PPBOX SDK 集成开发白皮书

修订记录

日期	作者	版本	主要内容	备注
2010-2-23	汪坤 王闻宇	1.0.0.1		
2010-2-25	汪坤 王闻宇	1.0.0.2	细化细节	为了便于理解，取消异步模式
2010-3-3	王闻宇	1.0.0.3	添加接口 去掉 handle	修改接口(见粗体文字)
2010-3-9	王闻宇	1.0.0.4	修改接口	修改接口
2010-3-15	王闻宇	1.0.0.5	修改接口	修改接口
2010-3-16	王闻宇	1.0.0.6	修改接口	修改接口
2010-3-18	王闻宇	1.0.0.7	修改接口	修改接口
2010-3-23	王闻宇	1.0.0.8	修改接口	添加 异步的打开视频函数 AsyncOpen 这样可以实现异步的打开
2010-3-25	王闻宇	1.0.0.9	修改接口	添加对 P2Pengine 的支持 新增函数 PPBOX_StartP2PEngine 新增函数 PPBOX_StopP2Pengine 将之前的 PPBOX_Stop 函数 改名的 PPBOX_Close
2010-4-7	郭春茂	1.0.0.10	统一文档格式 增加模型描述	
2010-4-8	仲巍	1.0.0.11	修改接口	添加扩充接口： PPBOX_GetStreamInfoEx PPBOX_ReadSampleEx2
2010-5-10	郭春茂	1.0.0.12	细化细节	细化对流格式描述
2010-5-21	郭春茂	1.0.0.13	增加模型描述的影片缓冲、下载驱动	
2010-6-9	郭春茂	1.0.0.14	细化细节	细化对 Sample 格式描述
2010-6-11	郭春茂	1.0.0.15	细化细节	细化对 Sample 格式描述
2010-11-29	郭春茂	1.0.0.16	增加 Sample 格式描述	独立一章
2011-04-19	项东涛	1.0.0.17	修改 format_type, 增加接口	根据新直播流修改 type 定义，增加故障检测接口
2011-04-21	项东涛	1.0.0.18	接口调用推荐方式	增加接口调用推荐方式
2011-06-01	郭春茂	1.0.0.19	增加内容	增加系统结构、移植需求、媒体规格、安装配置、集成开发章节

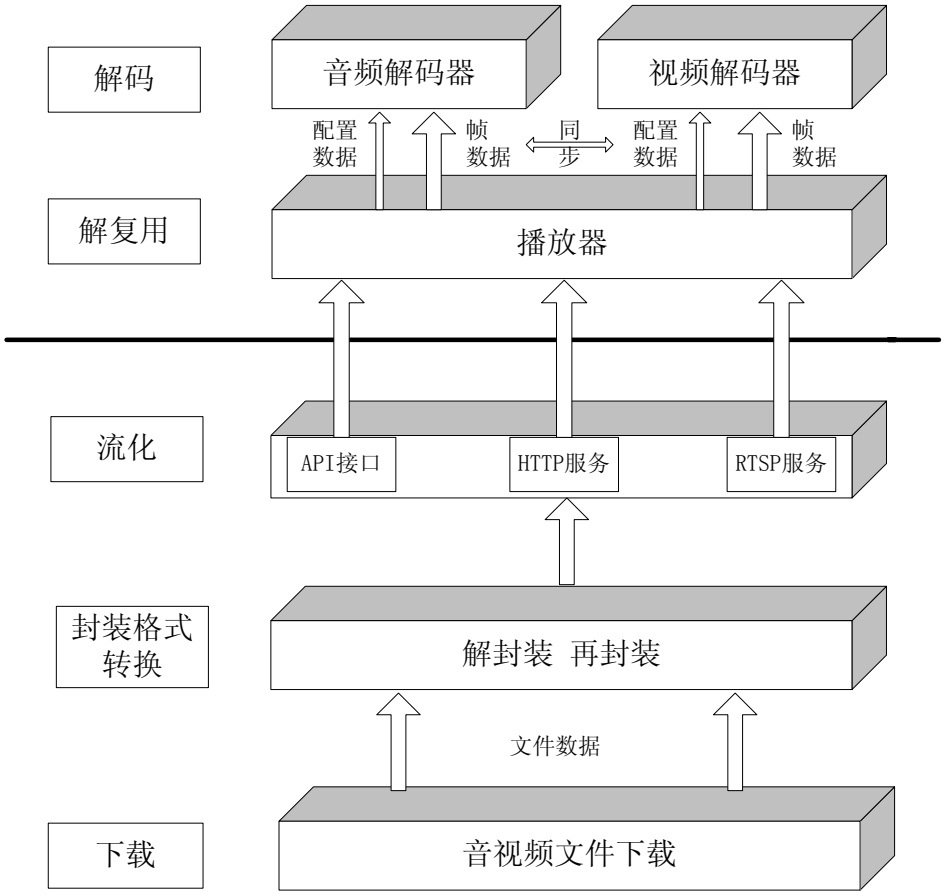
2011-06-03	郭春茂	1.0.0.20	增加内容	增加故障诊断流程
2012-02-18	郭春茂	1.1.0.0	大整理	去除 API 播放修改内容
2012-02-22	郭春茂	1.1.1.0	增加内容	增加插入播放的描述内容

目录

1 概述

本文档描述了 PPBox 提供的对外三种接口类型：API 接口、HTTP 接口以及 RTSP 接口，同时描述了这三种接口如何集成 PPBox 音视频内容，分别从播放系统架构、平台移植需求、媒体规格、安装调试、开发方法等方面对集成工作做了说明。

2 系统结构



我们把整个播放系统分成“下载”、“封装格式转换”“流化”、“解复用”和“解码”无个部分，其中前三个部分是 PPBox 开发库已有的功能，集成方需要提供或者开发后两个部分功能。集成方一般以播放器插件的形式集成 PPBox 的 API 接口，或者用播放器直接连接 PPBOX 的 HTTP 服务、RTSP 服务获取音视频流。

3 移植说明

3.1 硬件需求

需求		指标	备注
文件储存		7.7M～19.2M	不同平台有很大不同，一般使用 glibc 比 uclibc 大，arm 比 mips 大
内存	下载缓存	8M	可静态配置（点播），播放状态才有此消耗
	Demux 缓存	6M	可动态配置，播放状态才有此消耗
	代码映射	7.7M～19.2M	与文件大小有很大关系
	其他	4M	零散对象，堆内存，共享内存
CPU	下载	30%～50%	播放状态才有此消耗
	Demux	5%	播放状态才有此消耗
	其他	1%	

注：以上是完整功能配置，还可根据实际需要裁减

3.2 软件需求

此处只描述了 Linux（或者类 Linux）系统平台的情形，绝大部分情况都是此类平台。我们也支持 Window 平台（包括 WinCE）。

3.2.1 C 库

C 库涉及功能列表

需求	使用 API	备注
内存 ✧ 文件映射 ✧ 内存页	mmap munmap getpagesize	
文件 ✧ 文件创建 ✧ 文件读写 ✧ 文件锁 ✧ 文件句柄复制 ✧ 管道	open64 open close read write readv writev lseek dup2 pipe ioctl	为了增加日志写效率，采用了 writev 系统调用。 ioctl 用来枚举网卡设备。 fcntl 使用文件锁功能，只需要支持建议锁，不需要强制锁。

	fcntl poll	
文件系统 ✧ 目录创建 ✧ 目录读取 ✧ 符号链接 ✧ 文件重命名 ✧ 程序当前路径 ✧ 文件属性	remove statvfs64 link readlink readdir64_r closedir opendir utime readdir64 chdir rename getcwd pathconf symlink mkdir	PPBOX 依赖一些与文件系统相关的环境变量，包括 TMP。 环境变量 TMP 指向的目录必须是 tmpfs 文件系统，正常情况下我们会使用其中的 800KB 存储空间，其中 400KB 用于日志，300KB 用于进程间共享内存交互，其他的用于域名解析临时缓存文件，文件锁临时文件。在很多系统中，共享内存使用的是系统 API 创建，所以有可能不在 TMP 文件系统中。 为避免文件名字冲突，列出 PPBOX 使用的所有文件目录名（在 TMP 目录下）： ppbox.log、ppbox_alive.log、vod_worker.log、live_worker.log、framework_1.2.0.2231（随版本不同版本号会变化）。
时间 ✧ 本地时间 ✧ 时间分解 ✧ 时间格式化 ✧ 系统单调时间 ✧ 时区	ctime gmtime_r strftime timezone localtime_r gettimeofday mktime time clock_gettime	嵌入式平台的系统时间是通过启动后网络对时获取的，这就会出现系统时间的跳变，这种跳变会影响程序的时间依赖行为。我们采用了 clock_gettime 获取系统单调时间 CLOCK_MONOTONIC，来避免这个问题，但是一些系统 API 还是依赖于真实时间，比如 pthread_cond_timedwait，所以尽可能在网络对时之后启动 PPBOX。 在 freebsd 系统中没有 mktime，使用 timegm 代替，此时不需要 timezone
网络 ✧ 套接字创建 ✧ 套接字收发数据 ✧ 域名解析 ✧ IO 分派	recvmsg shutdown inet_ntop getaddrinfo inet_pton getsockopt sendmsg freeaddrinfo if_nametoindex if_indextoname connect select setsockopt socket	epoll、kqueue、select，只要支持一种就可以，按顺序优先使用
进程 ✧ 进程创建	unsetenv kill	

✧ 进程信号 ✧ 进程环境变量 ✧ 进程等待子进程	sigfillset getppid execvp getenv alarm _exit getpid setenv sigaction fork waitpid sched_yield nanosleep	
进程间通信 ✧ 共享内存	shm_open shm_unlink shmget shmat shmctl shmdt	对嵌入式平台，我们使用文件锁来实现进程间互斥锁。 shm_*（POSIX 标准）和 shm*（SYSV 标准）只要支持其中之一就可以，在两种都不支持的情况下，我们采用文件映射来实现内存共享。两者都支持的时候优先采用 SYSV 标准。
系统	get_nprocs sysconf	get_nprocs 如果不支持，我们直接假定其返回值为 1
线程 ✧ 线程创建 ✧ 线程互斥锁 ✧ 线程条件变量 ✧ 线程信号 ✧ 线程局部储存	pthread_cond_destroy pthread_once pthread_mutex_trylock pthread_sigmask pthread_cond_wait pthread_cond_timedwait pthread_join pthread_key_delete pthread_create pthread_cond_init pthread_mutex_lock pthread_mutex_init pthread_mutex_unlock pthread_detach pthread_setspecific pthread_cond_broadcast pthread_key_create pthread_getspecific pthread_mutex_destroy pthread_cond_signal	支持 LinuxThreads 线程模型 在 LinuxThreads 线程模型下，多线程进程有多个 pid，会在/proc 文件系统中对应多项，这个我们做了合并处理。一个线程创建的进程，另一个同一进程的线程不能 wait，我们确保了同一个线程去 fork、wait 子进程。
动态库	dlopen	dladdr 可选

✧ 动态库加载 ✧ 动态库符号解析	dlsym dlclose dlerror dladdr	
内存操作 ✧ 内存申请、释放 ✧ 内存拷贝、比较	realloc malloc free memset memcmp memchr memcpy memmove	
数学运算 ✧ 浮点数 ✧ 随机数	ceil log rand srand	
字符串 ✧ 字符串比较 ✧ 字符串拷贝 ✧ 字符串分割 ✧ 字符串查找	strcpy strncmp strchr strrchr strncpy strcmp strlen strstr strcat ftok	
字符集 ✧ 字符分类 ✧ 整数转换 ✧ 大小写转换 ✧ 格式化	isgraph isalpha isalnum isspace atoi atol towlower* towupper* tolower toupper atof snprintf vsnprintf sprintf sscanf	宽字符集可选
C 文件（FILE*） ✧ 文件创建、打开	ftello64 fscanf	

✧ 文件读写 ✧ 文件定位	fputs fseek fflush fclose feof fprintf fwrite fgets fputc ferror fopen64 fopen	
标准输入输出	printf stdin stdout stderr	
错误码	strerror_r errno	

3.2.2 C++库

C++库涉及功能列表

需求	使用 API	备注
字符串	std::basic_string	wchar 可选
流	std::basic_istream std::basic_ostream std::basic_streambuf <iostream> <fstream> <stringstream>	
容器	std::list std::vector std::map std::deque std::hash_map std::hash_set std::set	
异常	std::exception	

3.2.3 系统

需求	指标	备注

gcc	版本>=3.4	支持 c++98 标准
内存文件系统	tmpfs	存放程序运行中产生的临时文件
本地回环网络（lo）	接口 127.0.0.1 tcp port 9000-9006	

4 媒体规格

参见《PPTV Media Specification》。

5 安装配置

5.1 取得授权

请通过商务途径联系，获取软件授权验证码。该验证码通过 API 接口传给 PPBOX 软件库。

5.2 取得开发包

请向 ppbox 索取开发包，在此之前我们需要你们提供 toolchain 给 ppbox，并且提供编译，链接的特殊参数。

5.3 开发包目录结构

- PPBOX 中间层接口.doc	本文档
+ include/	接口头文件目录
|-- IPpbox.h	外围接口函数定义
|-- IDemuxer.h	播放相关接口函数定义
|-- IAdapter.h	播放适配层接口函数定义
|-- IDownloader.h	下载相关接口函数定义
- ppbox.conf	接口链接库配置文件
- ppbox_test.conf	测试程序配置文件
+ test/	测试程序源码
|-- FileWriter.h	将接口输出数据重新组装成分段文件的代码
|-- PlaySession.cpp	模拟一次播放过程（对各个接口的调用示例）
|-- Test.cpp	包含主函数的文件
|-- AudioStreamWriter.h	提取视频数据（H.264 ES 流）的示例代码
|-- VideoStreamWriter.h	提取音频数据（MPEG4 AAC）的示例代码
|-- PlaySession.h	

|-- InputHandler.h
|-- Common.cpp
|-- Common.h
|-- WaitClose.h
+ [平台名]/ 平台相关的二进制文件目录
|-- libppbox-[P]-[T]-mt-[V].so 接口链接库
|-- ppbox_alive-[P]-[T]-mt-[V] 性能统计程序
|-- vod_worker-[P]-[T]-mt-[V] 点播数据下载进程
|-- libpeer-[P]-[T]-mt-[V].so 点播数据核心链接库
|-- live_worker-[P]-[T]-mt-[V] 直播数据下载进程
|-- liblive-[P]-[T]-mt-[V].so 直播数据核心链接库
|-- ppbox_test-[P]-[T]-mt 接口测试程序

注：二进制文件名中[P]、[T]、[V]分别代表“平台名称”、“编译器名称版本”、“接口库版本”。

5.4 运行环境搭建

5.4.1 安装步骤

1. 拷贝二进制可执行文件

将二进制文件目录下的 vod_worker-[P]-[T]-mt-[V]、live_worker-[P]-[T]-mt-[V]、ppbox_alive-[P]-[T]-mt-[V]拷贝到/usr/bin 目录（或者/usr/local/bin 目录，或者任意目录，但该目录需要通过 PATH 环境变量导出）

2. 拷贝二进制库文件

将二进制文件目录下的 libppbox-[P]-[T]-mt-[V].so、libpeer-[P]-[T]-mt-[V].so、liblive-[P]-[T]-mt-[V].so 拷贝到/usr/lib 目录（或者/usr/local/lib 目录，或者任意目录，但该目录需要通过 LD_LIBRARY_PATH 环境变量导出）

3. 指定临时目录

导出环境变量 TMP，指示临时文件目录，（如果没有导出，则默认是/tmp）。该临时文件目录必须是 tmpfs 文件系统，否则如果是固化储存、频繁读写会产生硬件损耗、减少设备寿命。
以上环境变量的设置应该写入 shell 初始化脚本中。

5.4.2 测试安装

通过测试程序测试安装的正确性，需要做以下步骤：

1. 拷贝测试程序和配置文件

拷贝二进制文件目录下的 ppbox_test-[P]-[T]-mt 和 ppbox.conf、ppbox_test.conf 到任意目录，但必须在同一目录。

2. 配置测试程序

修改配置文件 ppbox_test.conf，将验证码填入相应位置。

```
……
[Certify]
gid=1
pid=2
auth=ABCDEF1234567890 ABCDEF1234567890
```

3. 启动测试程序

./ppbox_test-[P]-[T]-mt

4. 验证正确性

一段时间后（视网络情况时间长短不一），如果出现下面类似的信息，则说明启动成功。

```
 [DEBUG] [Daemon] [start] end
<2012-02-16 11:15:32>
[EVENT] [Certifier] certify: success
```

5. 验证播放

在 PC 上，使用 vlc player，播放下面的 URI：

http://192.168.1.65:9006/play.ts?playlink=ppvod%3a%2f%2f1d7c1WCcaMih2c7j4a5ampmYzM6UyGpamMibkMukipaWZ5aYzcyUxphelrqkmdbfmQ%3d%3d

注意修改地址“192.168.1.65”为设备的地址。

5.5 开发环境搭建

先参考运行环境搭建方法在开发环境中做相同安装步骤。此外，包含的头文件和链接的库文件说明如下。

5.5.1 头文件

本文档描述的 API 接口只需要包含（include）IDemuxer.h，HTTP 接口和 RTSP 接口只需包含 IPpbox.h。

5.5.2 库文件

播放器需要链接 ppbox 库，文件名 libppbox-[P]-[T]-mt-[V].so。

6 集成开发

集成开发主要分为通用功能和播放功能两部分，其中通用功能需要调用 PPBox 提供的 API 接口，而播放功能又分为 API 接口、HTTP 接口和 RTSP 接口三种方式，下面将进行详细介绍。

6.1 引擎启动停止

启动引擎接口（***PPBOX_StartP2PEngine***）需要在其他接口之前调用，停止引擎接口（***PPBOX_StopP2PEngine***）在其他接口之后调用。

如果播放第二个影片需要重启播放器进程，那么最好通过一个精灵进程来启动引擎。下面将解释这样做的原因以及如何实现。

实际上启动引擎需要做两件工作：第一件是启动服务程序（下载服务、认证服务），第二件是初始化当前进程的必要状态。其中第一件工作比较耗时，但是如果精灵进程已经通过启动引擎接口启动了服务程序，那么播放器进程就可以跳过，从而加快了启动过程，使得播放体验更好。

关于实现，精灵进程在开始运行后，调用启动引擎接口，收到退出信号（比如 kill SIG_INT）后，调用停止引擎接口，然后自己也结束。

UI 在进入 PPBOX 的内容时，启动精灵进程，在退出 PPBOX 的内容时，停止（通过 kill SIG_INT）精灵进程。

每当播放一个影片时，播放器进程启动，一开始也调用启动引擎接口，这样做是为了初始化播放器进程的必要状态，播放完成后，播放器进程直接退出，不需要调用停止引擎接口。

6.2 音视频播放

6.2.1 音视频 URI

- 格式

  <protocol>://<host>:[<port>]/<path>?<parmlist>
  <protocol>:///<path>?<parmlist>
  注：某些协议不需要指定服务器名称端口。
  <protocol>://<path>?<parmlist>

注：表示相对路径。

- 协议类型

协议	说明	路径格式	支持参数
ppvod	PPTV 点播	PPTV 点播串	format
pplive	PPTV 直播	PPTV 直播串	
pplive2	PPTV 二代直播	PPTV 二代直播串	
http	HTTP 音视频	HTTP 路径	
file	本地音视频	本地文件路径	
pipe	管道输入	管道号.封装格式	

- URI 举例

ppvod://1d7c1WCcaMih2c7j4a5ampmYzM6UyGpamMibkMukipaWZ5aYzcyUxphelrqkmdbfmQ%3d%3d

pplive://29+p3dnV5KqbmKyWnqGhnqSdnqKko66VqKCn3dnV5KqbmKiVop6dmqiToqCamqaaqqicoaag5NPc3bCUn6KemqSXoKSam6qWnqiko66VqKCn0erZ4OSmmKWXoqGam6aZnqKgmqSdqKqkmbHN5OTc3bCUn6Kcm6SWoaKam6aToaCho66Vq9jg3ebZqp+bn6eToaWhl66Toqimoaag5dTc3bCUn6KemaSWpqWamqqToqemoaaVoKvhzebZqp+bmqeenqGioKSXoaiaoKefqKCcmbHa1ODgo6WUpqGamquanqiam62fqKCcmbHa1ODgo6WUpaman6mToaWil6+Wqqicmaag5dTc3bCUn6KdoaSboZ6il6eapKqkmaaVq+XQ2eqfn5+em6eToqCgl6iZoZ6kmrCdoKCcpOvJ4OSmmKWboZ6dnquTqJ6enLCdoKCcpJymxK2dmaeXltPUyuTT1dyp5NeW0aCkmqzLndXQmdySpNOloKPG1qKllqie1aXSnduZ1aahoPOL09nQpqicpKGSwtvR3N/judfM1bnQpq+eoZavyurG3N/Tstqip6Kgna2L5tmpn5y5xq2S19fS1a0mZzo0PjA2PJzU5K2cj+XRrZbP3bOWlg==

pplive2://e9301e073cf94732a380b765c8b9573d-5

http://192.168.1.100/1.ts

file://yu.mp4

pipe://4.flv

6.2.2 播放协议

基于 PPTV API 流媒体协议的播放方式请参照协议文档《PPTV API Streaming Protocol》。

基于 PPTV HLS 协议的播放方式请参照协议文档《PPTV HLS Protocol》。

基于 PPTV HTTP 流媒体协议的播放方式请参照协议文档《PPTV HTTP Streaming Protocol》。

基于 PPTV HTTP 扩展流媒体协议的播放方式请参照协议文档《PPTV HTTP Extend Streaming Protocol》。

基于 PPTV RTSP 协议的播放方式请参照协议文档《PPTV RTSP Protocol》。

协议	URI 协议	封装格式	说明
API	<无>	ES（默认） MPEG2 TS ASF FLV MKV	ES 格式是指没有封装的音视频数据其他封装格式，每次读取的是一个封装单元
HTTP Extend	http	MPEG2 TS ASF FLV MKV	
HLS	http	M3U8/TS	
RTSP	rtsp	ES（默认） MPEG2 TS	ES 格式是指 RTP 直接封装编解码数据，采用扩展名 es MPEG2 TS 格式是指 RTP 封装 MPEG2 TS 数据
HTTP	http	MPEG2 TS ASF FLV MKV	

注：表格中排列顺序对应 PPBOX 推荐次序。

6.2.3 播放 URI

播放 URI 用于 HTTP（包括 HLS），RTSP 协议，不适用于 API 协议（API 协议直接使用影片 URI）。

- 格式
  - 正常格式

    <protocol>://<host>:[<port>]/<path>?<parmlist>

  - Base16 编码格式

    <protocol>://<host>:[<port>]/base16_Base16::Encode(<path>?<parmlist>)

  - Base64 编码格式

    <protocol>://<host>:[<port>]/base64_Base64::Encode(<path>?<parmlist>)

注：某些播放器不支持 URL 参数，为此，提供后两种格式（Base16，Base64）。

- 路径

在基于 HTTP 的协议（包括 HLS）中，路径必须是“play”，因为 PPBOX 的 HTTP 协议族流媒体服务器除了支持基本的播放功能外，还支持其他功能，比如录制功能（路径为“record”）。

在其他协议中，路径可以是播放串，也可以是其他任何内容。

- 参数列表

参数	参数意义	说明
playlink	音视频 URI	1、需要做 URI 编码，去除特殊字符（API 协议除外） 2、playlink 也可以通过播放 URI 路径表示（基于 HTTP 的协议除外）
format	输出的封装格式	1、封装格式定义在《PPTV Media Specification》中（4 节），采用扩展名作为 format 参数的值 2、format 可以附加在路径名后面，如 play.ts 表示 format 为 ts
time	播放开始时间点	3、可以省略，默认为 0，单位：秒

- URI 举例

  http://127.0.0.1:9006/play.ts?playlink=file%3a%2f%2fyu.mp4

  rtsp://127.0.0.1:5054/play?playlink=file%3a%2f%2fyu.mp4&format=es

6.3 插入播放

插入播放将多个影片文件合并成一个音视频流输出。该功能不需要播放器做任何修改，但是使用 PPTV HTTP 流协议的情况下，因为无法预估 HTTP 返回的文件大小，不支持插入播放。

插入播放主要用视频广告的插播。

插入播放的流程如下：

1. 设置插入影片

在播放影片之前，通过 ***PPBOX_InsertMedia*** 传入插入影片的信息（URI，时长，文件大小）和插入时间点。可以插入多个，插入的影片在合并到下次播放音视频流中。

2. 获取插入影片处理进度通知

通过 ***PPBOX_GetInsertMediaEvent*** 获取插入影片处理进度通知，比如调整插入时间点通知，影片开始插入通知，影片结束插入通知等。

6.4 故障诊断

故障诊断的原理是把 PPBOX 内部的日志信息暴露给外部调用者，在正常播放的同时获取诊断信息。故障诊断的流程如下：

3. 进入调试模式

调用 ***PPBOX_DebugMode*** 传入 true 参数进入调试模式，只有在调试模式，才能够获取到诊断信息。

4. 不断获取诊断信息

循环调用 ***PPBOX_DialogMessage*** 获取诊断信息并展现，同时进行正常播放。

5. 离开调试模式

当故障诊断完成后，调用 ***PPBOX_DebugMode*** 传入 false 参数离开调试模式。

注：诊断信息可以跨进程传递，所以可以在任何进程通过加载 PPBOX 库进入调试模式，并获取诊断信息，不必要在播放器进程。

7 API 接口索引

7.1 版本、错误信息

7.1.1 获得模块版本号

```
PP_char const * PPBOX_GetVersion();
```

参数：无

返回值：返回模块的版本号，例如 1.2.54.0

7.1.2 获取错误码

```
PP_int32 PPBOX_GetLastError();
```

参数：无

返回值：上次接口调用产生的错误码

7.1.3 获取错误描述

```
PP_char const * PPBOX_GetLastErrorMsg();
```

参数：无

返回值：字符串，对上次接口调用产生的错误的描述

7.2 P2P 引擎控制

7.2.1 启动 P2P 引擎

```
PP_int32 PPBOX_StartP2PEngine(
```

```
    PP_char const * gid,
    PP_char const * pid,
    PP_char const * auth);
```

参数：

gid	客户 ID，PPLive 提供，请向 PPLive 申请
pid	项目 ID，PPLive 提供，请向 PPLive 申请
auth	认证码，PPLive 提供，请向 PPLive 申请

返回值：错误码

返回 ***ppbox_success*** 表示成功，其他见错误码表。

7.2.2 停止 P2P 引擎

```
void PPBOX_StopP2PEngine();
```

参数：无

返回值：无

说明：

如果已经调用 ***PPBOX_Open*** 或者 ***PPBOX_AsyncOpen*** 成功打开视频，建议先调用 ***PPBOX_Close*** 关闭播放实例，再调用此函数，否则将强制调用 ***PPBOX_Close***

注意：

不要在一个影片播放结束后调用 ***PPBOX_StopP2PEngine***。在整个进程存活期间，应该至多调用 ***PPBOX_StartP2PEngine*** 与 ***PPBOX_StopP2PEngine*** 一次。

7.3 属性配置

```
void PPBOX_SetConfig(
    PP_char const * module,
    PP_char const * section,
    PP_char const * key,
    PP_char const * value)
```

参数：

module	指定设置配置的模块
section	指定配置的区域
key	指定配置的属性
value	指定配置属性对应的值

返回值：无

7.4 插入播放

7.4.1 设置插入影片

```
void PPBOX_InsertMedia(
    PP_uint count,
    InsertMedia const * media);
```

参数：

count	插入影片的个数
media	插入影片的信息

返回值：无

其中，InsertMedia 的数据结构如下：

```
typedef struct tag_InsertMedia
{
    PP_uint id;
    PP_uint time;
    PP_uint duration;
    PP_uint size;
    PP_uint report;
    PP_char const * url;
    PP_char const * report_begin_url;
    PP_char const * report_end_url;
} InsertMedia;
```

具体字段意义为：

id	影片 ID，由外部保证唯一，后续的处理进度通知中以该 ID 作为关键字
time	插入时间点
duration	插入的影片的时间长度
size	插入的影片的文件大小（字节）
report	是否需要将开始、结束插入的事件汇报到服务器
url	插入影片的 URI 地址，\0 结尾
report_begin_url	开始插入的事件汇报的请求 URI，\0 结尾
report_end_url	结束插入的事件汇报的请求 URI，\0 结尾

7.4.2 获取插入影片处理进度通知

```
PP_int PPBOX_GetInsertMediaEvent(
    InsertMediaEvent * event);
```

参数：

event 返回的通知事件，每次最多返回一个

返回值：错误码

返回 ***ppbox_success*** 表示成功获取到一个事件，返回 ***ppbox_would_block*** 表示暂时没有事件，其他见错误码表。

其中，InsertMediaEvent 的数据结构如下：

```
typedef struct tag_InsertMediaEvent
{
    PP_uint media_id;
    PP_uint event_time;
    PP_uint event_type;
    PP_uint argment;
} InsertMediaEvent;
```

具体字段意义为：

media_id 影片 ID，在 ***InsertMedia*** 中传入的 ID

event_time 事件产生的时间，time()返回值

event_type 事件类型，

0-调整插入时间

1-开始插入

2-停止插入

argment 事件附带的额外信息，

在类型 0 中表示调整后的插入时间点，

在类型 2 中表示实际插入时间长度（有可能用户提前结束播放）

7.5 故障检测

7.5.1 进入或者退出调试模式

```
void PPBOX_DebugMode(
    PP_bool mode )
```

参数：

mode 设置开关， true 进入，false 退出

返回值：无

说明：

在进入调试模式时，该接口可能会阻塞一段时间（至多 3 秒），以确保内部所有模块都进入调试模式。

7.5.2 获取诊断信息

```
PP_uint32 PPBOX_DialogMessage(
    DialogMessage * vector,
    PP_int32 count,
    PP_char const * module,
    PP_int32 level);
```

参数：

vector	保存诊断信息的数组
count	期望获取的诊断信息的条数，数组大小需要保证能够容纳
module	获取日志的模块名称，（传 NULL 表示获取任意模块的诊断信息）
level	指定获取诊断信息的最大优先级

返回值：实际获取信息的条数，0 表示没有获取到信息。

DialogMessage 的数据结构：

```
typedef struct tag_DialogMessage
{
    PP_uint32 time;
    PP_char const * module;
    PP_uint32 level;
    PP_uint32 size;
    PP_char const * msg;
} DialogMessage;
```

具体字段意义为：

time	该诊断信息产生的时间，time(NULL)返回的值
module	该诊断信息对应的模板名，\0 结尾
level	该诊断信息优先级
size	该诊断信息的长度，不包括\0 结尾
msg	诊断信息内容，\0 结尾

说明：

该接口在没有诊断信息的时候不会阻塞，所以外部调用者要注意不要频繁调用，以免损耗性能，在返回 0 的时候最好线程睡眠一段时间。

诊断信息的优先级值越小，对应的优先级越高。所以传递的 level 参数越大，获取的信息越多越详细。目前优先级的值有 1～5。

DialogMessage 结构体的内存有调用者提供，返回的 module、msg 的内存是接口内部存储，外界不需要释放，再次调用该接口函数时，接口内部存储就会失效。

7.5.3 提交日志文件

```
void PPBOX_SubmitMessage(PP_char const * msg,  PP_int32 size)
```

参数：

msg	提交给服务器的信息，
size	msg 的字节长度

返回值：无

调用者在发现有故障的时候，如果有必要的话，调用该接口将故障信息提交给我们的服务器，信息内容任意。

7.6 错误码表

枚举类型	描述
ppbox_success	正确，无错误
ppbox_not_start	P2P引擎没有打开
ppbox_already_start	P2P引擎已经打开
ppbox_not_open	视频还没有打开，需要调用Open
ppbox_already_open	视频已经打开
ppbox_operation_canceled	异步操作被取消
ppbox_would_block	操作不能立即完成
ppbox_stream_end	用于ReadSample的时候，表示视频流已经结束
ppbox_logic_error	逻辑错误，程序bug，请报告PPBox
ppbox_network_error	网络发生错误
ppbox_demux_error	Demux错误
ppbox_certify_error	认证错误
ppbox_other_error	其他错误

8 HTTP 接口索引

8.1 通用部分

8.1.1 请求格式

请求格式遵循标准的 HTTP 协议 1.1 版本（兼容 1.0 版本），采用 HTTP-GET 方法，采用带参数的 URL 格式。其他 URL 中的路径（URL 中域名之后，参数之前的部分）表示具体的请求接口，URL 中的参数表示请求接口的参数。

8.1.2 URL 格式

- 正常格式

http://<host>:[<port>]/<path>?<parmlist>

- Base16 编码格式

http://<host>:[<port>]/base16_Base16::Encode(<path>?<parmlist>)

- Base64 编码格式

http://<host>:[<port>]/base64_Base64::Encode(<path>?<parmlist>)

注：某些播放器不支持 URL 参数，为此，提供后两种格式。

8.1.3 错误应答报文格式

报文格式：

- HTTP 报文头（包括状态行和若干消息头）

常见的状态码：

404 (Not Found)：在指定的位置不存在所申请的资源。

500 (Internal Server Error)：服务器端发生错误。

- HTTP 报文体

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
    <module>???</module>
    <value>???</value>
    <message>???</message>
</root>
```

注：

module	错误分类名
value	错误码
message	具体的错误信息

例如：

```
HTTP/1.1 404 Not Found
Date: Thu, 26 Jul 2011 14:00:02 GMT
Content-Length: 105
```

```
Content-Type: text/xml
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
<root>
    <module>system</module>
    <value>2</value>
    <message>系统找不到指定的文件。</message>
</root>
```

8.2 音视频服务

音视频服务通过 9006 端口输出，该服务实现了 PPTV HTTP 相关流媒体协议，同时提供其他额外功能。

接口名称（URL 路径）	功能	应答内容（正常情形）	说明
mediainfo	或者影片的基本信息	xml 元数据	
play	请求影片的音视频数据	虚拟音视频文件	
info	获取播放状态信息	虚拟文件	
seek	调整前一个 play 的应答数据流，从新的时间点开始发送数据	无	不推荐使用
pause	暂停前一个 play 的应答数据流，暂停发送数据	无	
resume	恢复前一个 play 的应答数据流，恢复发送数据	无	

8.2.1 播放

http://<host_port>/play?format=<输出格式>&playlink=<播放串>

参数：

playlink: “播放串”是播放视频的串

type 指定播放类型， 以下是对应关系

ppvod 指定点播

pplive 指定一代直播

pplive2 指定二代直播

ppfile-asf 指定本地 asf 文件

ppfile-mp4 指定本地 mp4 文件

format 指定返回视频流的格式， 目前支持 flv，ts，asf，mkv 以及 m3u8 三种

seek 指定开始播放影片时的时间点，单位为秒

如播放一个点播， 则可以通过，
http://127.0.0.1:9006/play.flv?format=flv&type=ppvod&playlink=1d7c1WCcaMih2c7j4a5amsaY0M%2bUyWNanJSYz6CUyJRiXMimkM7QipWYXMnWmdbfmQ%3d%3d 链接返回这个视频的 flv 流。

调用成功返回相应的视频流。

注意：

如果 URL 中没有 format 参数指定返回视频流格式，则通过 play+”.”+”返回视频流格式”来指定；如果两种方式都指定返回视频流格式，以 format 指定格式为准。

8.2.2 获取影片信息

http://<host_port>/mediainfo?format=<输出格式>&playlink=<播放串>

参数：（同 play）

返回：

```
<?xml version="1.0" encoding="utf-8" ?>
<template module="mediainfo" version="1.0">
    <duration value="11547" />
    <video codec="h264">
        <property frame-rate="24" width="1280" height="720" />
    </video>
    <audio codec="aac">
        <property channels="2" sample-rate="48000" sample-size="16" />
    </audio>
</template>
```

注：

duration value	指视频的总时长，单位是秒
video codec	指视频的编码格式
frame-rate	指视频参数帧率
width	指视频参数宽度
height	指视频参数高度
audio codec	指音频的编码格式
channels	指音频参数频道数
sample-rate	指音频参数采样率
sample-size	指音频参数采样位宽

8.2.3 获取播放信息

http://<host_port>/info

参数：(无)

返回：(无)

```
<?xml version="1.0" encoding="utf-8" ?>
    <template module="info" version="1.0">
    <time value="30862" />
    <present value="100" />
    <status value="playing" />
</template>
```

注：

time value	指播放缓冲时间，单位是秒
present value	指缓冲百分比
status value	指播放缓冲状态

8.2.4 拖动

http://<host_port>/seek?time=<拖动目标时间点>

参数：

time 指定拖动后影片的播放时间点，单位为秒。

返回：(无)

8.2.5 暂停

http://<host_port>/pause

参数：(无)

返回：（无）

8.2.6 恢复

http://<host_port>/resume

参数：(无)

返回：（无）

8.3 其他服务

其他服务通过 9003 端口输出。

8.3.1 获取故障诊断信息

暂时不提供该接口。