Low Complexity），现在的手机比较常见的 MP4 文件中的音频部份就包括了该规格音频文件
5. MPEG-4 AAC Main 主规格
6. MPEG-4 AAC SSR 可变采样率规格（Scaleable Sample Rate）
7. MPEG-4 AAC LTP 长时期预测规格（Long Term Predicition）
8. MPEG-4 AAC LD 低延迟规格（Low Delay）
9. MPEG-4 AAC HE 高效率规格（High Efficiency）

　　上述的规格中，主规格（Main）包含了除增益控制之外的全部功能，其音质最好，而低复杂度规格（LC）则是比较简单，没有了增益控制，但提高了 编码效率，至‘SSR’对‘LC’规格大体是相同，但是多了增益的控制功能，另外，MPEG-4 AAC/LTP/LD/HE，都是用在低比特率下编码，特别是‘HE’是有 Nero ACC 编码器支持，是近来常用的一种编码器，不过通常来说，Main 规格和 LC 规格的音质相差不大，因此目前使用最多的 AAC 规格多数是‘LC’规格，因为要考虑手机目前的存储器能力未达合理水准。

## [编解码学习笔记（六）：H.26x系列](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://javasogo.iteye.com%2Fblog%2F1015207)

部分资料来源与wiki以及[http://www.365pr.net/tech\_view.asp?id=315](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://www.365pr.net%2Ftech_view.asp%3Fid%3D315)。

### 简介

H.26x有H.261,H.262，H.263, H.263v2以及H.264，H.261基本上已经不再使用。其中H.262和H.264已经在MPEG系列中介绍，他们分别对应MPEG2的第2部和MPEG-4的第10部。不在整理这方面的资料。

### H.261

H.261其速率为64kbps的整数倍（1～30倍）。它最初是 针对在ISDN(综合业务数字网,Integrated Services Digital Network)上双向声像业务(特别是可视电话、视频会议)而设计的。

H.261是最早的运动图像压缩标准，它只对CIF和QCIF两 种图像格式进行处理，每帧图像分成图像层、宏块组(GOB)层、宏块(MB)层、块(Block)层来处理；并详细制定了视频编码的各个部分，包括运动补偿的帧间预测、DCT(离散余弦变换)、量化、熵编码，以及与固定速率的信道相适配的速率控制等部分。实际的编码算法类似于MPEG算法，但不能与后者兼 容。H.261在实时编码时比MPEG所占用的CPU运算量少得多，此算法为了优化带宽占用量，引进了在图像质量与运动幅度之间的平衡折衷机制。也就是说，剧烈运动的图像比相对静止的图像质量要差。因此这种方法是属于恒定码流可变质量编码。

H.261是第一个实用的数字视频编码标准。H.261的设计相当成功，之后的视频编码国际标准基本上都是基于 H.261相同的设计框架，包括 MPEG-1，MPEG-2／H.262，H.263，甚至 H.264。同样，H.261开发委员会（由Sakae Okubo领导，他的日文姓名是大久保荣）的基本的运作方式也被之后的视频编码标准开发组织所继承。H.261使用了混合编码框架，包括了基于运动补偿的 帧间预测，基于离散余弦变换的空域变换编码，量化，zig-zag扫描和熵编码。

实际上H.261标准仅仅规定了如何进行视频的解码（后继的各个视频编码标准也继承了这种做法）。这样的话，实际上开发者在编码器的设计上拥有相当的自由来设计编码算法，只要他们的编码器产生的码流能够被所有按照H.261规范制造的解码器解码就可以了。编码器可以按照自己的需要对输入的视频进行任何预处理，解码器也有自由对输出的视频在显示之前进行任何后处理。去块效应滤波器是一个有效的后处理技术，它能明显的减轻因为使用分块运动补偿编码造成的 块效应（马赛克）--在观看低码率视频（例如网站上的视频新闻）的时候我们都会注意到这种讨厌的效应。因此，在之后的视频编码标准如H.264中就把去块 效应滤波器加为标准的一部分（即使在使用H.264 的时候，再完成解码后再增加一个标准外的去块效应滤波器也能提高主观视频质量）。

后来的视频编码标准都可以说是在H.261的基础上进行逐步改进，引入新功能得到的。现在的视频编码标准比起H.261 来在各性能方面都有了很大的提高，这使得H.261成为了过时的标准，除了在一些视频会议系统和网络视频中为了向后兼容还支持H.261，已经基本上看不到使用H.261的产品了。 但是这并不妨碍H.261成为视频编码领域一个重要的里程碑式的标准。

### H.263

H.263最初设计为基于H.324的系统进行传输 （即基于公共交换电话网和其它基于电路交换的网络进行视频会议和视频电话）。后来发现H.263也可以成功的应用与H.323（基于RTP／IP网络的视 频会议系统），H.320（基于综合业务数字网的视频会议系统），RTSP（流式媒体传输系统）和SIP（基于因特网的视频会议）。

基于之前的视频编码国际标准（H.261，MPEG-1和H.262／MPEG-2），H.263的性能有了革命性的提高。它的第一版于1995年 完 成，在所有码率下都优于之前的H.261。 之后还有在1998 年增加了新的功能的第二版H.263+，或者叫H.263v2，以及在2000年完 成的第三版H.263++，即H.263v3。

H.263v2（通常也叫做H.263+或者1998年版H.263）是ITU-TH.263 视频编 码标准第二版的非正式名称。 它保持了原先版本H.263的所有技术，但是通过增加了几个附录显著的提高了编码效率并提供了其它的一些能力，例如增强了抵抗传输信道的数据丢失的能力（Robustness）。H.263+ 项目于1997年底／1998年初完成（这取决于我们怎么定义"完成"）。

H.263v3：接下来一个被称为"H.263++" 的项目被随即推出，在H.263+的基础上增加了更多的新的功能。H.263++于2000年底完成。增加了下面的附录：

* Annex A - Inverse transform accuracy specification
* Annex B - Hypothetical Reference Decoder
* Annex C - Considerations for Multipoint
* Annex D - Unrestricted Motion Vector mode
* Annex E - Syntax-based Arithmetic Coding mode
* Annex F - Advanced Prediction mode
* Annex G - PB-frames mode
* Annex H - Forward Error Correction for coded video signal

在H.263之后，ITU-T（在与MPEG的合作下）的下一代视频编解码器是H.264，或者叫AVC以及MPEG- 4第 10部分。由于H.264在性能上超越了H.263很多，现在通常认为H.263是一个过时的标准（虽然它的开发完成并不是很久以前的事情）。大多数新的 视频会议产品都已经支持了H.264视频编解码器，就像以前支持H.263和H.261一样。

话虽然如此，H.263在3GPP中仍然占有很高的地位，后继修订的版本，包括运营商的标准一直保留着H.263， 作为必选的要求，地位远远高于H.264，这是个奇怪的现象。一个重要的可能原因是H.263的编码比H.264的要轻载，在手机的modem中提供 H.263的编解码能力，不提供H.264的编解码能力，或者只提供H.264的解码能力不提供编码能力，如果不是智能手机不能在主板的其他芯片（例如 CPU）提供H.264的编解码能力，开发者就没什么办法，H.263可以通过软件来提供，H.264对处理能力的要求很高，目前需要依赖硬件能力提供。因此H.263仍然具有很大的市场，尤其对于小尺寸的手持设备，屏幕分辨率有限，高清无意义。

### H.264

H.264等同于MPEG-4的第10部，在这里仍然收集资料进行学习记录。

在H.263之后，ITU-T（在与MPEG的合作下）的下一代视 频编解码器是H.264，或者叫AVC以 及MPEG-4第 10部分。由于H.264在性能上超越了H.263很多，现在通常认为H.263是一个过时的标准（虽然它的开发完成并不是很久以前的事情）。大多数新的 视频会议产品都已经支持了H.264视频编解码器，就像以前支持H.263和H.261一样。

H.264／AVC可工作于多种速率，广泛应用于Internet／intranet上的多媒体流服务、视频点播、可视游戏、低码率移动多媒体通信 (视频 手机等)、交互式多媒体应用、实时多媒体监控、数字电视与演播电视和虚拟视频会议等，大有在上述领域一统天下的趋势，有非常广泛的开发和应用前景。

H.264是一种视频高压缩技术，同时称为MPEG-4 AVC，或MPEG-4 Part10。ITU-T从1998年就H.26L的H.26S两个分组，H.26L研制节目时间较长的高压缩编码技术，H.26S则指短节目标准制订部 门。前面的H.263就是H.26S标准化技术，而H.264标准是在H.26L基础上发展而来的。为了不引起误解，ITU-T推荐使用H.264作为这一标准的正式名称。H.264集中体现了当今国际视频编码解码技术的最新成果。在相同的重建图像质量下，H.264比其他视频压缩编码具有更高的压缩比、 更好的IP和无线网络信道适应性。

首先，H.264具备超高压缩率，其压缩率为MPEG-2的2 倍，MPEG-4的1.5倍。这样的高压缩率是以编码的大运算量来换取的，H.264的编码处理计算量有MPEG-2的十多倍。不过其解码的运算量并没有上升很多。从CPU频率和内存的高速发展的角度来看，1995年推出MPEG-2时，主流的CPU是奔腾100，内存更是小的可怜。而如今主流CPU的工 作频率比那个时候快了30倍，内存扩大了50多倍。所以H.264编码的大运算现在也不算什么大问题了。

高压缩率使图像的数据量减少，给存储和传输带来了方便。加上基本规格公开的国际标准和公正的许可制度，所以，电视广播、家电和通信三大行业都进入到 H.264的实际运用研发中心。美国高等电视系统会议和日本无线电工业和事务协会都准备把H.264作为地面便携式数字电视广播的编码方式。欧洲数字电视广播标准化团体也正在将H.264作为数字电视的一种编码方式来采用。

家电行业中的视频存储设备厂商也看中了H.264。东芝和NEC推出的下一代采用蓝色激光的光碟HD DVD-ROM，因为容量小于Sony等九大公司的蓝光碟，故将视频压缩编码改用H.264，从而使最终的节目录制时长能与蓝光碟相近。H.264也能使 HDTV节目录像和 SDTV的长时间录像成为可能。因而，生产LSI芯片的厂商也十分重视H.264。D9型DVD碟只有8.5GB，不足以存放2小时的HDTV节目，如用 H.264来压缩就变得有可能。同时，在通讯领域，互联网工程任务已开始将H.264作为实时传输协议流的格式进行标准化。互联网和手机的视频传送也会有 H.264作为编码方式。

相对于MPEG压缩编码H.264的变化之一是在帧内编码I画面中，又加入了帧内预测编码技术，即解码时可用周围数据的差分值来重构画面。在运动预 测块中，H.264采用全面运动预测和I画面帧内预测后，编码量得到减少，但LSI的运算处理量增大。为此，H.264引入了DCT的简化处理技术，来减 轻LSI的负担，画质也有所改善。H.264与MPEG-2和MPEG-4的不同还存在于熵编码块中，H.264的熵编码CAVLC（内容自适应可变长度 码）和CABAC（内容自适应二进制算法编码）能提高纠错能力。而MPEG-2和MPEG-4是霍夫曼编码。另外，还加入了解锁滤波器 （Deblocking Filter），有降低噪声的效果。H.264的整数变换以4×4像素块为单位，已比原来的8×8像素块的块噪声少，再次降低，画质得到了进一步提高。

H.264标准分为三档：基本档次；主要档次（可用于SDTV、HDTV和DVD等）；以及扩展档次（用于网络的视频流）。其中H.264的基本档 次是免费，用户可以无偿使用，现得到美国苹果公司和美国Cisco系统公司、中国联想公司、诺基亚、美国On2技术公司、德国西门子、TI公司等的支持； 其许可体系要比MPEG-4单纯，公正无差别对待用户和专利持有者。H.264替代MPEG-4的呼声很高，除了其高性能外，低额专利费和公正的无差别许 可制度也至关重要。由于技术的日益成熟，半导体厂商已在进行H.264的编码/解码LSI的开发。特别是HDD录像机和DVD录像机等设备中，采用 H.264的实例已很多，更引起了半导体厂商的关心。加之，H.264采用的动画编码方式和音频编码方式具有多样化特性，今后几乎将会是全部厂商的主要规格之一。

### 编码效率比较

|  |  |  |  |
| --- | --- | --- | --- |
| Codec | MPEG-4 | H.263 | MPEG-2 |
| H.264 | 39% | 49% | 64% |
| MPEG-4 | -- | 17% | 43% |
| H.263 | -- | -- | 31% |

## [编解码学习笔记（七）：微软Windows Media系列](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://javasogo.iteye.com%2Fblog%2F1015205)

资料来源于wiki与 [http://chaoqunz.blog.163.com/blog/static/6154877720084493941186/](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://chaoqunz.blog.163.com%2Fblog%2Fstatic%2F6154877720084493941186%2F)。

Microsoft 公司主导的音频视频编码系列，它的出现主要是为了进行网络视频传输，现在已经向HDTV 方面进军，开发了 WMV HD 应用。WMV（Windows Media Video）是微软公司开发的一组数字视频编 解码格式的通称，它是Windows Media架构下的一部分。它最初是为低速率流媒体应用作为专有编解码开发出来的，但是2003年微软公司基于Windows Media Video第9版编解码起草了视频编解码规范并且提交给SMPTE申请作为标准。这个标准在2006年3月作为SMPTE 421M被正式批准，这样Windows Media Video 9编解码就不再是一个专有的技术。早期的变解码版本（7和8）仍然被认为是专有技术，因为它们不在SMPTE 421M标准的涵盖范围内。

微软媒体系列分为WMV（Windows Media Video）和WMA（Windows Media Audio），说白了就是微软的视频和音频。

**容器**

视频流通常与Windows Media Audio音频流组合在一起并且使用扩展名为.wmv或者.asf的Advanced Streaming Format的文件进行封装。WMV通常使用Advanced StreamingFormat（ASF） 封装，它也可以使用AVI或者Matroska格 式封装。如果是AVI封装的文件结果文件可以是.avi，如果是ASF封装的话则是.wmv或者.asf， 如果是MKV封装的话则是.mkv。当使用VirtualDub编 码器编码和WMV9 VCM编解码实现的时候WMV可以存储在AVI文件中。用于Mac的微软公司媒体播放器不支持所有的WMV编码的文 件，因为它只支持ASF文件 封装，Flip4Mac和QuickTime或 者用于MacOSX的MPlayer可 以播放更多的文件。

**WMV**

扩展名： .wmv   
互联网媒体类型： video/x-ms-wmv   
统一类型标识： com.microsoft.windows-?media-wmv   
开发者： 微软公司   
格式： 数字视频

WMV（Windows Media Video）是微软公司开发的一组数字视频编解码格式的通称，它是Windows Media架构下的一部分。它最初是为低速率流媒体应用作为专有编解码开发出来的，但是2003年微软公司基于Windows Media Video第9版编解码起草了视频编解码规范并且提交给SMPTE申请作为标准。这个标准在2006年3月作为SMPTE 421M被正式批准，这样Windows Media Video 9编解码就不再是一个专有的技术。早期的变解码版本（7和8）仍然被认为是专有技术，因为它们不在SMPTE 421M标准的涵盖范围内。

WMV不是仅仅基于微软公司的自有技术开发的。从第七版（WMV1）开始，微软公司开始使用它自己非标准MPEG- 4 Part 2。但是，由于WMV第九版已经是SMPTE的一个独立标准（421M，也称为VC- 1），有理由相信WMV的发展已经不像之前那样是一个它自己专有的编解码技术。现在VC-1专利共享的企业有16家（2006年4月），微软公司也是 MPEG-4 AVC/H.264专利共享企业中的一家。

|  |  |  |  |
| --- | --- | --- | --- |
| **正式名称** | [**FourCC**](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://javasogo.iteye.com%2Fblog%2Fzh-cn%2FFourCC) | **Codec版本** | **描述** |
| Windows Media Video v7 | WMV1 | 0 |  |
| Microsoft [MPEG-4](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://javasogo.iteye.com%2Fblog%2Fzh-cn%2FMPEG-4)Video Codec v3 | MP43 | 1 |  |
| Windows Media Video v8 | WMV2 | 2 |  |
| Microsoft [MPEG-4](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://javasogo.iteye.com%2Fblog%2Fzh-cn%2FMPEG-4)Video Codec v2 | MP42 | 3 |  |
| Microsoft ISO [MPEG-4](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://javasogo.iteye.com%2Fblog%2Fzh-cn%2FMPEG-4)Video Codec v1 | MP4S | 4 |  |
| Windows Media Video v9 | WMV3 | 5 |  |
| Windows Media Video v9 Advanced Profile | WMVA | 6 | deprecated as not [VC-1](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://javasogo.iteye.com%2Fblog%2Fzh-cn%2FVC-1)不完全兼容。 |
| Windows Media Video v9 Advanced Profile | WVC1 | 7 | [VC-1](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://javasogo.iteye.com%2Fblog%2Fzh-cn%2FVC-1)完整支持 |

FourCC全称Four-Character Codes，是由4个字符（4 bytes）组成，是一种独立标示视频数据流格式的四字节，在wav、avi档案之中会有一段FourCC来描述这个AVI档案，是利用何种codec来 编码的。因此wav、avi大量存在等于“IDP3”的FourCC。

**Microsoft MPEG-4 v1/v2/v3**

常见的有1.0、2.0、3.0三种版本，是基于MPEG-4技术的，其中3.0并不能用于AVI的编码，只能用于生成支持“视频流”技术的ASF文件。

**Microsoft MPEG-4 version 1**  
是微软基础的视频编码，是非标准的MPEG-4，与MPEG-4 Part2不兼容。FourCC:MPG4

**Microsoft MPEG-4 version 2**  
VFW的基础编解码。与V与MPEG-4Part2不兼容。VFW（Video for Windows），是微软推出的关于数字视频的一个软件开发包，VFW的核心是AVI文件标准。AVI(AudioVideo Interleave)文件中的音、视频数据帧交错存放。围绕AVI文件，VFW推出了一整套完整的视频采集、压缩、解压缩、回放和编辑的应用程序接口 (API)。由于AVI文件格式推出较早且在数字视频技术中有广泛的应用，所以VFW仍然有很大的实用价值，而且进一步发展的趋势。在VC++开发环境中 调用VFW和使用其它开发包没有什么不同，只是需要将VFW32.lib 文件加入工程中，但在开放视频捕捉与压缩管理程序时需要其它软件硬件设置。VFW为AVI文件提供了丰富的处理函数和宏定义，AVI文件的特点在于它是典 型的数据流文件，它由视频流、音频流、文本流组成。所以对AVI文件的处理主要是处理文件流。FourCC：MP42

**Microsoft MPEG-4 version 3**  
VFW的基础编解码。与V与MPEG-4Part2不兼容。最终只用于ASF文件。FourCC：MP43

除此之外：   
**Microsoft ISO MPEG-4 version 1**  
基于DirectX Media Objects (DMO)-based编解码，与MPEG-4 SP（Simple Profile）兼容 。FourCC：MP4S。

**Microsoft ISO MPEG-4 version 1.1**  
与MPEG-4 ASP（AdvancedSimple Profile）兼容。FourCC：M4S2

目前在Windows平台上比较实用的MPEG-4视频编解码器主要有：由微软公司开发的Microsoft MPEG-4 Codec v1/v2 /v3，主要是配合微软公司的流媒体技术使用；在Microsoft MPEG-4 v3的基础上，由DivXNetworks公司开发的 DivX Codec；在OpenDivX的基础，遵循GPL开发的开源XviD Codec。  
这些codec在windows中以dll方式呈现。

**Windows Media Video 7**

是DirectXMedia Objects (DMO)-based codec。Microsoft 正式开发的 第一个 Windows Media Video，开始脱离了MPEG-4，和MPEG-4 不兼容，从这一点上可见微软的野心（微软公司开始使用它自己非标准MPEG-4Part 2）。可惜这个版本压缩效果非常烂，打破了微软一飞冲天的美梦，不过它在压缩速度上非常快，现在网络上有很多采用这种格式压缩的 WMV。FourCC：WMV1

**Windows Media Video ８**

在WMV7 基 础上改进的版本，质量上面进不了不少。是DMO-based codec。FourCC：WMV2.

**Windows Media Video 9**

微软的重头戏，不光是这一个编码，V9 系列更是一个平台，让微软有足够的能力挑战 MPEG，ITU 等标准化组织。虽然这个版本并没有微软吹得那么厉害，特别 是低码率下比较差，不过跟以前版本相比进步还是非常多的。特别是 WMV HD 的应用，让微软也跻身视频标准领域。  
DMO-based codec. Video for Windows (VfW/VCM) version also available. FourCC：WMV3

**Windows Media Video 9 Advanced Profile**

在WMV4的Simple和Main profile levels与VC-1标准中的是相同的。VC-1中的AdvancedProfile采用一个新的WMV编解码，称为WindowsMedia Video 9 Advanced Profile。它提高隔行扫描的内容压缩，并与传输无关，使得他可以封装在MPEG的传输流（TS）中或者采用RTP传输。它与之前的WMV9的 codec是不兼容的。

随着 Windows Media Player 10 推出的编码器，能够更进一步控制WMV9 的质量。但是不能在老版本的 WMP9 上播放，也就是不兼容老版本的 WMP9，真不知微软在搞什么？

FourCC：WVC1，VC-1兼容。FourCC：WMVA，不推荐使用非VC-1兼容方式。我们可以认为WMV9与VC-1兼容。

**Windows Media Video 9 Screen**

静态屏幕无损压缩编码，质量非常好，压缩率高，只针对如屏幕等变化非常小的环境。WMV Screen是一个screencast编解码器，它可以捕抓动态的屏幕内容，将第三方抓屏程序转换到WMV9 Screen文件中。用于之一是计算机的step-by\*step演示视频，第一版本是WMV7 Screen，第二个版本也就是当前的版本是WMV9 Screen，支持CBR和VBR。

**Windows Media Video 9 Image**

静态图像压缩编码。WMV Image是一个视频幻灯片的编码器，播放多个画面时可根据时间进行平移和过度效果展现。与WMV9相比，具有高压缩率和高态图像质量。由于编解码依赖解 码器（播放器）生成实施视频帧，播放WMV Image文件（即使在普通的1024×768，30fps）要求很高的处理能力。在最新的版本，WMV9.1 Image，使用Photo Story3提高转换效果，耽于原来的WMV9 Image不兼容。

**视频质量**

微软宣称WMV9提供的压缩率为MPEG-4的2倍，MPEG-2的3倍。微软还宣称WMV9的压缩效率笔WMV8高15%～50%。但是在2005年的一个测试报告中，显示WMV9的压缩效率比WMV8要差。

Windows Media Player 10 Mobile

在wiki上，我们注意到“Windows Media Player 10 Mobile”，显示WMV10将用于移动，可能是Windows Mobile。但是我们没有查到进一步的资料。

**WMA**

扩展名 .wma   
互联网媒体类型 audio/x-ms-wma   
统一类型标识 com.microsoft.windows-?media-wma   
开发者 微软   
格式 数字音频

WMA（Windows Media Audio）是微软公司开发的一种数字音频压缩格式。一些使用Windows Media Audio编码格式编码其所有内容的纯音频ASF文件也使用WMA作为扩展名。WMA格式为微软公司私有，但是随着苹果公司的iTunes对它的支持，这 个格式正在成为MP3格式的竞争对手。它兼容MP3的ID3元数据标签，同时支持额外的标签。

WMA可以用于多种格式的编码文件中。应用程序可以使用Windows MediaFormat SDK进行WMA格式的编码和解码。一些常见的支持WMA的应用程序包括Windows Media Player、Windows Media Encoder、RealPlayer、Winamp等等。其它一些平台，例如Linux和移动设备中的软硬件也支持此格式。

WMA 7之后的WMA支持证书加密，未经许可（即未获得许可证书），即使是非法拷贝到本地，也是无法收听的。同时，微软公司开始时宣称的：同文件比MP3体积小 一倍而音质不变，这也得到了兑现。另外，微软公司在WMA 9大幅改进了其引擎，实际上几乎可以在同文件同音质下比MP3体积少1/3左右，因此非常适合用于网络流媒体。

跟MP3相比，WMA在高位的音质渲染力明显不足，甚至还比MP3更差;与MP3相同，通常的WMA也是有损数据压缩的文件格式，对于有更高要求的用户来说WMA并不是一个适合的格式。但在WMA9版本开始支持无损压缩——Windows Media Audio 9 Lossless（在安装WMP11或Windows Media Format 11之后升级至9.1，无损压缩版本最高支持5.1声道编码）。此外WMA也与MP3一样同为有专利版权的文件格式。支持的设备需要购买使用版权。

**Windows Media Audio v1/v2**

微软最早的音频编码技术，用于 ASF 中，后来被破解也用在 DivX Audio 中，质量比较差。

**Windows Media Audio 7/8/9**

随着各种不同的 WMV 而推出的相应的音频编码，质量节节提升，不过还没有达到 64kbps CD音质的神化。

**Windows Media Audio 9 Professional**

WMA9 中 出现的新编码，主要用于多声道编码和高采样率音频的编码，质量不错。

**Windows Media Audio 9 Voice**

针对语音的编码，最高 20kbps ，不过和 AMR 相比，效果就太差了。

**Windows Media Audio 9 Lossless**

无损音频编码，可以完美保留CD原质量，是CD备份的不错选择，不过代价是体积过大。

**VC-1**

VC-1，全名VC-1视讯编解码器（Video Codec 1），是基于微软WMV9，并推广为工业标准。2003年提出标准化申请，最早名字是VC-9。2006年4月正式通过成为标准。VC-1是SMPTE 421M视频编解码标准的正式名称。HD DVD 和蓝光光碟（Blu-ray Disc）都支援VC-1。微软表示Windows Vista将支援VC-1影像压缩技术的HD DVD规格。电影及电视学会（SMPTE）已采用VC-1为视讯压缩标准。

VC-1是基于Windows Media Video 9压缩技术的影像压缩标准，由三大编解码元件所组成，每一个编解码元件都具有其独自的FourCC编码。

**WMV3** ：

即WMV9。VC-1的Simple和Main这两种Profile就是WMV3应用，使得与WMV 9兼容，支持逐行扫描编解码。隔行扫描的编解码也提供，但在很快地，在微软推出WMVAdvanced profile后，不推荐采用。逐行扫描编解码用于YUV4:2:0，隔行的（不推荐）用于YUV4:1:1。

WMV3用于高质量的视频和流媒体。同样的质量，它只是MPEG-2的带宽的1/2～1/3。在商业上用于高清电影和视频的WMV HD，编码为WMV3 Main Profile @ High Level(MP@HL)。

**WMVA** ：

它是在WMV Advanced Profile被SMPTE吸收为作为VC-1草案之间出现的。它与WVC1之间细微的差役，因此解码器也不一样，2006年起，WMVA被认为是个过时的编码，因为与VC-1不完全兼容。

**WVC1** ：

也就是WMV 9 Advanced Profile，实现了个更新的，完全符合的AdvancedProfile的VC-1编码标准。它支持隔行扫描内容，与底层传输无关。

压缩技术整合了MPEG及H.264之优点，采用Biliner和Bicubic方式，次像素（Sub- Pixel）最小可达4分之1像素。VC-1只有4种动作补偿（motion composition），压缩比无法胜过H.264。VC-1在压缩时间上，明显比H.264短了许多，复杂度约只有H.264的50%，对特效电影有 很杰出的效能表现。由于H.264 使用较小尺寸的转换公式与无法调整的量化矩阵，造成不能完整保留影像的高频细节资讯。

wiki上有一篇 [VC-1和H.264比较](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://zh.wikipedia.org%2Fzh-cn%2FH.264%25E8%2588%2587VC-1%25E7%259A%2584%25E6%25AF%2594%25E8%25BC%2583)的文章，很值得看一看。我看到这样的一个段：

VC-1：需付费。Reference decoder 并非免费，不过附赠外部文件

H.264：免费.Reference encoder 和 decoder 也是免费。此外，核查小组及M4IF邮寄名单可在一个可能会收到回答了AVC的有关问题。

除此之外，我们在Google上search H.264 license中也看到了说free的字样。但是，果真如此吗？

## 版权问题

我一直认为H.264是需要付费的，因此我对上面的表述不确定是否正确。我上网查了一下：也看到“H.264的基本系统无需使用版权 ，具有开放的性质，能很好地适应IP和无线网络的使用，”的字样。基于这些疑问，我上网查了一下。IP总是个麻烦的问题，最好向Andriod那样，提供平台，版权，产品维护都是手机生产厂家的事情。

MPEG LA公司是世界领先的替代技术特许服务提供商，它使用户能够通过单笔交易从多个专利持有人那里购买某项技术标准或平台所必须的全球专利权，而不必分别洽谈各项特许权。只要独立管理的一站式专利特许权能够开启方便之门、帮助用户推广某项技术， MPEG LA公司率先采用的特许模式就能提供解决方案。MPEG LA公司的特许权之一就是MPEG-2数字视频压缩，这一技术帮助产生了消费电子历史上应用最为广泛的标准。包括57个国家逾870个MPEG-2基本专利的MPEG-2专利组合特许权拥有至少1500家被许可人，覆盖目前全球市场上的大多数MPEG-2产品，包括机顶盒、DVD播放器、数字电视机、个人 电脑和DVD视频光盘。作为独立特许经营管理人，MPEG LA公司不与任何标准管理机构相关联，也不是任何专利权持有人的关联人。欲了解更多信息，请登录http://www.mpegla.com。 （[http://www.dvbcn.com/2010-01/28-44547.html](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://www.dvbcn.com%2F2010-01%2F28-44547.html)）

## [编解码学习笔记（八）：Real系列](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://javasogo.iteye.com%2Fblog%2F1015204)

下面资料从wiki中整理获取。

Real系列是RealNetworks提供的，分为RealVideo和RealAudio。

**RealVideo**

RealVideo是一种影片格式由RealNetworks于 1997年所开发，至2006年时已到RealVideo版本10。它从开发伊始就定位为应用为网络上视频播放上的格式。支持多种播放的平台，包含 Windows、Mac、Linux、Solaris以及某些移动电话。相较于其它的视频编解码器，RealVideo通常可以将视频数据压缩得更小。因此它可以在用56Kbps MODEM拨号上网的条件实现不间断的视频播放。

一般的文件扩展名为.rm/.rvm，现在广泛流行的是rmvb格式，即动态编码率的real video。

RealVideo早期使用H.263，在RealVideo8及之后公司采用私有或者不公开的视频格式。官方的的播放器是 RealNetworks RealPlayer SP，最新的版本是v12，在多个平台，包括Windows，Macintosh，和Linux中提供。RealNetworks公司同时也开发了开源的 Helix播放器，但是在Helix项目中没有提供RealVideo，因为Real系列的编解码仍是不公开的。

RealMedia文件可以通过RTSP在网络上传输，但是RTSP只用于建立和管理连接，真正的视频数据是通过 Real自己私有的 RDT（Real Data Transport）协议传输。这种方式引起很大的批评，因为难以在其他播放器和服务器中使用RealVideo，现在一些开源项目，例如MPlayer 已经可以播放RDT流。为了支持实时流，RealVideo和RealAudio通常采用CBR（恒定码速）编码，使得每秒传递的数据相等。后来，公司开发了可变码速，成为RealMedia Variable Bitrate（RMVB），提供更高的水平资料，但是这种格式不适合作为流，因为难以预测某个特定的媒体流需要多少网络资源。带有快速移动和场景变化的视频需要更高的码流，如果码流超出网络所能提供的速率，将会引起中断。

RealNetworks公司说RealVideo和RealAudio编解码的源代码不在RPSL的许可内。在许可源代码在不支持的处理器和操作系统的商用移植中许可。虽然公司拥有大多数的知识产权，但是允许第三方为某个特性享有版权。

**RealVideo 1.0**

RealVideo的第一个版本在1997年释放，是基于H.263格式。在RealPayer5中提供。FourCC为rv10，rv13

**RealVideo G2和RealVideoG2+SVT**

也是就要H.263，在RealPlayer6中提供。质量比较糟糕。FourCC为rv20

**RealVideo 8**

随着 RealPlayer 8 推出的视频格式，是现在主流的网络视频编码之一。编码速度较慢，质量也只能算一般。猜测是基于早期的H.264 draft，在RealPlayer 8中提供，FourCC为rv30。

**RealVideo 9**

RealNetworks 开 发的新一代编码，质量进步了很多，特别是在底码率下，而且编码速度很快，做到了速与质的很好统一。  
猜测是基于H.264，在RealPlayer9中提供。FourCC为rv40.

**RealVideo 10**

在RealVideo 9 基 础上加入了一些参数，如 EHQ 等，更加精确控制码率，和 RealVideo 9 兼容。在RealPlayer10中提供，FourCC为rv40（与RealVideo9相同）

**RealAudio**

文件名扩展： .ra, .ram

Internet media type：audio/vnd.rn-realaudio,audio/x-pn-realaudio

RealAUdio是RealNetwoks公司私有的宾朋编解码格式，最早在1995年释放。它包括一系列的音频编解码，从古老拨号modem的 低速率格式到高质量的音乐。可用于媒体流，过去很多的互联网电台使用RealAudio作为他们节目的实时音频流，近年使用得较少，让位与其他流行的格 式。

RealAudio文件后缀为.ra。1997年，公司开始提供视频格式成为RealVideo。将音频和视频合并是以哦那个的容器后缀为.rm。然而，最新的版本，使用.ra用于音频文件，使用.rv用于带或不带音频的视频文件，用.rmvb用于可变速率的视频文件。

.ram(Real Audio Metadata)和.smil(SynchronizedMultimedia Integration Language)文件格式用于流媒体中的链接。在很多情况下，网络并不直接链接一个RealAudio文件，而是链接.ram和.smil文件。这是一 个很小的文本文件，包括音频流的链接。当用户点击一个链接，网络浏览器下载.ram和.smil文件，并加载用户的媒体播放器，媒体播放器从文件中读取 pnm或者rtsp的URL，并开始播放流。

RealAudio文件包括多种音频编解码，每个编解码都是用FourCC（Four Character Code）来表示，介绍如下：

* lpcJ: IS-54 VSELP (RealAudio 1)
* 28\_8: G.728 LD-CELP (RealAudio 2)
* dnet: Dolby AC3 (RealAudio 3)
* sipr: Sipro Lab Telecom ACELP-NET (RealAudio 4/5)
* cook: G2/Cook Codec (RealAudio 6)
* atrc: Sony ATRAC3 (RealAudio 8)
* raac: MPEG-4 LC-AAC (RealAudio 9)
* racp: MPEG-4 HE-AAC (RealAudio 10)
* ralf: RealAudio Lossless Format (RealAudio 10)

## [编解码学习笔记（九）：QuickTime系列](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://javasogo.iteye.com%2Fblog%2F1015198" \t "_blank)

扩展名 .mov,.qt  
互联网媒体类型 video/quicktime  
类型代码 MooV  
统一类型标识 com.apple.quicktime-movie  
开发者 苹果公司  
格式 　媒体容器  
专门属 音频、视频、文字

　　或者标题可以改为Apple系列。QuickTime 并不是一个编码，而是一个多媒体平台，它的上面有众多编码，这里只针对几个主流的编码器。  
　　QuickTime技术拥有三种主要的组件：

* 媒体播放器，苹果计算机在他自己的网站免费让人下载以及内建在他的计算机中。
* QuickTime 档案格式-公开文件并且任何人都可以使用，不须权利金。
* 软件开发工具可用于Macintosh和Windows平台。这些工具允许人们开发他们自己的软件来操作QuickTime以及其它媒体档案。这些对已注册开发人员是 免费的（注册免费）。

　　苹果在Mac OS和Windows平台推出免费之官方媒体播放软件，名字为"QuickTime Player"（早期的版本简单地使用　　了"MoviePlayer"这个名称）。这个播放器也包含一些媒体编辑和媒体创作的特色，但是使用者必须从 Apple购买序列号来打开这些功能，把这个播放器转变为"QuickTimePro"。

**QuickTime历史：1991年到1998年：1.x- 2.x**

　　苹果电脑于1991年12月2日释出第一个QuickTime的版本，作为System 7上的多媒体附加功能。QuickTime的首席开发者，Bruce Leak，于1991年五月的Worldwide Developers Conference做了第一次的公开展示。他在Mac上展示苹果电脑有名的1984电视广告，在那时候是一种令人印象深刻的突破。Microsoft的 竞争技术-Video for Windows- 在1992年11 月之前都还未出现。

　　第一个版本的QuickTime制定的基本架构，到现在基本上还存在未更改，包含多重电影轨道，可扩充的媒体形态支持，一种开放的文件格式，以及完整的编辑功能。原本的视频codec包含：

* Apple视频codec（也称作"Road Pizza"），适合普通现场动作图像。
* 动画codec，使用简单的run-length图形压缩方式，适合卡通形态的大区域颜色很适合。
* 图形codec，对于每一点8位（8-bit-per-pixel）的图像优化，包含有抖色en:dithering的图形。

　　苹果电脑在1992年后期发放了Mac操作系统的1.5版本。

　　苹果电脑在1994年二月发布了QuickTime 2.0 for Mac OS版--这个是唯一的一个不免费的版本。在这个版本中加入了对音乐轨迹的支持，音乐轨迹相当于MIDI的数据，这个功能可以驱动sound- synthesis引擎自我创建于QuickTime中（使用的声音许可证来自Roland），或者是任何外部的MIDI设备，因此创建出来的声音只占用 一小部分的电影数据。

　　在接下来的2.1和2.5版本中，QuickTime继续免费。工程师改良了对音乐的支持并增加了sprite轨迹，这个功能可以实现创建复杂的动画，文件大小就只比静态的图片大一点。

　　QuickTime 2.0 for Windows发布于1994年11月。

**QuickTime历史：1998 - 2001: 3.0 and 4.0**

　　运行于Mac OS的QuickTime 3.0于1998年3月30日发行。其现有的功能是免费的，但如果要获得Apple所提供的具有更多特性的QuickTime Player和Picture Viewer程序，最终用户需要通过购买一个QuickTime Pro 许可证来解除对软件的限制。

　　QuickTime 3.0增加了支持图像导入的组件，从而可以从GIF、JPEG、TIFF和其他文件格式中读取图像。而通过FireWire主要作为视频数据输出的视频输 出组件同样增加了视觉效果，使程序员可以把实时技术运用到视频轨道中。一些效果甚至可以响应用户的鼠标单击，就像是电影本身的交互支持一样。

　　苹果于1999年6 月10日发行了QuickTime 4.0 for Mac OS。它增加了图像导出组件，支持输出成与预导入者可以阅读的相同格式的非GIF格式（或许是因为LZW许可）。它增加了Sorenson video codec的第一个版本，并且支持streaming。

　　QuickTime 4.1于2000年伊始发布，增加了在Mac OS 9及后续版本中播放超过2G的电影；并且终止了对68K Mac的支持。用户获得了操作QuickTime Player viaAppleScript的能力。

**QuickTime历史：2001 - present 5.0 and later**   
　　QuickTime 5.0 for Mac OS于2001年4月23日出现。它增加了"皮肤"功能和多处理图像压缩支持。在这一版本中只有拥有QuickTime Pro许可证的用户才可以使用全屏模式，这一做法引起了争议，至今尚未解决。

**QuickTime历史：QuickTime 6.x**  
　　QuickTime 6.0 for Mac OS，于2002年7月15日释出，第一次包括Mac OS X使用的版本。

|  |  |  |  |
| --- | --- | --- | --- |
| **更新至QuickTime 6** | | | |
| **释出日期** | **版本** | **平台** | **特色** |
| 2002 年 7 月15 日 | QuickTime 6 | Mac OS 8.6 –Mac OS X ,Windows | MPEG-2 、MPEG-4 以及AAC |
| 2003 年 1 月14 日 | QuickTime6.1 | Mac OS X | 品质与效能的增进 |
| 2003 年 3 月31 日 | QuickTime 6.1 | Windows | 修正CAN-2003-0168 安全性弱点 |
| 2003 年 4 月29 日 | QuickTime 6.2 | Mac OS X | 对于iTunes 4 的支持、增进的AAC 支持 |
| 2003 年 6 月3 日 | QuickTime 6.3 | Mac OS X, Windows | 3GPP 与AMR |
| 2003 年 10 月16 日 | QuickTime 6.4 | Mac OS X, Windows | Pixlet 编码解码器、集成3GPP |
| 2003 年 12 月18 日 | QuickTime 6.5 | Mac OS X, Windows | 3GPP2 与AMC 移动式多媒体格式 |
| 2004 年 4 月28 日 | QuickTime 6.5.1 | Mac OS X, Windows | Apple Lossless |
| 2004 年 10 月27 日 | QuickTime 6.5.2 | Mac OS X, Windows （对Windows 98/Me 的最后一个版本） | 修正程序错误、安全性更新、以及品质与效能的增进 |
| 2005 年 10 月12 日 | QuickTime 6.5.3 | Mac OS X v10.2.8 |  |

**QuickTime历史QuickTime 7.x**

　　更新至QuickTime 7。

|  |  |  |  |
| --- | --- | --- | --- |
| **更新至QuickTime 7** | | | |
| **释出日期** | **版本** | **平台** | **特色** |
| 2005 年 5 月31 日 | QuickTime 7.0.1 | Mac OS X | 修复一个关于Quartz Composer 外挂模块 的安全性问题 |
| 2005 年 7 月15 日 | QuickTime 7.0.2 | Mac OS X | 修复程序错误与兼容性的增进 |
| 2005 年 9 月7 日 | QuickTime 7.0.2 | Windows 2000 /XP | 第一个非预览版本释出 |
| 2005 年 10 月12 日 | QuickTime 7.0.3 | Mac OS X & Windows 2000 /XP | 流 与H.264 的程序错误修复。 需要通过iTunes Music Store 支持视频的购买。 |
| 2005 年 10 月29 日 | QuickTime 7.0.3.50 | Windows 2000 /XP |  |
| 2006 年 1 月10 日 | QuickTime 7.0.4 | Mac OS X & Windows 2000 /XP | 第一个通用二进制 版本。 众多的程序错误修复与H.264 效能的改善。 |
| 2006 年 5 月11 日 | QuickTime 7.1 | Mac OS X & Windows 2000 /XP | 众多的程序错误修复、对于iLife '06 的支持、以及H.264 效能的改善 |
| 2006 年 5 月31 日 | QuickTime 7.1.1 | Mac OS X |  |
| 2006 年 6 月28 日 | QuickTime 7.1.2 | Mac OS X | Addresses an issue previewing iDVD projects. |
| 2006 年 9 月12 日 | QuickTime 7.1.3 | Mac OS X & Windows 2000 /XP | 程序错误修复与严重的安全性问题。 |

　　现时QuickTime的最新版本是QuickTime 7.6，但Windows 2000可以使用的最高版本却是7.1.6。在7.5.5之前的版本，都存在Cross site scriping的安全性问题。  
**QuickTime的历史：QuickTime X**

　　QuickTime X（读作Quicktime Ten，当中 的"X"是罗马数字的十）是下一世代的QuickTime，在2008年6 月9日的WWDC上 发表。产品预期会在2009年的年中随同Mac OS X v10.6推出[3]。Version X会使用与iPhone OS相同的媒体技术，并支持更新的编码及更具效益的媒体播放功能。  
**Sorenson Video**

Sorenson Video 2：

Sorenson Media 公 司开发的编码器，主要用于QuickTime 4 的视频编码，质量较差。

Sorenson Video 3：

Sorenson Media 公 司随 QuickTime 5发布的编码器，质量很不错，已经成为 QuickTime 的标准视频编码，网络上大部分电影预告片都采用这种编码。

**Apple MPEG-4**

　　Apple 公司自己开发的 MPEG-4 编码器，随 QuickTime 6 发布，质量很差。

**Apple H.264**

　　Apple 公司自己开发的 H.264 编码器，随 QuickTime 7 发布，支持 HDTV。

**Audio QDesign Music**

QDesign Music 1

QDesign 公司开发的音频编码器，这个版本现在已经开不到它的身影了。

QDesign Music 2

QDesign Music 的 第二个版本，也是最后一个版本，在时下这些先进的音频编码面前，它已经没有生命力了，主要应用于网上的电影预告片。

**Audio Apple MPEG-4 AAC**

　　Apple 公司自己开发 的 AAC 编码器，质量非常好，是最优秀的 AAC 编码器之一，随 QuickTime 6 发布。

**Apple Lossless**

　　Apple 公司开 发的无损音频编码，主要应用于 iTunes 抓取 CD。Apple Lossless（AppleLossless Audio Codec、ALAC）为苹果的无损音频压缩编码格式。 在 iTunes 上名称为 Apple Lossless。

　　可将非压缩音频格式（WAV、AIFF）压缩至原先容量的40%至60%左右，编译码速度很快。也因为是无损压缩，听起来与原档案完全一样，不会因解压缩和压缩而改变。

　　它在2004年4月28日公布的iTunes4.5和QuickTime6.5.1 的其中一部份。目前携带型数字多媒体播放器中只有 iPod 可播放。

　　虽为非自由软件或开放原始码软件、但 Apple Lossless 的开放原始码译码器已经释出。

注：上述材料来源于wiki的整理。

## [编解码学习笔记（十）：Ogg系列](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://javasogo.iteye.com%2Fblog%2F1015197" \t "_blank)

Ogg是一个自由且开放标准的容器格式，由Xiph.Org 基金会所维护。Ogg格式并不受到软件专利的限制，并设计用于有效率地串流媒体和处理高质量的数字多媒体。

　　Ogg意指一种文件格式，可以纳入各式各样自由和开放源代码的编解码器，包含音效、视频、 文字（像字幕）与元数据的处理。

**OggTheora**

　　Theora是一个免权利金、开放格式的有损影像压缩技术，由Xiph.Org基金会开发，该基金会还开发了著名的声音编码技术Vorbis， 以及多媒体容器档案Ogg。Theora 是由 On2Technologies 公司专属的 VP3 编码器经过开放源代码后衍生而来。Theora 的命名来自于一个电视节目Max Headroom。

　　Theora是一个可 变位速率、以DCT为基础的影像压缩格式。和多数的影像编码格式一样，Theora 使用了色度抽样、block basedmotion compensation 和 8×8 DCT block，也支援视 讯压缩图像类型和 视 讯压缩图像类型，但是不支持使用在 H.264 和 VC-1 的 bi-predictive frames（B-frame），Theora 也不支援 隔 行扫描, variable frame rates, 或 bit-depths larger than 8 bits per component。

　　Theora 的影像流可以储存在任何的容器档案格式中，最常用的是和声音编码Vorbis一起储存在Ogg档案格式中，这种方式可以提供完全开放、免权利金的多媒体档案。此外Theora影像也可以储存在Matroska档 案中。

　　Google官方blog称，Web视频目前没有一个标准，有些网站使用Flash，但这要求用户有Flash播放 器;有些使用Java播放器，但为了在JVM虚拟机中解码视频和音频用户需要一个配置很高的机器;等等诸如此类。

　　好消息是新一代的 HTML 5标准引入了视频元素，Web开发者可以用一种标准的方式指定视频的外观。现在问题变成了使用哪一种视频格式。

　　Google 认为开放标准格式可以成为目前无序的视频格式之争的底线。最后的权益方案不需要最复杂的格式，或者是最大张旗鼓宣传，几乎已成为行业标准的格式，因此他们决定选择支持使用广泛的开源Ogg Theora格式。

　　Theora是On2 Technologies 公司的 VP3 编码器的开源衍生版，Google于去年收购了On2公司。

　　资料来源：[http://tech.it168.com/a2010/0412/872/000000872493.shtml](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://tech.it168.com%2Fa2010%2F0412%2F872%2F000000872493.shtml)

　　在2002年3月，On2将许可改为下周VP3开源代码为LGPL。在2002年6月，On2将VP3作为 Xiph.Org仅仅和下的一个类似BSD的开源许可。On2还制定了一个不可更改的免版权费的声明，任何人可以用于任何软件，任何的衍生产品以及任何目 的。2002年88月，On2与Xiph.Org签订一项协议，将VP3作为一个新的，免费的视频编解码，成为Theora。On2宣称Theora是 VP3的一个继任者。在2002年10月3日，On2和Xiph宣布Theora最早的Alpha代码释放。

　　比特流格式在2004年冻结（version 1.0 alpha3），有经过了几年的beta版本，Theora第一稳定版本（v1.0）在2008年11月释放。Theora任何版本的视频编码在格式冻结后都与未来的播放器兼容。目前的工作集中在“Thusnelda”分支的bug修复，目前为beta版本，将最后作为Theora1.1版本释放。

　　Theora视频压缩格式基本上与VP3视频压缩格式兼容，包含一个后先兼容的超集。Theora是VP3和 VP3流（有小量语法修订）的超集，VP3流可以不经过重新压缩改为Theora流，但是反过来把成立。VP3视频压缩可以由Theora来实现解码，但 是Theora视频演示通常不能够使用古老的VP3来进行解码。

　　Theora将视频格式建立在开源的基础上，并作为基百科视频内容选择的编码格式。然而，Theora缺少商业支持，并正努力获取分销商尤其是网络分销商接受。

　　Mozilla使用这项技术在Firefox上提供HTML5视频。苹果和微软的HTML5视频都准备采用 MPEG LA管理的H.264。该团体的成员包括微软和苹果，及许多科技公司。

　　这里面的争议关键在于license的问题，H.264是需要给license的。

　　Mozilla则发出以下声明：“我们相信，HTML5视频在多方、开放和无权利金的编/解码器，以同于W3C 授权标准之方式支持下，才符合公众利益。如 果MPGA LA愿意根据W3C标准定义的开放网络条件提供H.264，我们绝对会考虑采用这项技术。本组织坚持我们对Theora的立场。”

　　Opera首席技术官Hakon Wium Lie也提供下列声明；“为了开放网络的成长茁壮，所有媒体(包括视频)，必须在无需支付编/解码器授权费的前提下使用。真正支持一个开放网络的浏览器制造商，必须努力建造一个无授权费的基本影音编/解码器。”

　　微软的公司博客写道：“源代码能否取得，与知识产权之间的区别，在于可取用的源代码是极度必要的。目前，H.264的知识产权，可通过MPEG LA管理的一个定义明确的方案取得。其它编/解码器的权利通常较不清楚。”

**Ogg Vorbis**

Ogg 的音频编码， 质量非常优秀，特别是低码率下，支持多声道。最高码率能够达到500kbps，是 AAC 的有力竞争者。

　　「Ogg」这个词汇通常意指Ogg Vorbis此一音频文件格式，也就是将Vorbis编码的音效包含在Ogg的容器中所成的格式。在以往，.ogg此一扩展名曾经被用在任何Ogg支持格 式下的内容，但在2007年，Xiph.Org基金会为了向后兼容的考虑，提出请求，将.ogg只留给Vorbis格式来使用。Xiph.Org基金会决定创造一些新的扩展名和媒体格式来描述不同类型的内容，像是只包含音效所用的.oga，包含或不含声音的影片（涵盖 Theora）所用的.ogv和程序所用的.ogx。

　　Vorbis的是一个 开源自由软件 项目负责人是 Xiph.Org基金会。该项目产生的数位音频格式规范和软件实施（编译码器）为有损音频压缩。 Vorbi是最常用的结合与Ogg容器格式，因此通常被称为 Ogg Vorbis格式.

　　Vorbis是一个延续的音频压缩的开发始于 1993年克里斯蒙哥马利.集约发展始于1998年9月后，信弗劳恩霍夫协会 该公司宣布将收取许可费为 MP3音频格式。Vorbis的项目开始作为公司的一部分，Xiph.Org 基金会的Ogg项目（也称为 OggSquish多媒体项目）。克里斯蒙哥马利开始工作的项目，并协助越来越多的其它开发商。他们继续完善源代码 直到Vorbis的文件格式被冻结 2000年5月为 1.0 和一个稳定的版本（1.0）的参考软件发布于 2002年7月19日。

**OggSpeex**

　　Ogg 的语音编码，专门针对低码率的语音编码。

**Ogg FLAC**

Ogg 的无损音频编码。

**On2 VPX 系列**

On2 公司开发了一系列优秀的视频编码，现在应用得最多的恐怕是 NullsoftVideo 的视频，它们就采用了 VP3，VP5，VP6 视频编码。

**VP3**

　　已经作为开放源代码公布，现在 是 Ogg Theora 项目，当然，Theora 的质量可比 VP3 好多了。

**VP4**

　　On2 公司当年吹牛全球最好的视频编码，后来证明质量很一般。

**VP5**

　　至今还很神秘，On2 并没有放出来，只在 NullsoftVideo 里面见到他的身影。

**VP6**

　　从一开始，On2 就把这个编码器提供给大家下载，质量还是不错的。不过最近似乎又关闭了，主页上只有一个解码器。On2 TrueMotion VP6是一个专有的有损视频编解码格式和视频编解码器。它是TrueMotion视频编解码的具体体现，是一系列由On2开发的视频编解码，通产个用于 Adobe flash，Flash Video和JavaFX媒体文件。

**VP7**

　　On2 最新的编码器，在 VP6 上有不少进步。在2005年1月，On2宣布推出比VP6有更好的压缩比的新的编解码VP7。在2005ian4月，On2公司许可On2视频编码器9包 括VP6和VP7）用于Macromedia Flash。在2005年8月，Macromedia宣布他们选择VP6作为新的Flash Player8的视频回放的旗舰式编解码。

**VP8**

　　Google在2009年收购了On2 Technologies，并于2010年5月19日在Google I/O会议上宣布将VP8以BSD许可证的形式开源。。VP8是On2 Technologies继VP3之后宣布开源的第二个编解码器。（Xiph.Org 基金会于2002年接手VP3并将之冠名以Theora，之后以BSD许可证的形式将Theora开源）。要求Google将VP8开源的最大呼声来自于 自由软件基金会。2010年3月12日，自由软件基金会给Google发了一封公开信，请求Google逐渐用开源形式的VP8和HTML 5取代YouTube上的Adobe Flash和H.264。

　　2010年5月19日，WebM启动。WebM包含了来自于Mozilla、Opera、Google以及其他四十多家出版商和计算机软硬件供 应商（包括AMD、NVIDIA）的贡献，旨在大力倡导在HTML5中使用VP8。InternetExplorer 9在安装了适合的编解码器之后也能支持VP8。

注：上述材料来源于wiki的整理。

## [编解码学习笔记（十一）：Flash Video系列](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://javasogo.iteye.com%2Fblog%2F1015195" \t "_blank)

用于在 Flash 中压缩视频。FLV流媒体格式是一种新的视频格式，它的出现有效地解决了视频文件导入Flash后，使导出的SWF文件体积庞大，不能在网络上有效使用等 缺点。一般FLV文件包在SWF PLAYER 的壳里，并且FLV可以很好的保护原始地址，不容易被下载到，从而起到保护版权的作用。

* 文件名： .flv, .f4v, .f4p, .f4a, .f4b
* 媒体类型：video/x-flv, video/mp4, video/x-m4v, audio/mp4a-latm, video/3gpp, video/quicktime, audio/mp4
* 厂家：Adobe Systems (originally developed by Macromedia)
* Type of format： Media container
* Container for ：Audio, video, text, data Extended from
* 扩展：FLV: SWF ，F4V: MPEG-4 Part 12

**Flash介绍**

　　Flash Video是一个文件容器格式，是Adobe Flash播放器版本6-10用于在互联网上传递视频。Flash视频内容也可以封装在SWF文件。Flash视频有两种不同的文件格式：FLV和 F4V。FLV文件中，同SWF文件一样音频和视频数据采用相同方式进行编码。晚出现的F4V，它的格式是基于ISO为基础的媒体文件格式，并在 Flash Player 9 update 3中开始得到支持。这些格式Adobe Flash播放器都能支持，并由Adobe公司开发，其中FLV最找石油Macromedia公司开发的。

　　Flash视频FLV文件所包含的媒体的编码通常采用Sorenson Spark和VP6视频压缩格式。最新发布的Flash播放器支持H.264视频和HE-AAC音频。所有的这些编解码目前受到专利的限制。

　　Sorenson编解码看参考以下两种专用的视频编解码：Sorenson Video或者Sorenson Spark。Sorenson Video也被称为Sorenson Codec，Sorenson Video Quantizer或者SVQ。Sorenson Spark也称为Sorenson H.263。这些编解码都是有Sorenson 媒体公司设计（及以前的Sorenson Vision公司）。Sorenson Video在Apple的QuickTime中使用，SorensonSpark在Adobe Flash（以前的MacromediaFlash）中使用。

　　Flash视频通过广泛使用的Adobe Flash播放器和浏览器的plugin或者其他的第三方程序，使它能在绝大多是的操作系统都可以使用。

通常Flash视频FLV文件包含的视频比特流是一个专有的H.263视频标准的变体，FourCC为 FLV1（Sorenson Spark）。SorensonSpark是一个FLV文件老式编解码，但被广泛应用和兼容，因此它是第一个被Flash Player支持的视频编解码。这是在Flash Player 6和7要求的视频压缩格式。Flash Player 8和更新的版本支持On2 TrueMotion VP6视频比特流回放（FourCC VP6F或者FLV4）。On2 VP6是FlashPlayer 8或者更高版本优先使用的视频压缩格式。On2 VP6可以提供能够提供比Sorenson Spark更高视觉质量，尤其在低比特流中。另外它的计算更为复杂，因此在某些古老的系统配置中无法很好使用。

　　Flash 9 update 3，在2007年12月3日释放，提供了新的Flash视频文件格式F4V，支持H.264视频标准（也即MPEG-4part 10或者AVC），H.264需要更复杂的技术，但是提供更加卓著质量/比特流比例。具体而言，Flash播放器现在支持H.264视频压缩（MPEG- 4 Part 10）,AAC音频压缩（MPEG-3 Part 3），F4V，MP4（MPEG-4 Part14），M4V，3GP和MOV多媒体容器格式，3GPP Timed Text标准（MEPG-4 Part 17）（这是一个标准的的字幕格式，能够部分解析ID3的’ilist’，等同于iTunes使用的metadata存储。不支持MPEG-4 Part 2视频（例如有DivX或者Xvid创建）。Jonathan Gay，一个Flash的主要程序员在接收BBC新闻采访时表示：公司最初想在Flash中使用H.264，但是被每年5百万美金（3.5百万英镑）的专 利许可费而却步。

　　Flash Video FLV文件格式支持两个成为“screenshare‘（Screen Video）编解码的版本，这是一种用于桌面演示的编码格式。这两种格式都是基于tmap平铺方式，可以通过减少色深进行有损编码，并使用zlib压缩。 第二个版本之在Flash播放器8以及更新的版本支持。

　　在Flash视频文件中，通常使用MP3作为音频编码然而，在Flash视频FLV文件通过麦克风录音使用专用 的Nellymoser Asao编解码（Flash Player 10 在2008年发布，也支持开源的Speex编解码）。FLV文件支持不压缩的音频或ADPCM音频格式。最新的Flahs Player 9 支持AAC（HE-AAC/AAC SBR，AAC Main Profile，和AAC-LC）。

　　编码为Flash Video文件有一个编码工具提供，包括Adobe的Flash Professional和Creative Suite产品，On2的Flix编码工具，SorensonSqueeze，FFmepg和其他第三方工具。

**容器**

　　在2002年发布的Flash Player6增加了对SWF文件格式的支持。在2003年，FlashPlayer7增加了对FLV文件格式的直接支持。由于FLV文件格式的限制，Adobe System公司在2007年提出下面列出的新的文件格式，是基于ISO基本媒体文件格式（MPEG-4 Part 12）。Flash播放器不检查文件的扩展名，而直接查看文件，检查属于那种格式。

|  |  |  |
| --- | --- | --- |
| 文件扩展名 | Mime Type | 描述 |
| .f4v | video/mp4 | Video for Adobe Flash Player |
| .f4p | video/mp4 | Protected Video for Adobe Flash Player |
| .f4a | video/mp4 | Audio for Adobe Flash Player |
| .f4b | video/mp4 | Audio Book for Adobe Flash Player |

　　在Flash Player6及以后的版本对SWF文件的支持，使得可以与Adobe Flash媒体服务器通过RTMP来进行音频，视频和数据的交互。Flash媒体服务器的数据支持采用FLV文件格式的文件（MIME类型为 video/x-flv）。从Flash Player 9 Update 3开始创建的SWF文件，Flash Player可以播发新的F4V文件格式。

**媒体格式**

　　在FLV文件中支持的媒体类型：

* 视频：On2 VP6，Sorneson Spark（Sorenson H.263），Screen Video，H.264
* 音频：MP3，ADPCM，Linear PCM，Nellymoser，Speex，AAC，G.711（保留用于互通需求）

　　在F4V文件中支持的媒体类型：

* 视频：H.264
* 图像（视频数据的静止帧）：GIF，PNG，JPEG
* 音频：AAC，HE-AAC，MP3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 在Flash Player和Flash Video中支持的音视频压缩格式 | | | | |
| **Flash Player version** | **Released** | **File format** | **Video compression formats** | **Audio compression formats** |
| 6 | 2002 | SWF | Sorenson Spark, Screen video | MP3, ADPCM, Nellymoser |
| 7 | 2003 | SWF, FLV | Sorenson Spark, Screen video | MP3, ADPCM, Nellymoser |
| 8 | 2005 | SWF, FLV | On2 VP6, Sorenson Spark, Screen video, Screen video 2 | MP3, ADPCM, Nellymoser |
| 9.0.115.0 | 2007 | SWF, FLV | On2 VP6, Sorenson Spark, Screen video, Screen video 2, H.264[\*] | MP3, ADPCM, Nellymoser, AAC[\*] |
|  |  | SWF, F4V, ISO base media file format | H.264 | AAC, MP3 |
| 10 | 2008 | SWF, FLV | On2 VP6, Sorenson Spark, Screen video, Screen video 2, H.264[\*] | MP3, ADPCM, Nellymoser, Speex, AAC[\*] |
|  |  | SWF, F4V, ISO base media file format | H.264 | AAC, MP3 |

[\*]在FLV文件格式中使用H.264和AAC压缩有一些限制，Flash Player的作者强烈推荐大家使用新的F4V文件格式。

**Flash传递的几种方式**

一、作为一个标准的flv文件。

二、嵌入SWF文件，使用Flash认证工具（在FlashPlayer ６及以后的版本支持）。

三、通过HTTP的渐进流下载（progressive download）。这种方式使用ActionScript，包括客户端侧的一个外部托管Flash Video文件用于播放。然而，和使用RTMP的媒体流不一样，HTTP“流”不支持实时广播。HTTP流要求一个定制的播放器以及包含每个关键帧精确开 始字节位置以及时间码的特定FlashVideo元数据的加入。使用这些特定的信息，定制Flash Video播放器可以要求在任何指定的关键帧中开始播放。例如，Google Video，Youtube和BitGravity支持渐进流下载，可以在缓存满之前查看视频的任何部分。在服务器侧，这种“假HTTP流“方式实现相当 简单，例如可以采用Apache的PHP模块，使用lighttpd。

四、采用RTMP协议的流，可提供的有Flash媒体服务器（以前称为Flash Communication Server），VCS，Electro Server，Helix Universal Serval，Wowza Pro，用于.NET的WebORB，用于Java的WebORB，以及开源的Red5服务器。在2008年4月，这个协议有流录像提供，不需要重新编码 的screencast软件。

　　RTMP，实时消息协议，Real Time Message Protocol是一个有Adobe System为在互联网的音频，视频和数据流开发的私有协议，运行在Flash播放器和服务器之间。RTMP协议有三个方式：  
1、 通过在TCP上，使用1935端口的“纯”协议。   
2、 用于在穿越防火墙时，在HTTP请求中封装的RTMPT。   
3、 在HTTPS的安全连接中使用的RTPMS。

注：上述材料来源于wiki的整理。

## [编解码学习笔记（十二）：其他编解码](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://javasogo.iteye.com%2Fblog%2F1015194" \t "_blank)

### M-JPEG

　　M-JPEG（Motion-JoinPhotographicExpertsGroup）技术即运动静止图像（或逐帧）压缩技术，广泛应用于非线性编辑领域可精确到帧编辑和多层图像处理，把运动的视频序列作为连续的静止图像来处理，这种压缩方式单独完整地压缩每 一帧，在编辑过程中可随机存储每一帧，可进行精确到帧的编辑，此外M-JPEG的压缩和解压缩是对称的，可由相同的硬件和软件实现。

　　同样格式的MPEG视 频压缩不同于帧间压缩，因为压缩比特率比较低，所以编码与 解码相对比较容易，并不需要过多的运算能力，也使得软件或者芯片可以十分容易地对Motion JPEG进行编辑。也因为此，一些移动设备，如数码相机使用MotionJPEG来进行短片的编码。

### Motion JPEG 2000

　　JPEG2000是基于小波变换的图像压缩标准，由Joint PhotographicExperts Group组织创建和维护。JPEG2000通常被认为是未来取代JPEG（基于离散余弦变换）的下一代图像压缩标准。JPEG2000文件的副档名通常为.jp2，MIME类型是image/jp2。

　　虽然JPEG2000在技术上有一定的优势，但是到目前为止（2006年），互联网上采用JPEG2000技术 制作的图像文件数量仍然很少，并且大多数的浏览器仍然没有缺省支持JPEG2000图像文件的显示。但是，由于 JPEG2000在无损压缩下仍然能有比较好的压缩率，所以JPEG2000在图像品质要求比较高的医学图像的分析和处理中已经有了一定程度的广泛应用。

### DivX

* 文件名扩展： .divx
* 类型：DIVX
* 开发者： DivX，Inc
* 格式类型：媒体容器，用于MPEG-4 Part 2–compliant video
* 扩展来源：AVI

　　这是由MPEG－4衍生出的另一种视频编码(压缩)标准，也即通常所说的DVDrip格式，它采用了MPEG4 的压缩算法同时又综合了MPEG-4 与MP3各方面的技术，说白了就是使用DivX压缩技术对DVD盘片的视频图像进行高质量压缩，同时用MP3或AC3对音频进行压缩，然后再将视频与音频 合成并加上相应的外挂字幕文件而形成的视频格式。其画质直逼DVD并且体积只有DVD的数分之一。这种编码对机器的要求也不高，所以DivX视频编码技术 可以说是一种对DVD造成威胁最大的新生视频压缩格式，号称DVD杀手或DVD终结者。

　　DivX，是DivX公司（前身是DivXNetworks公司）的著名品牌，是一种MPEG-4技术视频编译码器（codec），2007年秋以2200万美元收购德国MainConcept。

　　ISO公布了“超低比特率活动图像和语音压缩标准 ”，排序MPEG－4,1998年10月批准第一版，1994年4月又公布了第二版及其校验模型（VM），MPEG－4正式编号是ISO/IEC国际标准 14496，它是一种新型的多媒体标准，它与前标准一个重要区别就在于，是一个基于对象的视编码压缩标准，所定义的码率控制的目标就是获得在给定码率下的最优质量，它为互联网上传输高质量的多媒体视频提供了很好的技术平台。

　　1998年微软开发了第一个在PC上使用的MPEG－4编码器，它包括MS MPEG4V1、MSMPEG4V2、MS MPEG4V3的系列编码内码，其中V1和V2用来制作AVI文件，一直到现在它都是作为Windows的默认组件，不过V1和V2的编码质量不是很好，一直到MS MPEG4V3才开始有好转，画质有了显着的进步，但是不知微软出于什么目的，却将这个MS MPEGV3的视频编码内核封闭，仅仅使其应用于Windows Media流媒体技术上，也就是我们熟悉的ASF流媒体文件中。ASF文件虽然有一些优势，但是由于过分的封闭不能被编辑，末得到广泛应用，这便惹怒了那些个不怕天不怕地的视频黑客和致力于钻研视频编码的高手，后来，这些小组不仅破解了微软的视频编码，而且经过他们的修改，一种新的视频编码诞生了：那就是 广为流传的MPEG编码器－DivX3.11。

　　DivX采用了MS的MPEGV3，改良后并加入自己功能称之为DivX3.11，也是目前互联网上普通采用的 MPEG－4编码器之一。很快，DivX被传得红得发紫，几乎成了业界的标准，但是，同样很快地出现了，DivX的基础技术是非法盗用微软的，微软声称将 对所有推动DivX发展的人、企业进行追究，可是DivX技术的创造者之一罗达（Rota）正全面申请将DivX合法化，这是基于DivX虽然是从 Window的发明出来的，但却没有用过任何微软的技术，更组建新公司DivXNetworks全力推广DivX，看来DivX（俗称压缩电影）蓬勃发展 的大潮是势不可挡了。

　　看来任何吸引眼球的故事在关键时刻都会发生转折，DivX的发展竟也不能脱离这一俗套，就在DivX顺利发展时 期，DivX的技术逐渐成熟，商机无限的时候，一台好戏上演了，DivXNetworks成立初衷就是摆脱微软的技术封闭，因而发起一个完全开放源码的项目，名为 “ Projet Mayo ”，目标是开发一套全新的、开放源码的MPEG4编码软件，由于它完全符合ISO　MPEG标准，又是完全开放源代码，OpenDivXCODEC吸引了 很多软件，视频高手参与，很快便开发出具有更高性能的编码器Encore2等等，就在DivX最辉煌的时期，DXN公司突然封闭了DivX的源代码，并在 Encore2的基础上发布了自有产品DivX4，原来DXN早就给自己留了后门，DivX采取的是LGPL协议，而不是GPL协议，虽说它们都是公共许 可证协议，保障自由使用和修改软件或源码的权利，但LGPL允许私有，DXN就是利用这一协议初其不备的耍出了大刀。

　　接着，很多被DXN公司狠狠涮了一回的软件、视频团体另起门户，逐渐重新聚拢开发力量，高举复仇大旗，在OpenDivX版本基础上，再次开发出一种新的MPEG－4编码--XviD，名字的顺序和DviX刚好相反，仅仅从名字就可以看出Xvid充满了复仇的力量。

　　DivX是近一两年来称霸网络视频的图像压缩编码标准。起初它是以微软MPEG 4视频编码标准为基础修改和开发的，并以免费方式发布。其特点是具有十分不错的压缩比率，可以将一整套DVD质量的影片压缩存放到一张CD-R光盘中。现 在的DivX分为普通版和Pro版，其中后者还有收费版和Adware（广告）版两种，自带DivX Player程序进行播放。用户如果安装了免费的DivX Codec后，也可用Windows Media player观看DivX影片。

注：上述材料来源于wiki的整理。

## [编解码学习笔记（十三）：容器](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://javasogo.iteye.com%2Fblog%2F1015193" \t "_blank)

视频是现在电脑中多媒体系统中的重要一环。为了适应储存视频的需要，人们设定了不同的视频文件格式来把视频和音频放在一个文件中，以方便同时回放。视频档实际上都是一个容器里面包裹着不同的轨道，使用的容器的格式关系到视频档的可扩展性。

### FourCC

FourCC全称Four-Character Codes，是由4个字符（4 bytes）组成，是一种独立标示视频数据流格式的四字节，在wav、avi档案之中会有一段FourCC来描述这个AVI档案，是利用何种codec来 编码的。因此wav、avi大量存在等于“IDP3”的FourCC。

|  |  |
| --- | --- |
| ISO/IEC | [MPEG-PS](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMPEG_program_stream) **·** [MPEG-TS](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMPEG_transport_stream) **·** [MPEG-4 Part 12](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMPEG-4_Part_12) /[JPEG 2000 Part 12](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FJPEG_2000%23Motion_JPEG_2000) **·** [MPEG-4 Part 14](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMPEG-4_Part_14) |
| ITU-T | [H.222.0](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMPEG-2%23Systems) |
| Others | [3GP and 3G2](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2F3GP_and_3G2) **·** [ASF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAdvanced_Systems_Format) **·** [AVI](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAudio_Video_Interleave) **·** [Bink](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FBink_Video) **·** [DivX Media Format](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FDivX_Media_Format) **·** [DPX](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FDigital_Picture_Exchange) **·** [EVO](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FEnhanced_VOB) **·** [Flash Video](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FFlash_Video) **·**[GXF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FGeneral_Exchange_Format) **·** [M2TS](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2F.m2ts) **·** [Matroska](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMatroska) **·** [MXF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMaterial_Exchange_Format) **·** [Ogg](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FOgg) **·** [QuickTime File Format](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FQuickTime_File_Format) **·** [RealMedia](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FRealMedia) **·**[REDCODE RAW](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FREDCODE) **·** [RIFF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FResource_Interchange_File_Format) **·** [Smacker](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FSmacker_video) **·** [MOD and TOD](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FMOD_and_TOD_%2528video_format%2529) **· VOB ·** [WebM](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FWebM) |
| Audio Only | [AIFF](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAudio_Interchange_File_Format) **·** [AU](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FAu_file_format) **·** [WAV](http://blog.csdn.net/byxdaz/article/details/link.php?url=http://en.wikipedia.org%2Fwiki%2FWAV) |

### 3GP和3G2容器

　　3GP（3GPP文件格式）是一个多媒体容器由第三代合作伙伴计划（3GPP）针对3G UMTS多媒体服务定义。它用于3G移动电话，但也可以用于某些2G和4G的电话。3GP在ETSI 3GPP技术规范中定义，他是视频文件格式，并带有讲话/音频媒体类型和带时间信息的文本，用于IMS，MMS，多媒体广播/多播服务（MBMS）和传输端到端的包交换流媒体服务（PSS）。

　　3G2（3GPP2文件格式）是一个多媒体容器有3GPP2为3G CDMA 2000多媒体服务定义。她与3GP文件格式非常相似，但与之相比存在一些扩展和限制。3G2在3GPP2技术规范中定义。

　　3GP和3G2文件格式都是基于在ISO/IEC 14496-12（MPEG-4 Part 12）定义的ISO基础媒体文件格式，但是老板不的3GP文件格式不具有其中某些属性。3GP和3G2与MP4（MPEG-4 Part 14）相似，MP4也是基于MPEP-4 Part 12。3GP和3G2设计目的是为移动电话减少存储和带宽要求，它们是非常相似的标准，但有区别：

* 3GPP 文件格式用于GSM类电话，文件扩展名：.3gp
* 3GPP2文件格式用于CDMA类电话，并具有文件扩展名：.3g2

　　3GP文件存贮视频流：MPEG-4 Part2，H.263，MPEG-4 Part 10（AVC/H.264），音频流 AMR-NB, AMR-WB, AMR-WB+,AAC-LC, HE-AAC v1 和 Enhanced aacPlus (HE-AAC v2)。3GPP允许ISO基础文件格式（MPEG-4Part12）中使用AMR和H.263编解码，因为3GPP在ISO基础文件格式中规定了采样条目和模板字段的使用，可以为编解码定义新的box。这些 扩展在ISO基础媒体文家格式（“MP4家族”文件）中由登记授权登记为code-point。对于在3GP文件中存贮MPEG-4媒体，3GP规定参加了MP4和AVC文件格式规范，它们也是基于ISO基础媒体文件格式。MP4和AVC文件格式规范描述在ISO基础媒体文件格式中使用MPEG-4内容。 有些手机使用.mp4作为3GP视频的扩展。

　　3G2文件格式可以存贮与3GP文件格式相同的视频流和觉得部分的音频流。此外3G2可以存在音频流还包括 EVRC，EVRC-B，EVRC- WB，13K（QCELP），SMV，和VMR-WR。3G2规范还定义了在3GPP带时间文件的某些货站。3G2文件格式不支持Enhanced aacPlus (HE-AAC v2) 和AMR-WB+ audiostreams。对于在3G2文件中存在MPEG-4媒体（AAC音频，MPEG-4 Part 2音频，MPEG-4 Part 10/H.264/AVC），3G2规范提及了MP4 文件格式和AVC文件格式规范，在那里描述了如果在ISO基础媒体文件格式中使用这些内容。对于在3G2中存储H.263和AMR内容，3G2规范参见了 3GP文件格式规范。

　　3GP格式视频有两种分辨率：

* 分辨率176×144，适合市面上所有支持3GP格式的手机。
* 分辨率320×240，清晰，适合高档手机、MP4播放器、PSP以及苹果iPod.

### ANIM

　　ANIM标准的多媒体文件用于经典的Commodore Amiga的数字动画。它遵循IFF ILBM主规范，他是第一个动画格式被操作系统正式采纳。

### ASF

　　微软WMA和WMV的标准容器。

　　WMV（Windows Media Video）是微软公司开发的一组数字视频编解码格式的通称，ASF（Advanced Systems Format）是其封装格式。ASF封装的WMV档具有"数字版权保护"功能。扩展名:wmv/asf、wmvhd。

　　ASF (Advanced Streaming format高级流格式)。ASF 是 MICROSOFT为了和现在的 Real player 竞争而发展出来的一种可以直接在网上观看视频节目的文件压缩格式。ASF使用了 MPEG4 的压缩算法，压缩率和图像的质量都很不错。因为 ASF 是以一个可以在网上即时观赏的视频"流"格式存在的，所以它的图像质量比 VCD 差一点点并不出奇，但比同是视频"流"格式的 RAM 格式要好。

* 文件扩展名 ：.asf .wma .wmv
* 互联网媒体类型： video/x-ms-asf, application/vnd.ms-asf
* 类型码 ： 'ASF\_'
* 唯一类型码 ：Identifier com.microsoft.advanced-systems-format
* Magic number ：30 26 b2 75
* 开发者 ：Microsoft
* 格式类型 ：Container format
* 容器容纳 ：WMA, WMV, MPEG4 etc.

### AVI

　　AVI (the standard Microsoft Windows container, also based on RIFF)。AVI是英语Audio Video Interleave（"音频视频交织"或译为"音频视频交错"）的首字母缩写，由微软在 1992年11月推出的一种多媒体文件格式，用于对抗苹果Quicktime的技术。现在所说的AVI多是指一种封装格式。

　　比较早的AVI是Microsoft开发的。其含义是Audio Video Interactive，就是把视频和音频编码混合在一起存储。AVI也是最长寿的格式，已存在10余年了，虽然发布过改版（V2.0于1996年发布），但已显老态。AVI格式上限制比较多，只能有一个视频轨道和一个音频轨道（现在有非标准插件可加入最多两个音频轨道），还可以有一些附加轨道，如文字等。AVI格式不提供任何控制功能。扩展名：avi。

　　AVI能使用的编码：

* 视频名称（括号内表示的是此视频的FourCC）   
  o MPEG-1/-2 (MPEG/MPG1/MPG2)  
  o MPEG-4 (MP4V/XVID/DX50/DIVX/DIV5/3IVX/3IV2/RMP4)  
  o MS-MPEG4 (MPG4/MP42/MP43)  
  o WMV7/WMV8/WMV9 (WMV1/WMV2/WMV3)  
  o DV(DVSD/DVIS)  
  o Flash Video (FLV1/FLV4)  
  o Motion JPEG (MJPG)  
  o LossLess JPEG (LJPG)  
  o H.264 (AVC1/DAVC/H264/X264)  
  o H.263 (H263/S263)  
  o H.261 (H261)  
  o Huffyuv (HFYU)  
  o AVIzlib (ZLIB)  
  o AVImszh (MSZH)  
  o Theora (THEO)  
  o Indeo Video (IV31/IV32)  
  o Cinepak (cvid)  
  o Microsoft Video 1 (CRAM)  
  o On2VP3 (VP30/VP31)  
  o On2VP4 (VP40)  
  o On2 VP6 (VP60/VP61/VP62)  
  o VC-1 (WVC1)
* 音频   
  o PCM  
  o MP3 (0x0055)  
  o AC-3 (0x0092)  
  o AAC   
  　　－ HE-AAC  
  　　－ LC-AAC  
  o FLAC  
  o Indeo Audio  
  o TrueSpeech  
  o WMA  
  o Vorbis

　　编码组合能根据以下的例子自由选择。

* (DivX或XviD+MP3).avi，
* (H.264+MP3).avi
* (WMV9+MP3).avi

　　以XviD+MP3构成的AVI最为常见。

### DVB－MS

　　DVR-MS (Microsoft Digital Video Recording，微软数字视频录制)是一种专用的视频和音频文件容器格式，有微软开发，用于存储由Windows XPMedia Center Edition，Windows Vista和Windows 7录制的电视内容。多个数据流（视频和音频）在带有DVR-MS扩展的ASF容器中封装。视频使用MPEG-2标准编码，音频使用MPEG-1 Layer II或者杜比数字AC-3(ATSC A/52)。扩展的格式包括内容和数字版权管理的元数据。这些格式的文件有流缓存引擎（SBE.dll）生成，这是一个在Windows XP Service Pack 1的DirectShow组件。

### MPEG/MPG/DAT

　　MPEG格 式：MPEG(Moving Picture Experts Group)，是一个国际标准组织（ISO）认可的媒体封装形式，受到大部份机器的支持。其存储方式多样，可以适应不同的应用环境。MPEG-4档的档容 器格式在Part 1(mux)、14(asp)、15(avc)等中规定。MPEG的控制功能丰富，可以有多个视频（即角度）、音轨、字幕（位图字幕）等等。MPEG的一个简化版本3GP还广泛的用于准3G手机上。扩展名:dat(用于VCD)、vob、mpg/mpeg、3gp/3g2（用于手机）等。

　　MPEG也是Motion Picture Experts Group 的缩写。这类格式包括了 MPEG-1, MPEG-2 和 MPEG-4在内的多种视频格式。MPEG-1相信是大家接触得最多的了，因为目前其正在被广泛地应用在 VCD 的制作和一些视频片段下载的网络应用上面，大部分的 VCD 都是用 MPEG1 格式压缩的( 刻录软件自动将MPEG1转为 .DAT格式 ) ，使用 MPEG-1 的压缩算法，可以把一部 120 分钟长的电影压缩到 1.2 GB 左右大小。MPEG-2 则是应用在 DVD 的制作，同时在一些 HDTV（高清晰电视广播）和一些高要求视频编辑、处理上面也有相当多的应用。使用MPEG-2 的压缩算法压缩一部 120 分钟长的电影可以压缩到 5-8 GB 的大小（MPEG2的图像质量MPEG-1 与其无法比拟的）。

　　MPEG-PS：MPEG节目流（programstream），是MPEG-1和MPEG-2基准流的标准容器，用于在可靠介质上，例如磁盘，也用于DVD-Video光碟。

　　MPEG-TS：MPEG传输流，是数字广播和在非可靠媒体传输的标砖容器，也在蓝光光碟使用，通常携带多个视频和音频流以及一个电子节目指南。

### AVI

　　如果你发现原来的播放软件突然打不开此类格式的AVI文件，那你就要考虑是不是碰到了n AVI。n AVI是 New AVI 的缩写，是一个名为 Shadow Realm 的地下组织发展起来的一种新视频格式。它是由MicrosoftASF 压缩算法的修改而来的（并不是想象中的 AVI），视频格式追求的无非是压缩率和图像质量，所以 NAVI 为了追求这个目标，改善了原始的 ASF 格式的一些不足，让 NAVI 可以拥有更高的帧率。可以这样说，NAVI 是一种去掉视频流特性的改良型 ASF 格式。

注：上述材料来源于wiki的整理。

### Matroska（MKV）

　　MKV ，不是任何的编解码或者系统的标准，但实际上可封装任何的东西。是一个开放以及开源的容器格式。

扩展名 　　　　　.mkv .mka .mks   
互联网媒体类型 　video/x-matroska audio/x-matroska 开  
发者 　　　　　　Matroska.org   
格式 　　　　　　视频文件格式   
专门属 　　　　　多媒体   
自由文件格式？ 　Yes: GNU LGPL

　　Matroska，很多人把它当作为MKV，其实MKV只 是Matroska媒体系列的其中一种文件。Matroska是一种新的多媒体封装格式，这个封装格式可把多种不同编码的视频及16条或以上不同格式的音频和语言不同的字幕封装到一个Matroska Media档内。它也是其中一种开放源代码的多媒体封装格式。

　　多媒体封装格式，简称MCF、多媒体容器，是一个开放（没有身份规限，免费）及自由把数据存放的格式。开发者承诺大家可以自 由地使用这种格式和经这种格式所开发的软件；又不会在这种格式普遍的时候变成一个商业的科研项目。

Matroska媒体定义了三种类型的档：

* MKV (Matroska Video File) ：视频档，可以包含音频和字幕；
* MKA (Matroska Audio File) ：单一的音频档，可以有多条及多种类型的音轨；
* MKS (Matroska Subtitles) ：字幕文件。

　　这三种文件中以MKV最为常见。

　　Matroska最大的特点就是能容纳多种不同类型的视频编码、音频编码及字幕流，并且它能把非常高密的RealMedia及QuickTime文 件也容纳在内，同时将它们的音频和视频重新组织起来，从而达到一个更好和鲜明的效果。

　　Matroska的开发是对多种传统媒体格式的一次大挑战，虽则如此，Matroska也被开发成一个多功能的多媒体容器。

### MP4

　　MP4，是MPEG-4定义的标准音视频容器，基于ISO基础媒体文件格式（在MPEG-4 Part 12以及JPEG 2000 Part 12中定义），在MPEG-4 Part 14中描述。是一种使用MPEG-4的多媒体电脑档案格式，副档名为.mp4，以储存数码音讯及数码视讯为主。

扩展名 　　　　.mp4   
互联网媒体类型 video/mp4, audio/mp4, application/mp4   
类型代码 　　　mpg4   
开发者　　　　 ISO   
格式 　　　　　视频文件格式   
专门属 　　　　Audio, video, text  
延伸自 　　　　QuickTime .mov and MPEG-4 Part 12   
标准　　　　　 ISO/IEC 14496-14

### MOD

　　MOD格式是JVC生产的 硬盘摄录机所采用的存储格式名称。

### MOV

　　MOV是评估公司的标准QuickTime视频容器。QuickTime Movie是由苹果公司 开发的容器，由于苹果电脑在专业图形领域的统治地位，QuickTime格式格式基本上成为电影制作行业的通用格式。1998年2月11 日,国际标准组织（ISO）认可QuickTime文件格式作为MPEG-4标准的基础。QT可存储的内容相当丰富，除了视频、音频以外还可支持图片、文字（文本字幕）等。扩展名：mov

　　使用过Mac机的朋友应该多少接触过QuickTime。QuickTime原本是Apple公司用于Mac计 算机上的一种图像视频处理软件。 Quick-Time提供了两种标准图像和数字视频格式 , 即可以支持静态的PIC和JPG图像格式，动态的基于Indeo压缩法的MOV和基于MPEG压缩法的MPG视频格式。

### Ogg

　　Ogg是Xiph.org音频编解码Vorbis和视频编解码Theora的标砖容器，Ogg Media一个完全开放性的多媒体系统计划，OGM（Ogg Media File）是其容器格式。OGM可以支持多视频、音频、字幕（文本字幕）等多种轨道。扩展名：ogg。

### OGM

　　OGM（Ogg Media），是Xiph.ofg的视频编解码容器，已经不再支持，并不鼓励使用。

### RealMedia

　　RealMedia是RealVideo和RealAudio的标准容器。Real Video或者称Real Media(RM)档是由RealNetworks开发的一种档容器。它通常只能容纳Real Video和Real Audio编码的媒体。该档带有一定的交互功能，允许编写脚本以控制播放。RM,尤其是可变比特率的RMVB格式，体积很小，非常受到网络下载者的欢迎。 扩展名：rm/rmvb

### RM

　　Real Networks公司所制定的音频/视频压缩规范Real Media中的一种，Real Player能做的就是利用Internet资源对这些符合Real Media技术规范的音频/视频进行实况转播。在Real Media规范中主要包括三类文件：RealAudio、Real Video和Real Flash （Real Networks公司与Macromedia公司合作推出的新一代高压缩比动画格式）。REAL VIDEO （RA、RAM）格式由一开始就是定位就是在视频流应用方面的，也可以说是视频流技术的始创者。它可以在用 56K MODEM 拨号上网的条件实现不间断的视频播放，可是其图像质量比VCD差些，如果您看过那些RM压缩的影碟就可以明显对比出来了。

### RMVB

　　这是一种由RM视频格式升级延伸出的新视频格式，它的先进之处在于RMVB视频格式打破了原先RM格式那种平均 压缩采样的方式，在保证平均压缩比的 基础上合理利用比特率资源，就是说静止和动作场面少的画面场景采用较低的编码速率，这样可以留出更多的带宽空间，而这些带宽会在出现快速运动的画面场景时被利用。这样在保证了静止画面质量的前提下，大幅地提高了运动图像的画面质量，从而图像质量和文件大小之间就达到了微妙的平衡。另外，相对于DVDrip 格式，RMVB视频也是有着较明显的优势，一部大小为700MB左右的DVD影片，如果将其转录成同样视听品质的RMVB格式，其个头最多也就400MB 左右。不仅如此，这种视频格式还具有内置字幕和无需外挂插件支持等独特优点。要想播放这种视频格式，可以使用RealOne Player2.0或RealPlayer8.0加RealVideo9.0以上版本的解码器形式进行播放。

### VOB

　　VOB文件（video Object）是一个DVD视频媒体的容器格式。VOB可以包含视频，音频，字母和菜单整合在一个流格式中。VOB是基于MPEG PS格式，但是有额外的限制和私有流的规范。MPEG PS提供非标准数据称为私有流。VOB文件是MEPG PS表站中非常严格的子集。经所有的VOB文件都是MPEG PS，但不是所有的MPEG PS都遵守VOB文件的定义。

　　与MPEG的PS相似，VOB文件可以包含H.262/MPEG-2 Part2或者MPEG-1 Part 2视频，MPEG-1 Audio LayerII或者MOEG-2 Audio Layer II音频，但是和MPEG PS相比，在VOB文件中使用这些压缩格式有某些限制。此外，VOB可以包括线性PCM，AC-3或者DTS视频以及字母。VOB文件不能包含AAC音频（MPEG-2 Part 7），MPEG-4压缩格式或者其他，而这些在MPEG PS标准中是允许的。

　　文件扩展名 .VOB   
　　开发者： 　DVD Forum   
　　类型：　　 媒体容器   
　　包含：　　 音频，视频，字母   
　　用于： 　　DVD-Video   
　　从何扩展： MPEG program stream, ISO/IEC 13818-1   
　　标准规范： DVD-Video Book

注：上述材料来源于wiki的整理。