google开源了WebRTC项目

======cymlife原创，转载请注明。======

WebRTC实现了基于网页的视频会议，标准是[WHATWG](http://www.whatwg.org/specs/web-apps/current-work/complete/video-conferencing-and-peer-to-peer-communication.html) 协议，目的是通过浏览器提供简单的javascript就可以达到实时通讯（Real-Time Communications (RTC)）能力。

通过对源码的粗略分析，WebRTC提供了视频会议的核心技术，包括音视频的采集、编解码、网络传输、显示等功能，并且还支持跨平台：windows，linux，mac，android。

WebRTC的视频处理分析（windows平台）  
  
WebRTC的视频部分，包含采集、编解码(I420/VP8)、加密、媒体文件、图像处理、显示、网络传输与流控(RTP/RTCP)等功能。

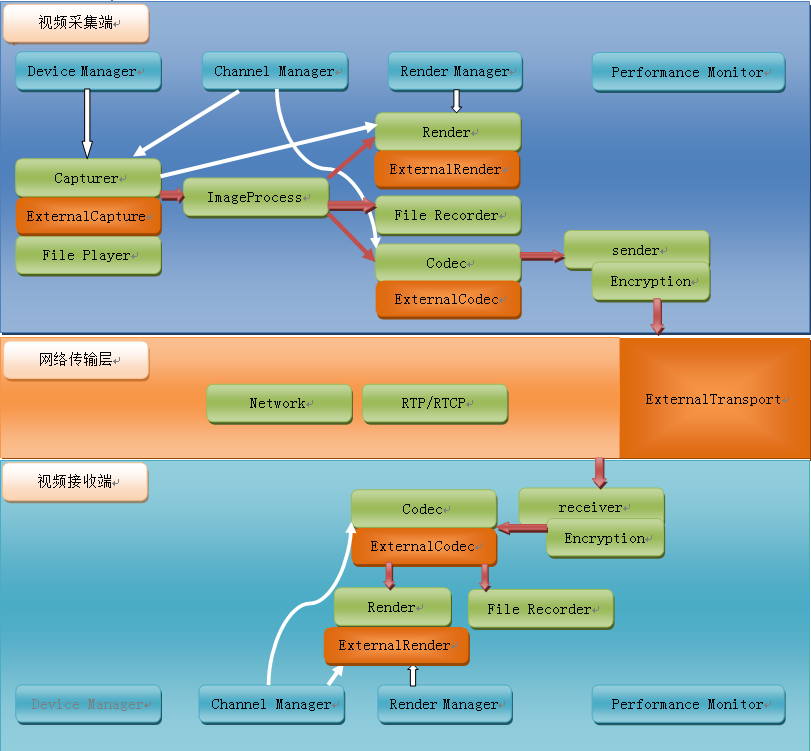
**视频采集---video\_capture**

源代码在webrtc/modules/video\_capture/main目录下，包含接口和各个平台的源代码。  
在windows平台上，WebRTC采用的是dshow技术，来实现枚举视频的设备信息和视频数据的采集，这意味着可以支持大多数的视频采集设备；对那些需要单独驱动程序的视频采集卡（比如海康高清卡）就无能为力了。  
视频采集支持多种媒体类型，比如I420、YUY2、RGB、UYUY等，并可以进行帧大小和帧率控制。  
  
**视频编解码---video\_coding**   
源代码在webrtc/modules/video\_coding目录下。  
WebRTC采用I420/VP8编解码技术。VP8是google收购ON2后的开源实现，并且也用在WebM项目中。VP8能以更少的数据提供更高质量的视频，特别适合视频会议这样的需求。  
  
**视频加密--video\_engine\_encryption**   
视频加密是WebRTC的video\_engine一部分，相当于视频应用层面的功能，给点对点的视频双方提供了数据上的安全保证，可以防止在Web上视频数据的泄漏。  
视频加密在发送端和接收端进行加解密视频数据，密钥由视频双方协商，代价是会影响视频数据处理的性能；也可以不使用视频加密功能，这样在性能上会好些。  
视频加密的数据源可能是原始的数据流，也可能是编码后的数据流。估计是编码后的数据流，这样加密代价会小一些，需要进一步研究。  
  
**视频媒体文件--media\_file**   
源代码在webrtc/modules/media\_file目录下。  
该功能是可以用本地文件作为视频源，有点类似虚拟摄像头的功能；支持的格式有Avi。  
另外，WebRTC还可以录制音视频到本地文件，比较实用的功能。  
  
**视频图像处理--video\_processing**   
源代码在webrtc/modules/video\_processing目录下。  
视频图像处理针对每一帧的图像进行处理，包括明暗度检测、颜色增强、降噪处理等功能，用来提升视频质量。  
  
**视频显示--video\_render**   
源代码在webrtc/modules/video\_render目录下。  
在windows平台，WebRTC采用direct3d9和directdraw的方式来显示视频，只能这样，必须这样。  
  
**网络传输与流控**   
对于网络视频来讲，数据的传输与控制是核心价值。WebRTC采用的是成熟的RTP/RTCP技术。

WebRTC的音频处理分析（windows平台）  
  
WebRTC的音频部分，包含设备、编解码(iLIBC/iSAC/G722/PCM16/RED/AVT、NetEQ)、加密、声音文件、声音处理、声音输出、音量控制、音视频同步、网络传输与流控(RTP/RTCP)等功能。  
  
**音频设备---audio\_device**   
源代码在webrtc/modules/audio\_device/main目录下，包含接口和各个平台的源代码。  
在windows平台上，WebRTC采用的是Windows Core Audio和Windows Wave技术来管理音频设备，还提供了一个混音管理器。  
利用音频设备，可以实现声音输出，音量控制等功能。  
  
**音频编解码---audio\_coding**   
源代码在webrtc/modules/audio\_coding目录下。  
WebRTC采用iLIBC/iSAC/G722/PCM16/RED/AVT编解码技术。  
WebRTC还提供NetEQ功能---抖动缓冲器及丢包补偿模块，能够提高音质，并把延迟减至最小。  
另外一个核心功能是基于语音会议的混音处理。  
  
**声音加密--voice\_engine\_encryption**   
和视频一样，WebRTC也提供声音加密功能。  
  
**声音文件**   
该功能是可以用本地文件作为音频源，支持的格式有Pcm和Wav。  
同样，WebRTC也可以录制音频到本地文件。  
  
**声音处理--audio\_processing**   
源代码在webrtc/modules/audio\_processing目录下。  
声音处理针对音频数据进行处理，包括回声消除(AEC)、AECM、自动增益(AGC)、降噪处理等功能，用来提升声音质量。  
  
**网络传输与流控**   
和视频一样，WebRTC采用的是成熟的RTP/RTCP技术。

WebRTC的官网是<http://www.webrtc.org/>，其中的FAQ解答了很多大家关心的问题。有这么几点可以关注：  
1.license  
采用BSD licenses，可以用于商业软件；即使你修改了源代码，也不必公开你修改的部分。  
  
2.为什么WebRTC要免费  
音视频针对目前的互联网应用，还是至关重要的。google这样做的目的，无非是吸引大量的开发者在chrome平台上开发音视频相关应用，和当年微软windows集成IE是一个道理。  
  
3.下载和编译  
a).下载google的源码工具depot\_tools：  
svn co http://src.chromium.org/svn/trunk/tools/depot\_tools  
并把depot\_tools添加到系统的环境变量PATH里；  
  
b).下载git工具：并把git添加到系统的环境变量PATH里；  
  
c).创建webrtc工作目录，并在该目录下运行：  
gclient config https://webrtc.googlecode.com/svn/trunk  
gclient sync --force  
  
sync命令也会产生基于本地环境(Linux:make, OSX: XCode, Windows: Visual Studio)的build文件。  
  
也可以用命令：  
gclient runhooks --force  
  
对windows平台，在trunk目录下，已经有针对Visual Studio的工程文件：  
webrtc.sln  
  
好了，打开它，这就可以编译了。  
  
注：webrtc默认使用了C:/Program Files/Microsoft SDKs/Windows/v7.1/Samples/multimedia/directshow/baseclasses。我的机器上只有v7.0的sdk，于是构造了一个v7.1的目录，仅包含Samples/multimedia/directshow/baseclasses。

WebRTC的video\_engine系统架构图，如下：



1. WebRTC提供了一个框架，对于视频部分而言，已经解决了从视频采集、编码、传输、显示的全部功能。

在架构图中，有颜色的箭头表示了视频流的数据流向：从视频采集端开始，经过网络传输层，再到视频接收端。

2. WebRTC的系统限制：

Device Manager最大可以管理10个输入设备；ChannelManager最大可以管理4个Channel；当然，你也可以修改这些最大值的。

Codec支持的最大分辨率是1920\*1200。

3. WebRTC的扩展

WebRTC不是一个封闭的系统，相反，它很容易扩展，包括Device，Codec,Render，Transport，都提供了可扩展的接口。

比如，可以增加H264的Codec，以方便加入对现有系统的支持。

对于视频会议，可以扩展网络传输层：当从视频采集端得到编码后的数据包时，直接通过视频会议的网络传输，由视频会议服务器发送到各个视频会议客户端，再交给WebRTC进行接收处理。这就达到了我们想要的功能。