

播放库

SDK 编程指南

(for Android)

V7.3.1.X

非常感谢您购买我公司的产品，如果您有什么疑问或需要请随时联系我们。

本手册可能包含技术上不准确的地方、或与产品功能及操作不相符的地方、或印刷错误。我司将根据产品功能的增强而更新本手册的内容，并将定期改进或更新本手册中描述的产品或程序。更新的内容将会在本手册的新版本中加入，恕不另行通知。

目 录

目 录	II
1 产品简介	1
2 SDK 版本更新	2
3 错误代码及说明	3
4 函数调用顺序	4
5 函数说明	5
系统操作及错误号获取	5
5.1 获取播放库版本号 GETSDKVERSION	5
5.2 获取错误码 GETLASTERROR	5
5.3 获取播放端口 GETPORT	5
5.4 释放播放端口 FREEPORT	5
文件操作	6
5.5 打开数据文件 OPENFILE	6
5.6 关闭数据文件 CLOSEFILE	6
流操作	6
5.7 设置流播放模式 SETSTREAMOPENMODE	6
5.8 打开流 OPENSTREAM	7
5.9 关闭流 CLOSESTREAM	7
5.10 输入流数据 INPUTDATA	7
播放控制	8
5.11 开启播放 PLAY	8
5.12 停止播放 STOP	8
5.13 暂停数据播放 PAUSE	8
5.14 快速播放 FAST	8
5.15 慢速播放 SLOW	9
5.16 以独占方式播放音频 PLAYSOUND	9
5.17 关闭音频 STOPSOUND	9
5.18 设置文件当前播放位置（百分比）SETPLAYPOS	9
5.19 获取文件当前播放位置（百分比）GETPLAYPOS	10
5.20 设置文件当前播放时间（毫秒）SETPLAYEDTIMEEX	10
5.21 获取文件当前播放时间（毫秒）GETPLAYEDTIMEEX	10
5.22 设置文件当前播放帧号 SETCURRENTFRAMENUM	10
5.23 获取文件当前播放帧号 GETCURRENTFRAMENUM	11
获取播放或解码信息	11
5.24 获取文件总时间 GETFILETIME	11
5.25 获取全局时间 GETSYSTEMTIME	11
5.26 获取文件总帧数 GETFILETOTALFRAMES	12

5.27	获取当前帧率 GETCURRENTFRAMERATE	12
5.28	获取已解码的视频帧数 GETPLAYEDFRAMES	12
5.29	获取原始图像大小 GETPICTURESIZE	12
	解码操作及控制	13
5.30	设置视频帧解码类型 SETDECODEFRAMETYPE	13
5.31	设置解码回调函数 SETDECODECB	13
5.32	文件结束回调 SETFILEENDCB	14
5.33	解码密钥 SETSECRETKEY	14
	显示操作	15
5.34	设置或增加显示区域 SETDISPLAYREGION	15
5.35	设置或销毁显示区域 SETVIDEOWINDOW	16
5.36	设置同步回放组 SETSYCGROUP	16
	缓冲区操作	17
	解码前源缓冲区	17
5.37	获取源缓冲区剩余数据大小 GETSOURCEBUFFERREMAIN	17
5.38	清空所有缓冲区的剩余数据 RESETSOURCEBUF	17
5.39	清空指定缓冲区的数据 RESETBUFFER	17
	解码后播放缓冲区	18
5.40	设置播放缓冲区最大缓冲帧数 SETDISPLAYBUF	18
5.41	获取播放缓冲区最大缓冲帧数 GETDISPLAYBUF	18
	源、播放缓冲区	18
5.42	获取指定缓冲区的大小 GETBUFFERVALUE	18
	索引	19
5.43	索引回调 SETFILEREFCB	19
	抓图	19
5.44	抓图回调 SETDISPLAYCB	19
5.45	截取当前画面 BMP 图片 GETBMP	20
5.46	截取当前画面 JPEG 图片 GETJPEG	20
	硬解码	21
5.47	设置硬解码优先 SETHARDDECODE	21
5.48	设置最大硬解码路数 SETMAXHARDDECODEPORT	21
5.49	获取当前解码类型 GETDECODERTYPE	22
	预录像功能	22
5.50	设置预录像数据回调 SETPRERECORDCALLBACK	22
5.51	设置预录像开关 SETPRERECORDFLAG	22
	鱼眼功能	23
5.52	广角图像矫正 SETIMAGECORRECTION	23
5.53	设置鱼眼矫正类型 SETFECDISPLAYEFFECT	23
5.54	获取鱼眼矫正参数 GETFECDISPLAYPARAM	24
5.55	设置鱼眼矫正参数 SETFECDISPLAYPARAM	24
6	常见问题排查	26
	关于流模式接口	26
	Question 1 送入到播放库 PlayM4_OpenStream 接口报错	26
	流模式预览	26

Question 2 实时流预览有卡顿.....	26
Question 3 实时流预览不显示图像.....	26
Question 4 实时流预览延时.....	27
流模式播放.....	27
Question 5 流模式下快放.....	27
Question 6 实时流预览显示第一帧画面比较慢.....	28
关于加密码流.....	28
Question 7 加密码流无法正常播放.....	28
关于抓图.....	28
Question 8 如何将捕获的图片保存在内存中以供后续处理.....	28
关于播放声音问题.....	28
Question 9 音频播放卡顿，视频播放流畅.....	28
Question 10 复合流文件，播放音频时画面卡顿，不播放音频。则画面正常.....	28
Question 11 播放文件，无法听到声音.....	28
关于文件模式问题.....	29
Question 12 实时流播放正常，文件播放有快放现象.....	29
Question 13 文件打开失败.....	29
Question 14 文件模式如何定位.....	29
关于解码回调问题.....	29
Question 15 如何获取多路数据流解码后的数据.....	29
Question 16 实时流数据用解码回调有丢帧现象.....	30
Question 17 有些码流解码回调函数中获取的每帧视频高度只有显示的一半.....	30
Question 18 如何自己实现显示.....	30
Question 19 解码回调和抓图回调函数的区别.....	30
Question 20 解码回调函数中获取连续三帧音频数据时标信息重复.....	30
关于同步回放问题.....	30
Question 21 如何实现多个文件的同步回放.....	30
关于倍帧问题.....	31
Question 22 调用倍帧接口没有效果.....	31
关于 ANDROID 版本的播放库问题.....	31
Question 23 Android 版本的播放库不能正常预览.....	31

1 产品简介

播放库 SDK 是网络硬盘录像机、视频服务器、IP 设备等产品播放相关的二次开发包，适用于我司编码产品视音频数据的解码与播放。

播放库 SDK 主要功能：

实时码流预览，录像文件回放，播放控制如：暂停、单帧前进、单帧后退；获取码流基本信息，如文件索引、解码帧信息，分辨率、帧率；支持 JPG 和 BMP 两种形式下的播放截图等。

2 SDK 版本更新

版本号说明

播放库 SDK 版本号自 V7.0.0.6 起，规定版本号定义如下：

v 主版本号 . 子版本号 . 修正版本号 . 保留版本号

- 主版本号升级：工程作大规模改动、重构或优化
- 子版本号升级：功能增加
- 修正版本号升级：局部修改，bug 修正

Version 7.3.1.X

- 增加获取解码类型接口：[getDecoderType](#)。
- 增加设置最大硬解码路数接口：[setMaxHardDecodePort](#)。
- 增加设置同步回放组接口：[setSycGroup](#)。
- 修改默认显示节点为 6 个。
- 硬解码增加支持 H265、MPEG4 类型码流。
- 增加第 6 章常见问题排查。

Version 7.3.0.X

- 增加硬解码功能：[setHardDecode](#)。
- 增加广角图像矫正功能：[setImageCorrection](#)。
- 新增鱼眼矫正相关配置功能：
[setFECDisplayEffect](#)、
[getFECDisplayParam](#)、
[setFECDisplayParam](#)。
- 新增预录像功能：
[setPreRecordCallBack](#)
[setPreRecordFlag](#)。

Version 7.1.0.X

- 默认由按帧率播放变为按时间戳播放。
- 增加接口：
`setFileRefCB` 注册文件索引回调和 `SetDecodeCB` 注册解码回调。
- 增加接口：
`setVideoWindow` 不再有增加显示区域功能，相关功能由 `setDisplayRegion` 实现。
- 支持 Android 2.3 及以上系统（目前已经测试 android 系统版本 2.3~4.2）。

Version 7.0.2.0

- 支持 Android 2.2 及以上系统。

3 错误代码及说明

错误名称	代码	说明
PLAYM4_NOERROR	0	没有错误
PLAYM4_PARA_OVER	1	输入参数非法;
PLAYM4_ORDER_ERROR	2	调用顺序不对
PLAYM4_TIMER_ERROR	3	SDL 时钟设置失败
PLAYM4_DEC_VIDEO_ERROR	4	视频解码失败
PLAYM4_DEC_AUDIO_ERROR	5	音频解码失败
PLAYM4_ALLOC_MEMORY_ERROR	6	分配内存失败
PLAYM4_OPEN_FILE_ERROR	7	文件操作失败
PLAYM4_CREATE_OBJ_ERROR	8	创建线程事件等失败
PLAYM4_BUF_OVER	11	缓冲区满, 输入流失败
PLAYM4_CREATE_SOUND_ERROR	12	创建音频设备失败
PLAYM4_SET_VOLUME_ERROR	13	设置音量失败
PLAYM4_SUPPORT_FILE_ONLY	14	只能在播放文件时才能使用此接口
PLAYM4_SUPPORT_STREAM_ONLY	15	只能在播放流时才能使用此接口
PLAYM4_SYS_NOT_SUPPORT	16	不支持,
PLAYM4_FILEHEADER_UNKNOWN	17	没有文件头
PLAYM4_VERSION_INCORRECT	18	解码器和编码器版本不对应
PALYM4_INIT_DECODER_ERROR	19	初始化解码器失败
PLAYM4_CHECK_FILE_ERROR	20	文件太短或码流无法识别
PLAYM4_INIT_TIMER_ERROR	21	初始化 SDL 时钟失败
PLAYM4_BLT_ERROR	22	显示错误
PLAYM4_OPEN_FILE_ERROR_MULTI	24	打开文件错误,数据类型为混合流
PLAYM4_OPEN_FILE_ERROR_VIDEO	25	打开文件错误,数据类型为视频流
PLAYM4_JPEG_COMPRESS_ERROR	26	JPEG 压缩错误
PLAYM4_EXTRACT_NOT_SUPPORT	27	不支持此文件
PLAYM4_EXTRACT_DATA_ERROR	28	提取视频数据错误
PLAYM4_SECRET_KEY_ERROR	29	密码错误
PLAYM4_DECODE_KEYFRAME_ERROR	30	关键帧错误
PLAYM4_NEED_MORE_DATA	31	需要更多的数据
PLAYM4_INVALID_PORT	32	无效端口号
PLAYM4_FAIL_UNKNOWN	99	错误

[返回目录](#)

4 函数调用顺序

应用程序初始化

文件模式流程:

getPort
setFileEndCB
setFileRefCB
openFile
setDecodeCB
setDisplayCB
play
stop
closeFile
freePort

流模式流程:

getPort
setStreamOpenMode
openStream
setDisplayBuf
setDecodeCB
setDisplayCB
Play
inputData
stop
closeStream
freePort

鱼眼矫正流程:

getPort
openFile/openStream
play
setFECDisplayEffect
getFECDisplayParam
setFECDisplayParam
stop
closeFile/closeStream
freePort

应用程序结束

5 函数说明

系统操作及错误号获取

5.1 获取播放库版本号 **getSdkVersion**

函 数:	int getSdkVersion()	
参 数:		
返回值:	播放库当前版本号	
说 明:	高 16 位表示当前的 build 号。 9~16 位表示主版本号， 1~8 位表示次版本号。 如：返回值 0x06040105 表示：build 号是 0604，版本号 1.5。	
注 意:	如果只是修改 bug，我们只升级后面两位版本号。	

[返回目录](#)

5.2 获取错误码 **getLastError**

函 数:	int getLastError(int nPort)	
参 数:	int nPort	播放通道号
返回值:	错误码，详见 错误码宏定义	
说 明:	获得当前错误的错误码。用户在调用某个函数返回失败时，调用此函数获得错误的详细信息。	
注 意:		

[返回目录](#)

5.3 获取播放端口 **getPort**

函 数:	int getPort()	
参 数:		
返回值:	获取播放端口	
说 明:	获取未使用的通道号,通道号的上限为 16。	
注 意:		

[返回目录](#)

5.4 释放播放端口 **freePort**

函 数:	boolean freePort(int nPort)	
------	-----------------------------	--

参 数:	int nPort	播放端口
返回值:	成功返回 true; 失败返回 false	
说 明:	释放已使用的通道号, 释放成功后, 最好将 nPort 置为-1。	
注 意:		

[返回目录](#)

文件操作

5.5 打开数据文件 **openFile**

函 数:	boolean openFile(int nPort, String sFilePath)	
参 数:	int nPort	播放端口
	String sFilePath	文件路径
返回值:	成功返回 true; 失败返回 false	
说 明:		
注 意:	文件不能超过 4G 或小于 4K	

[返回目录](#)

5.6 关闭数据文件 **closeFile**

函 数:	boolean closeFile(int nPort)	
参 数:	int nPort	播放端口
返回值:	成功返回 true; 失败返回 false	
说 明:		
注 意:		

[返回目录](#)

流操作

5.7 设置流播放模式 **setStreamOpenMode**

函 数:	boolean setStreamOpenMode(int nPort, int nMode)	
参 数:	Int nPort	播放通道号
	unsigned int nMode	流播放模式: STREAME_REALTIME、STREAME_FILE
	STREAME_REALTIME	此模式(默认)下, 会尽量保证实时性, 防止数据阻塞, 而且数据检查严格。适合播放网络实时数据, 立刻解码
	STREAME_FILE	文件模式, 适合用户将文件以流方式输入, 按时间戳播放
返回值:	成功返回 true; 失败返回 false	
说 明:		
注 意:	STREAME_REALTIME 模式下 fast、slow 调用无效且不返回错误码。	

[返回目录](#)

5.8 打开流 **openStream**

函 数:	boolean openStream(int nPort, byte[] pFileHeadBuf, int nSize, int nBufPoolSize)		
参 数:	int	nPort	播放端口
	byte[]	pFileHeadBuf	文件头信息
	int	nSize	文件头长度
	int	nBufPoolSize	打开流分配缓存
	范围是 SOURCE_BUF_MIN~ SOURCE_BUF_MAX。该值过小会导致无法解码，建议标清设备该值大于等于 200*1024，高清设备大于等于 600*1024。		
	#define SOURCE_BUF_MAX	1024*100000	
	#define SOURCE_BUF_MIN	1024*50	
返回值:	成功返回 true；失败返回 false		
说 明:	当前播放库已经支持无头码流，文件头长度可以为零。若无头码流，码流必须为我司设备编码的码流，且从文件开头部分到能解析出正确的 I 帧的数据大小需小于 10K。		
注 意:			

[返回目录](#)

5.9 关闭流 **closeStream**

函 数:	boolean closeStream(int nPort)		
参 数:	Int	nPort	播放端口
返回值:	成功返回 true；失败返回 false		
说 明:			
注 意:			

[返回目录](#)

5.10 输入流数据 **inputData**

函 数:	boolean inputData(int nPort, byte[] pBuf, int nSize)		
参 数:	int	nPort	播放通道号
	byte[]	pBuf	流数据缓冲区地址
	int	nSize	流数据缓冲区大小
返回值:	true 表示已经输入数据。false 表示失败，数据没有输入		
说 明:	输入流数据，需要在开启流之后才能输入数据，返回 false，若错误码是 11 则是由于内部缓冲区满，建议客户暂停输入数据线程，再次输入数据，确保播放库不丢失数据。		
注 意:			

[返回目录](#)

播放控制

5.11 开启播放 **play**

函 数:	boolean play(int nPort, SurfaceHolder holder)	
参 数:	int nPort	播放通道号
	SurfaceHolder holder	显示窗口
返回值:	成功返回 true; 失败返回 false	
说 明:	播放开始, 播放视频画面大小将根据 holder 窗口调整, 要全屏显示, 只要把 holder 窗口放大到全屏。 如果已经播放, 重置当前播放速度为正常速度。	
注 意:		

[返回目录](#)

5.12 停止播放 **stop**

函 数:	boolean stop(int nPort)	
参 数:	int nPort	播放通道号
返回值:	成功返回 true; 失败返回 false	
说 明:		
注 意:		

[返回目录](#)

5.13 暂停数据播放 **pause**

函 数:	boolean pause(int nPort, int nPause)	
参 数:	int nPort	播放端口
	int nPause	1 暂停, 0 恢复
返回值:	成功返回 true; 失败返回 false	
说 明:		
注 意:		

[返回目录](#)

5.14 快速播放 **fast**

函 数:	boolean fast(int nPort)	
参 数:	int nPort	播放端口
返回值:	成功返回 true; 失败返回 false	
说 明:	每次调用将使当前播放速度加快一倍, 最多加速至 16 倍 注意: 实际能达到倍数取决与机器性能、视频的分辨率等多种因素。	
注 意:		

[返回目录](#)

5.15 慢速播放 **slow**

函 数:	boolean slow(int nPort)	
参 数:	int nPort	播放端口
返回值:	成功返回 true; 失败返回 false	
说 明:	慢速播放, 每次调用将使当前播放速度慢一倍; 最多减速至 1/16	
注 意:		

[返回目录](#)

5.16 以独占方式播放音频 **playSound**

函 数:	boolean playSound(int nPort)	
参 数:	int nPort	播放端口
返回值:	成功返回 true; 失败返回 false	
说 明:	<p>打开声音; 同一时刻只能有一路声音。</p> <p>如果现在已经有声音打开, 则自动关闭原来已经打开的声音。</p> <p>播放低帧率文件时, 必须先开启声音再播放, 否则可能造成声音播放不正常。</p> <p>如果 PlayM4_OpenStream 的时候 40 字节文件头未申明音频编码类型, PlayM4_PlaySound 会报错。</p>	
注 意:	默认情况下是不处理声音数据的, 不调用这个接口则不会解码音频数据	

[返回目录](#)

5.17 关闭音频 **stopSound**

函 数:	boolean stopSound()	
参 数:	无	
返回值:	成功返回 true; 失败返回 false	
说 明:	关闭最后一次使用 playSound 接口打开声音的播放通道。	
注 意:	playSound, stopSound 需要配对调用。	

[返回目录](#)

5.18 设置文件当前播放位置（百分比） **setPlayPos**

函 数:	boolean setPlayPos(int nPort,float fRelativePos)	
参 数:	int nPort	播放端口
	float fRelativePos	文件位置比例, 范围 0-100%
返回值:	成功返回 true; 失败返回 false	
说 明:	<p>设置文件播放指针的相对位置（百分比）。</p> <p>如果在建立文件索引的前提下使用, 为精确定位, 否则即为粗略定位;</p> <p>精确定位是定位到最近的一个视频帧, 粗略定位是位置之后的最近的一个关键帧。</p>	
注 意:		

[返回目录](#)

5.19 获取文件当前播放位置（百分比） **getPlayPos**

函 数:	float getPlayPos(int nPort)	
参 数:	int nPort	播放端口
返回值:	范围 0-100%	
说 明:	获得文件播放指针的相对位置	
注 意:		

[返回目录](#)

5.20 设置文件当前播放时间（毫秒） **setPlayedTimeEx**

函 数:	boolean setPlayedTimeEx(int nPort, int nTime)	
参 数:	int nPort	播放通道号
	int nTime	设置文件播放到指定时间，单位毫秒
返回值:	成功返回 true；失败返回 false	
说 明:	根据时间设置文件播放位置	
注 意:	如果在建立文件索引的前提下使用, 为精确定位, 否则即为粗略定位, 精确定位是定位到最近的一个视频帧, 粗略定位是位置之后的最近的一个关键帧	

[返回目录](#)

5.21 获取文件当前播放时间（毫秒） **getPlayedTimeEx**

函 数:	long getPlayedTimeEx (int nPort)	
参 数:	int nPort	播放通道号
返回值:	成功返回 true；失败返回 false	
说 明:	根据时间获取文件播放位置	
注 意:		

[返回目录](#)

5.22 设置文件当前播放帧号 **setCurrentFrameNum**

函 数:	boolean setCurrentFrameNum(int nPort, int nFrameNum)	
参 数:	int nPort	播放端口
	int nFrameNum	帧号
返回值:	成功返回 true；失败返回 false	
说 明:	设置当前播放播放位置到指定帧号；根据帧号来定位播放位置。	
注 意:	如果在建立文件索引的前提下使用, 为精确定位, 否则即为粗略定位, 若客户指定的帧号存在, 则精确定位是定位到指定的视频帧; 若不存在则是最近的视频帧; 而粗略定位是位置之后的最近的一个关键帧	

[返回目录](#)

5.23 获取文件当前播放帧号 `getCurrentFrameNum`

函 数:	Int getCurrentFrameNum(int nPort)	
参 数:	int nPort	播放端口
返回值:	当前播放的帧序号	
说 明:	得到当前播放的帧序号。	
注 意:		

[返回目录](#)

获取播放或解码信息

5.24 获取文件总时间 `getFileTime`

函 数:	long getFileTime(int nPort)	
参 数:	int nPort	播放端口
返回值:	文件总时间长度，单位秒	
说 明:		
注 意:	不支持对文件的数据追加（不支持对正在写入的文件进行时间的读取）	

[返回目录](#)

5.25 获取全局时间 `getSystemTime`

函 数:	Boolean getSystemTime(int nPort, MPSystemTime stSystemTime)	
参 数:	int nPort [out]MPSystemTime stSystemTime MPSystemTime 的结构体参数说明: public static class MPSystemTime { public int year; public int month; public int day; public int hour; public int min; public int sec; public int ms; }	播放端口 全局时间 年 月 日 时 分 秒 毫秒
返回值:	成功返回 true,失败返回 false	
说 明:		
注 意:		

[返回目录](#)

5.26 获取文件总帧数 **getFileTotalFrames**

函 数:	int getFileTotalFrames (int nPort)	
参 数:	int nPort	播放端口
返回值:	文件总帧数	
说 明:		
注 意:	不支持对文件的数据追加（不支持对正在写入的文件进行帧数的读取）。	

[返回目录](#)

5.27 获取当前帧率 **getCurrentFrameRate**

函 数:	int getCurrentFrameRate (int nPort)	
参 数:	int nPort	播放端口
返回值:	当前码流中当前编码帧率	
说 明:		
注 意:	如果码流中的帧率小于 1，则返回 0，表明该接口不支持	

[返回目录](#)

5.28 获取已解码的视频帧数 **getPlayedFrames**

函 数:	int getPlayedFrames(int nPort)	
参 数:	int nPort	播放通道号
返回值:	已经解码的视频帧数	
说 明:		
注 意:		

[返回目录](#)

5.29 获取原始图像大小 **getPictureSize**

函 数:	boolean getPictureSize(int nPort, MPInteger stWidth, MPInteger stHeight)	
参 数:	int nPort,	播放通道号
	[out] int * stWidth,	原始图像的宽度
	[out] int * stHeight	原始图像的高度
返回值:	成功返回 true；失败返回 false	
说 明:	获得码流中原始图像的大小，根据此大小来设置显示窗口的区域，可以不使用显卡做缩放工作，对于那些不支持硬件缩放的显卡来说非常有用。该函数获取的是当前播放显示的视频帧数据的宽高信息，因此正常预览显示后才能获取准确的值。 在尚未解码出第一帧图像前，返回默认值 352x288。 2CIF 分辨率码流实际返回分辨率为 4cif。	
注 意:		

[返回目录](#)

解码操作及控制

5.30 设置视频帧解码类型 **setDecodeFrameType**

函 数:	boolean setDecodeFrameType(int nPort, int nFrameType)	
参 数:	LONG nPort	播放通道号
	int nFrameType	解码帧类型说明
	<pre>#define DECODE_NORMAIL #define DECODE_KEY_FRAME #define DECODE_NONE</pre>	0 正常解码 1 只解关键帧 2 不解视频帧
返回值:	成功返回 true; 失败返回 false	
说 明:		

[返回目录](#)

5.31 设置解码回调函数 **setDecodeCB**

函 数:	boolean setDecodeCB (int nPort, PlayerDecodeCB decodeCB)	
参 数:	int nPort	播放通道号
	PlayerDecodeCB decodeCB	解码回调函数
	<pre>PlayerDecodeCB 回调函数 public void onDecode(int nPort, byte[] data, int nDataLen, int nWidth, int nHeight, int nFrameTime, int nDataType, int Reserved);</pre>	
	参数说明:	
	int nPort	播放器通道号
	byte[] data	解码后的音视频数据指针
	int nDataLen	解码后的音视频数据 pBuf 的长度
	int nWidth	视频图像宽/音频声道数
	int nHeight	视频图像高/音频采样率
	int nFrameTime	解码时间戳
	int nDataType	解码当前帧类型, 详见下文宏定义说明
	int Reserved	保留
	宏定义说明	
	T_AUDIO16 = 101;	音频数据;采样率 16khz, 单声道, 每个采样点 16 位表示。

	<p>T_AUDIO8 = 100;</p> <p>T_RGB32 = 7;</p> <p>T_UYVY = 1;</p> <p>T_YV12 = 3;</p>	<p>音频数据,采样率 8khz,单声道,每个采样点 16 位表示。</p> <p>视频数据。每个像素 4 个字节,排列方式与位图相似,“B-G-R-0 ...”,第一个像素位于图像左下角。</p> <p>视频数据,uyvy 格式。</p> <p>“U0-Y0-V0-Y1-U2-Y2-V2-Y3....”,第一个像素位于图像左上角。</p> <p>视频数据,yv12 格式。排列顺序“Y0-Y1-.....”,“V0-V1....”,“U0-U1-.....”。</p> <p>关于图像格式详细信息,用户可以查阅相关资料。</p>
返回值:	成功返回 true; 失败返回 false	
说 明:	<p>设置回调函数,替换播放器中的显示部分,由用户自己控制显示,该函数在 Play 之前调用,在 Stop 时自动失效,下次调用 Play 之前需要重新设置。</p> <p>解码部分不控制速度,只要用户从回调函数中返回,解码器就会解码下一部分数据。这个功能的使用需要用户对视频显示和声音播放有足够的了解,否则请慎重使用。</p> <p>当前解码的视频数据格式为 YV12,音频解码格式为 PCM。</p> <p>调用此回调函数后,则视频图像无显示。</p>	

[返回目录](#)

5.32 文件结束回调 setFileEndCB

函 数:	boolean setFileEndCB(int nPort, PlayerPlayEndCB playEndCB)	
参 数:	<p>int nPort</p> <p>PlayerPlayEndCB playEndCB</p> <p>回调函数 PlayerPlayEndCB</p> <p>public void onPlayEnd(int nPort);</p> <p>参数说明</p> <p>int nPort</p>	<p>播放通道号</p> <p>文件结束回调接口对象</p> <p>播放库通道号</p>
返回值:	成功返回 true; 失败返回 false	
说 明:	设置文件播放结束回调函数。在 openFile 之前调用才有效	
注 意:		

[返回目录](#)

5.33 解码密钥 setSecretKey

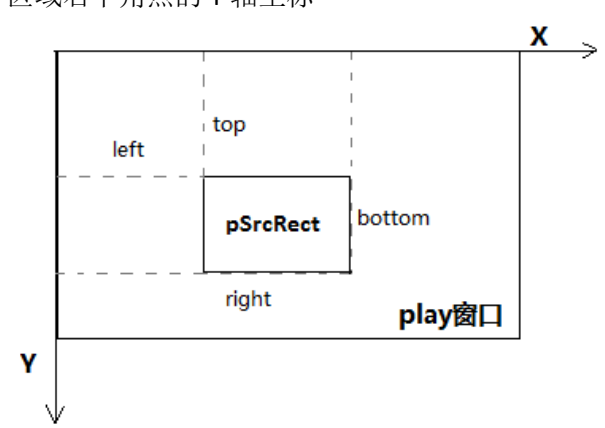
函 数:	boolean setSecretKey(int nPort, int nKeyType, byte[] pSecretKey, int nKeyLen)	
参 数:	<p>int nPort</p> <p>int lKeyType</p> <p>byte[] pSecretKey</p> <p>int lKeyLen</p>	<p>播放通道号</p> <p>密钥类型(取值为 0 不加密;取值为 1 标识 AES 加密)</p> <p>密钥串首地址</p> <p>密钥长度,单位为 bit(1 字节为 8 bit)</p>

返回值:	成功返回 true ；失败返回 false
说 明:	如果在编码时设置了密钥，那么在解码之前需要调用该接口设置密钥才能正常解码。该接口在 openSteam/openFile 之前调用才有效。
注 意:	

[返回目录](#)

显示操作

5.34 设置或增加显示区域 **setDisplayRegion**

函 数:	boolean setDisplayRegion(int nPort, int nRegionNum, MPRect pSrcRect, Surface hDestWnd, int bEnable)	
参 数:	<div><div>int nPort</div><div>int nRegionNum</div><div>[out]MPRect pSrcRect</div><div>Surface hDestWnd</div><div>int bEnable</div></div>	<div>播放通道号</div> <div>显示区域序号，0~(MAX_DISPLAY_WND-1)</div> <div>设置在要显示的原始图像上的区域，如：如果原始图像是 352*288，那么 pSrcRect 可设置的范围只能在(0, 0, 352, 288)之中。如果 pSrcRect=NULL,将显示整个图像</div> <div>设置显示窗口。如果 hDestWnd 同 play 接口的窗口句柄，区域选择的图像直接在该窗口上显示，源视频不再显示；如果 hDestWnd 传入新的窗口句柄，则原窗口显示图像不变，区域选择的图像在该句柄对应的窗口上显示。</div> <div>打开（设置）或关闭显示区域</div> <div>MPRect 结构体参数说明： public static class MPRect { public int left; public int right; public int top; public int bottom; }</div> <div>区域左上角点的 X 轴坐标 区域右下角点的 X 轴坐标 区域左上角点的 Y 轴坐标 区域右下角点的 Y 轴坐标</div> <div></div>

返回值:	成功返回 true ；失败返回 false
说 明:	设置或增加显示区域，可以做局部放大显示。 该接口在正常播放状态下调用，设置或增加显示区域，可以做局部放大显示。
注 意:	nRegionNum为0时，表示对主窗口进行处理，将忽略hDestWnd和bEnable的设置。 电子放大区域范围（pSrcRect）取值必须大于0，right和bottom的取值必须大于16，如果取值异常则显示原图像大小，不做放大处理。

[返回目录](#)

5.35 设置或销毁显示区域 **setVideoWindow**

函 数:	boolean setVideoWindow(int nPort, int nRegionNum, SurfaceHolder holder)		
参 数:	int	nPort	播放库端口
	int	nRegionNum	显示区域序号
	SurfaceHolder	holder	设置显示窗口
返回值:	成功返回 true ，失败返回 false		
说 明:	设置或者销毁显示区域。用于切换窗口，需要先调用该接口将原显示窗口设置为 null，然后再调用接口设置新的显示窗口。		
注 意:	最后一个参数 SurfaceHolder 拥有生命周期，在其中两个生命周期回调中： SurfaceCreated 中，调用 setVideoWindow(nPort, nRegionNum, holder) 创建 SurfaceHolder； SurfaceDestroyed 中，调用 setVideoWindow(nPort, nRegionNum, null)销毁 SurfaceHolder。		

[返回目录](#)

5.36 设置同步回放组 **setSycGroup**

函 数:	public boolean setSycGroup(int nPort, int nGroupIndex)		
参 数:	int	nPort	播放库端口
	int	nGroupIndex	同步回放分组号，取值在 0~4 之间，硬解码不支持。
返回值:	成功返回 true ，失败返回 false		
说 明:	可以多次调用将指定的 nPort 加入到同步组中； setSycGroup 在 play 之前调用； nGroupIndex 一个序号被视为一个同步组，每个同步组最多 16 个播放通道，若超过 16 给播放通道则返回失败，且必须 freePort 此播放通道才能退出同步组； 支持文件模式和文件流模式同步回放，同步回放需要全局时间，因此只支持我司设备码流。硬解码不支持同步。		
注 意:			

[返回目录](#)

缓冲区操作

解码前源缓冲区

5.37 获取源缓冲区剩余数据大小 **getSourceBufferRemain**

函 数:	int getSourceBufferRemain(int nPort)	
参 数:	int nPort	播放通道号
返回值:	当前源缓冲的大小 (BYTE)	
说 明:	获取数据缓冲 (对应 openStream 接口配置的 nPoolBufSize 参数) 中的剩余数据	
注 意:		

[返回目录](#)

5.38 清空所有缓冲区的剩余数据 **resetSourceBuf**

函 数:	boolean resetSourceBuf(int nPort)	
参 数:	int nPort	播放端口
	int nBufType	缓冲区类型
返回值:	成功返回 true, 失败返回 false	
说 明:	清空所有缓冲区的剩余数据	
注 意:		

[返回目录](#)

5.39 清空指定缓冲区的数据 **resetBuffer**

函 数:	boolean resetBuffer(int nPort, int nBufType)	
参 数:	int nPort	播放端口
	int nBufType	缓冲区类型
	缓冲区宏定义	
	BUF_VIDEO_SRC	视频数据源缓冲区，缓冲解码之前视频数据，只对流模式有效，单位 byte。
	BUF_AUDIO_SRC	音频数据源缓冲区，缓冲解码之前音频数据，只对流模式有效，单位 byte。且音视频数据分开送到情况下才有效
	BUF_VIDEO_RENDER	解码后视频缓冲区总节点个数，以帧为单位。 解码后音频缓冲区总节点个数，以帧为单位，音频 40ms 数据定为一帧。
	BUF_AUDIO_RENDER	
返回值:	成功返回 true,失败返回 false	
说 明:	清空指定缓冲区的剩余数据	
注 意:	在清空目标缓存数据时会同时清空显示缓存	

[返回目录](#)

解码后播放缓冲区

5.40 设置播放缓冲区最大缓冲帧数 `setDisplayBuf`

函 数:	boolean setDisplayBuf(int nPort, int nNum)	
参 数:	int nPort	播放通道号
	unsigned int nNum	播放缓冲区最大缓冲帧数。范围：MIN_DIS_FRAMES ~MAX_DIS_FRAMES。一帧 352*288 图像的所需内存最小值是 352*288*3/2 大约 150K。最大值是 352*288*4 大约 405K。
返回值:	成功返回 true；失败返回 false	
说 明:	<p>这个缓冲区比较重要，可以直接影响播放的流畅性和延时性。在一定范围内缓冲越大越流畅，同时延时越大。</p> <p>nNum 为 6 表示自适应模式，还支持设置 1、15 帧的缓冲区大小；其他值无效，默认为自适应模式</p> <p>在播放文件时用户最好可以考虑开大缓冲（如果内存足够大），我们的默认值是 6（帧），在 25 帧/秒的情况下即 0.24 秒的数据。</p>	
注 意:	通过设置不同的缓冲区帧数，控制流畅性和实时性。该接口需要在 <code>openStream</code> 之后， <code>play</code> 之前调用。	

[返回目录](#)

5.41 获取播放缓冲区最大缓冲帧数 `getDisplayBuf`

函 数:	int getDisplayBuf(int nPort)	
参 数:	int nPort	播放通道号
返回值:	播放缓冲区最大缓冲帧数	
说 明:	获取播放缓冲区帧数	
注 意:		

[返回目录](#)

源、播放缓冲区

5.42 获取指定缓冲区的大小 `getBufferValue`

函 数:	long getBufferValue (int nPort, int nBufType)	
参 数:	int nPort	播放通道号
	unsigned int nBufType	缓冲区类型
	缓冲区宏定义	
	BUF_VIDEO_SRC	视频数据源缓冲区大小，只对流模式有效，单位 byte。
	BUF_AUDIO_SRC	音频数据源缓冲区大小，只对流模式有效，单位 byte。

	BUF_VIDEO_RENDER BUF_AUDIO_RENDER	且音视频数据分开送到情况下才有效 解码后视频缓冲区总节点个数，以帧为单位。 解码后音频缓冲区总节点个数，以帧为单位。
返回值:	根据参数不同，返回缓冲区值，源缓冲区返回 byte,解码后缓冲区返回帧数。	
说 明:	获取播放器中的缓冲区大小（帧数或者 byte）。BUF_VIDEO_RENDER 可在网络延时方面有所估计。	
注 意:		

[返回目录](#)

索引

5.43 索引回调 **setFileRefCB**

函 数:	boolean setFileRefCB(int nPort, PlayerFileRefCB fileRefCB)	
参 数:	int nPort PlayerFileRefCB fileRefCB 回调函数 PlayerFileRefCB public void onFileRefDone(int nPort); 参数说明: int nPort	播放通道号 抓图回调函数，可以为 NULL 通道号
返回值:	成功返回 true；失败返回 false	
说 明:	建立文件索引回调。 为了能在文件中准确快速的定位，在文件打开的时候生成文件索引。这个过程耗时比较长。建立索引的过程是在后台完成，需要使用索引的函数要等待这个过程结束，而其他接口不会受到影响。 打开文件时是否建立文件的关键帧索引, 如果索引回调未触发, 表明录像文件异常。	
注 意:		

[返回目录](#)

抓图

5.44 抓图回调 **setDisplayCB**

函 数:	boolean setDisplayCB(int nPort, PlayerDisplayCB displayCB)	
参 数:	int nPort PlayerDisplayCB displayCB 回调函数 PlayerDisplayCB	播放通道号 抓图回调函数，可以为 NULL

	<pre>public void onDisplay(int nPort, byte[] data, int nDataLen, int nWidth, int nHeight, int nFrameTime, int nDataType, int Reserved);</pre> <p>参数说明:</p> <table> <tr> <td>int</td><td>nPort</td><td>通道号</td></tr> <tr> <td>byte[]</td><td>data</td><td>返回图像数据指针</td></tr> <tr> <td>int</td><td>nDataLen</td><td>返回图像数据大小</td></tr> <tr> <td>int</td><td>nWidth,</td><td>画面宽, 单位像素</td></tr> <tr> <td>int</td><td>nHeight</td><td>画面高</td></tr> <tr> <td>int</td><td>nFrameTime</td><td>时标信息, 单位毫秒</td></tr> <tr> <td>int</td><td>nDataType</td><td>数据类型, T_YV12, T_RGB32, T_UYVY,</td></tr> <tr> <td>int</td><td>Reserved</td><td>用户数据</td></tr> </table>	int	nPort	通道号	byte[]	data	返回图像数据指针	int	nDataLen	返回图像数据大小	int	nWidth,	画面宽, 单位像素	int	nHeight	画面高	int	nFrameTime	时标信息, 单位毫秒	int	nDataType	数据类型, T_YV12, T_RGB32, T_UYVY,	int	Reserved	用户数据	
int	nPort	通道号																								
byte[]	data	返回图像数据指针																								
int	nDataLen	返回图像数据大小																								
int	nWidth,	画面宽, 单位像素																								
int	nHeight	画面高																								
int	nFrameTime	时标信息, 单位毫秒																								
int	nDataType	数据类型, T_YV12, T_RGB32, T_UYVY,																								
int	Reserved	用户数据																								
返回值:	成功返回 true; 失败返回 false																									
说 明:	设置抓图回调函数; 注意要尽快返回, 如果要停止回调, 可以把回调函数指针 displayCB 设为 NULL。一旦设置回调函数, 则一直有效, 直到 closeStream。该函数可以在任何时候调用 回调在时钟线程中触发, 不能出现耗时操作, 否则会打乱时钟脉冲, 影响显示																									
注 意:																										

[返回目录](#)

5.45 截取当前画面 BMP 图片 getBMP

函 数:	boolean getBMP(int nPort, byte[] pBmp, int nBufSize, MPInteger pBmpSize)													
参 数:	<table> <tr> <td>int</td><td>nPort</td><td>播放端口</td></tr> <tr> <td>unsigned char*</td><td>pBitmap</td><td>存放 BMP 图像数据地址, 由用户分配, 不得小于 bmp 图像大小, 即 sizeof(BITMAPFILEHEADER) + sizeof(BITMAPINFOHEADER) + w * h * 4, 其中 w 和 h 分别为图像宽高。</td></tr> <tr> <td>unsigned int</td><td>nBufSize</td><td>申请的缓冲区大小</td></tr> <tr> <td>unsigned int *</td><td>pBmpSize</td><td>获取到的实际 bmp 图像大小</td></tr> </table>	int	nPort	播放端口	unsigned char*	pBitmap	存放 BMP 图像数据地址, 由用户分配, 不得小于 bmp 图像大小, 即 sizeof(BITMAPFILEHEADER) + sizeof(BITMAPINFOHEADER) + w * h * 4, 其中 w 和 h 分别为图像宽高。	unsigned int	nBufSize	申请的缓冲区大小	unsigned int *	pBmpSize	获取到的实际 bmp 图像大小	
int	nPort	播放端口												
unsigned char*	pBitmap	存放 BMP 图像数据地址, 由用户分配, 不得小于 bmp 图像大小, 即 sizeof(BITMAPFILEHEADER) + sizeof(BITMAPINFOHEADER) + w * h * 4, 其中 w 和 h 分别为图像宽高。												
unsigned int	nBufSize	申请的缓冲区大小												
unsigned int *	pBmpSize	获取到的实际 bmp 图像大小												
返回值:	成功返回 true; 失败返回 false													
说 明:														
注 意:	2CIF 分辨率的图像抓图后的分辨率为 4CIF。													

[返回目录](#)

5.46 截取当前画面 JPEG 图片 getJPEG

函 数:	boolean getJPEG(int nPort, byte[] pJpeg, int nBufSize, MPInteger pJpegSize)							
参 数:	<table> <tr> <td>int</td><td>nPort</td><td>播放通道号</td></tr> <tr> <td>byte[]</td><td>pJpeg</td><td>jpeg 图片数据缓存, 由用户分配, 因为无法预算出编码</td></tr> </table>	int	nPort	播放通道号	byte[]	pJpeg	jpeg 图片数据缓存, 由用户分配, 因为无法预算出编码	
int	nPort	播放通道号						
byte[]	pJpeg	jpeg 图片数据缓存, 由用户分配, 因为无法预算出编码						

	unsigned int nBufSize unsigned int * pJpegSize	后的数据大小，建议申请大小为 $w * h * 3/2$ ，其中 w 和 h 分别为图像宽高。 申请的缓冲区大小 获取到的实际 JPEG 图像数据大小
返回值：	成功返回 true；失败返回 false	
说 明：		
注 意：	2CIF 分辨率的图像抓图后的分辨率为 4CIF。	

[返回目录](#)

硬解码

5.47 设置硬解码优先 **setHardDecode**

函 数：	boolean setHardDecode(int nPort, int bEnable)	
参 数：	int nPort	播放通道号
	int bEnable	打开（设置）或关闭硬解码优先：0- 禁用，1- 启用
返回值：	成功返回 true；失败返回 false	
说 明：	设置硬解码优先后，符合硬解码的码流优先进行硬解码，不符合的会自动切换至软解，接口需要在 openStream 之后、play 之前调用。 另：开启硬解码后，在 stop 之后，close 之前一定要重新调用 setHardDecode(port, 0) 来关闭硬解码，否则会导致后续硬解码无法打开。 硬解码播放关闭后最后帧画面不会消除，需要用户自行处理。	
注 意：	Android 系统版本需要在 4.1（API16）及以上。 硬解码支持 H264、H265、MPEG4 类型码流的文件/实时流，支持主流分辨率（CIF、4CIF、720P、VGA、1080P、UXGA 等）。 启用硬解码时不支持以下功能接口： getJPG、getBMP、setImageCorrection、getPictureSize、setDisplayRegion、setVideoWindow、getDisplayBuf、setDisplayBuf、setFECDisplayEffect、setFECDisplayParam、resetBuffer(RENDER_BUFFER)。 硬解码不支持同步回放。	

[返回目录](#)

5.48 设置最大硬解码路数 **setMaxHardDecodePort**

函 数：	public boolean setMaxHardDecodePort(int nCount)	
参 数：	int nCount	最大硬解码路数
返回值：	成功返回 true；失败返回 false	
说 明：	最大硬解码路数取值范围需在 0~16 之间。该接口需要在开启硬解码 setHardDecode 之前调用。	
注 意：		

[返回目录](#)

5.49 获取当前解码类型 **getDecoderType**

函 数:	public int getDecoderType(int nPort)	
参 数:	int nPort	播放通道号
返回值:	返回 1-硬解码 0-软解码	
说 明:		
注 意:		

[返回目录](#)

预录像功能

5.50 设置预录像数据回调 **setPreRecordCallBack**

函 数:	boolean setPreRecordCallBack(int nPort, PlayerPreRecordCB PreRecordCB)	
参 数:	int nPort PlayerPreRecordCB PreRecordCB PlayerPreRecordCB 回调函数 public void onPreRecord(int nPort, byte[] data, int nDataLen) 回调函数参数说明 int nPort byte[] data int nDataLen	播放通道号 数据回调函数 用户指针 播放通道号 音视频数据 数据长度
返回值:	成功返回 1；失败返回 0	
说 明:	需要在play之后，正常显示第一帧之后调用。 回调函数中不能调用参数中带nPort的播放库接口，否则有可能造成死锁。	
注 意:	预录像回调中，回调出来的数据需要及时処理，建议不要每次回调都进行 IO 操作，而是先把回调的数据缓存起来，然后在回调的数据累积到一定大小（例如 2MB）后，再进行写目标文件。因为若不及时處理回调的数据或频繁进行 IO 等耗时操作，可能导致录像跟不上内部解析的速度，使录像文件存在丢帧的风险。	

[返回目录](#)

5.51 设置预录像开关 **setPreRecordFlag**

函 数:	boolean setPreRecordFlag(int nPort, int bFlag)	
参 数:	int nPort bool bFlag	播放通道号 设置预录像开关：1 表示开启，0 表示

		关闭
返回值:	成功返回 1；失败返回 0	
说 明:	需要在 play 之后，正常显示第一帧之后调用。 预录功能是把设备通过网络传输过来的码流转换为 PS 封装，然后通过回调函数返回给用户，方便用户保存成文件查看。	
注 意:	因缓存原因录像会有 2~3 秒延时	

[返回目录](#)

鱼眼功能

5.52 广角图像矫正 **setImageCorrection**

函 数:	boolean setImageCorrection(int nPort, int bEnable)	
参 数:	int nPort	播放通道号
	int bEnable	打开（设置）或关闭广角矫正：0- 关闭，1- 开启
返回值:	成功返回 true；失败返回 false	
说 明:		
注 意:	120°广角图像矫正。	

[返回目录](#)

5.53 设置鱼眼矫正类型 **setFECDisplayEffect**

函 数:	boolean setFECDisplayEffect(int nPort, int nRegionNum, MPVR_DISPLAY_EFFECT enDisplayEffect)	
参 数:	int nPort	播放通道号；
	int nRegionNum	窗口编号；
	MPVR_DISPLAY_EFFECT enDisplayEffect	鱼眼矫正类型
	MPVR_DISPLAY_EFFECT 枚举类型说明	
	VR_ET_NULL = 0,	不矫正
	VR_ET_FISH_PTZ_CEILING = 1,	应用于顶装鱼眼
	VR_ET_FISH_PTZ_FLOOR = 2,	应用于地面安装鱼眼
	VR_ET_FISH_PTZ_WALL = 3,	应用于壁装鱼眼
	VR_ET_FISH_PANORAMA_CEILING360 = 4,	应用于顶装鱼眼 1P
	VR_ET_FISH_PANORAMA_CEILING180 = 5,	应用于顶装鱼眼 2P
	VR_ET_FISH_PANORAMA_FLOOR360 = 6,	应用于地面安装鱼眼 1P
	VR_ET_FISH_PANORAMA_FLOOR180 = 7,	应用于地面安装鱼眼 2P
	VR_ET_FISH_LATITUDE_WALL = 8,	应用于壁装维度展开(广角)
	VR_ET_REDBLUE_3D = 9	红蓝 3D 效果
返回值:	成功返回 true；失败返回 false	
说 明:	鱼眼全景图像矫正后播放画面为正常图像。接口在 play 之后调用。	
注 意:	鱼眼矫正必须在已启用播放的窗口上进行，可以通过 setDisplayRegion 、 setVideoWindow 等接口设定新窗口后再进行鱼眼类型矫正(注意鱼眼矫正总是	

	对原始画面进行矫正)。 此版本不支持红蓝 3D 效果。
--	--------------------------------

[返回目录](#)

5.54 获取鱼眼矫正参数 **getFECDisplayParam**

函 数:	boolean getFECDisplayParam(int nPort, int nRegionNum, MPVR_FISH_PARAM stFishParam)	
参 数:	int nPort int nRegionNum MPVR_FISH_PARAM stFishParam MPVR_FISH_PARAM 结构体说明: public static class MPVR_FISH_PARAM{ public float xLeft; public float xRight; public float yTop; public float yBottom; public float angle; public float zoom; public float PTZX; public float PTZY; }	播放通道号; 窗口编号; 鱼眼矫正参数 最左边 X 坐标(min) 最右边 X 坐标(max) 最上边 Y 坐标(min) 最下边 Y 坐标(max) 180°矫正中心角 PTZ 矫正放大系数 PTZ 矫正中心 X 坐标 PTZ 矫正中心 Y 坐标
返回值:	成功返回 true; 失败返回 false	
说 明:	接口在 play 之后调用。	
注 意:		

[返回目录](#)

5.55 设置鱼眼矫正参数 **setFECDisplayParam**

函 数:	boolean setFECDisplayParam(int nPort, int nRegionNum, MPVR_FISH_PARAM stFishParam)	
参 数:	int nPort int nRegionNum MPVR_FISH_PARAM stFishParam MPVR_FISH_PARAM 结构体说明: public static class MPVR_FISH_PARAM{ public float xLeft; public float xRight; public float yTop; public float yBottom; public float angle; public float zoom; }	播放通道号; 窗口编号; 鱼眼矫正参数 最左边 X 坐标(min), 取值范围: [0.0, 1.0] 最右边 X 坐标(max), 取值范围: [0.0, 1.0] 最上边 Y 坐标(min), 取值范围: [0.0, 1.0] 最下边 Y 坐标(max), 取值范围: [0.0, 1.0] 180°矫正中心角, 取值范围: [0.0, 360.0] PTZ 矫正放大系数, 取值范围: (0.0, 1.0)

	<pre>public float PTZX; public float PTZY; }</pre>	<p>PTZ 矫正中心 X 坐标, 取值范围: [0.0, 1.0]</p> <p>PTZ 矫正中心 Y 坐标, 取值范围: [0.0, 1.0]</p>
返回值:	成功返回 true; 失败返回 false	
说 明:	<p>接口需要设置具体鱼眼效果后才可调用。</p> <p>参数需要满足以下条件:</p> <p>xLeft < xRight;</p> <p>yTop < yBottom;</p> <p>PTZ 矫正中心坐标需要位于点 (0.5, 0.5) 为圆心和半径为 0.4 的圆内。</p>	
注 意:		

[返回目录](#)

6 常见问题排查

关于流模式接口

Question 1 送入到播放库 PlayM4_OpenStream 接口报错

Answer 1

1. 检查参数是否正确，调用接口 [openStream](#)，需要确认送入正确的文件头（通过网络 SDK 中的实时流回调函数中获取的 NET_DVR_SYSHEAD，或者板卡中编码回调函数中获取的 PktSysHeader 类型的即为正确的文件头内容以及文件头长度）；
2. 调用失败后，若返回错误码是 PLAYM4_ALLOC_MEMORY_ERROR，说明内存不足，可以查看手机当前内存情况使用情况排查；
3. 此路流数据操作结束后，可以调用 [closeStream](#) 接口关闭流回收系统资源；closeStream 和 openStream 需要配对出现。

流模式预览

Question 2 实时流预览有卡顿

Answer 2

1. 检查数据是否流畅，若数据不流畅则检查网络部分；
2. 由于不同的分辨率对手机的性能要求不一样，如高清分辨率对手机性能要求比较高，多路高清实时预览时可能会有卡顿现象，此时可以适当降低预览的路数；
3. 检查视频当前的分辨率，缓冲区若开的太小，则可能送入数据（[inputData](#)）到播放库缓冲区满，此时需要重复送入未送成功的数据；

建议：若为 4CIF 分辨率，则建议 [openStream](#) 缓冲开到 1.5M 到 2M，1.5M 即 $1.5 * 1024 * 1024$ 。若 720P 则建议开到 3~4M；若 1080P 则建议开大到 6M 左右；

4. 建议检查 [setDisplayBuf](#)，设置为 15 个节点即可；
5. 可以将数据保存下来用播放器验证，若播放卡顿，则确认在网络部分是否丢数据，若播放正常则检查送数据部分是否正常。

Question 3 实时流预览不显示图像

Answer 3

1. 确认是否有正确的数据送入到播放库缓冲区，且调用过 play 开始解码；

2. 确认是否注册了解码回调不显示的回调函数 [setDecodeCB](#);
3. 确认是否可以正常抓取到图片; 若不能抓图成功则转 1, 若能抓图则转 4;
4. 抓图成功, 则连续抓图, 若图片随场景和 OSD 有变化, 则表示数据正常, 则检查显示的窗口句柄是否正确;
5. 可以将需要解码的数据保存为本地文件, 然后用播放器进行播放验证, 若能播放则转步骤 6, 若不能播放则转步骤 7;
6. 若可以正常播放, 则检查 [inputData](#) 送数据部分是否正确;
7. 若不能正常播放, 则检查数据码流是否正确, 若是标准码流则可以用 Elecard 或者 VLC 等标准第三方解码器进行验证; 若第三方的解码器可以解码则可以将测试码流发给我们分析; 若第三方的解码器不能解码则检查是否有网络丢包的现象;
8. 如果实时码流是 RTP 封装的码流, 则最好提供网络抓包文件, 或者如下格式: 4 字节 RTP 包的长度+RTP 数据包保存, 然后发给我们确认。

Question 4 实时流预览延时

Answer 4

1. 一般设备延时在 100 到 200ms 之间, 延时和网络的流畅性有很大关系。如在全帧率下延时正常, 而在低帧率情况下比较明显, 可以反馈给我们确认。
2. 如果是多路预览才出现延时的现象, 则请查看当前 CPU 的使用情况, 可能是 CPU 已经达到极限 (85%以上), 无法支持这么多路的实时流解码, 可以尝试减少几路播放, 或者提高硬件性能;
3. 若客户使用的是客户端软件, 则可以设置为流畅性好, 若是调用 SDK 进行二次开发, 则可以调用接口 [setDisplayBuf](#) 设置解码后缓冲区帧数;
4. 确定接口 [setStreamOpenMode](#) 设定为 STREAM_REALTIME;
5. 确定网络良好, 收发网络包延时较小。

流模式播放

Question 5 流模式下快放

Answer 5

流模式下支持码流的快放, 但是对当前送入的码流数据和网络环境是有要求的, 比如若正常情况下, 视频帧率为 25 帧的码流 1 倍速播放, 若需要 2 倍速快放, 则需要数据量是 1 倍速播放的 2 倍才能达到需求。

Question 6 实时流预览显示第一帧画面比较慢**Answer 6**

在低帧率下比较常见, 检查第一帧是否 I 帧, 若第一帧不是 I 帧则需要到 I 帧才能正常解码, 所以时间可能比较长。

关于加密码流**Question 7 加密码流无法正常播放****Answer 7**

1. 确认码流是否是基线的码流, 若是定制的码流则需要联系分公司确认加密方式和订单号;
2. 确认密钥和调用设置密钥接口被正确调用, 且密钥正确 [setSecretKey](#);
3. 可以将码流发给我们确认。

关于抓图**Question 8 如何将捕获的图片保存在内存中以供后续处理****Answer 8**

在开始正常解码播放 play 以后, 可以通过播放库函数 [getBMP](#) 和 [getJPEG](#) 把图片保存在内存中。

关于播放声音问题**Question 9 音频播放卡顿, 视频播放流畅****Answer 9**

确认音频数据是否均匀。

Question 10 复合流文件, 播放音频时画面卡顿, 不播放音频。则画面正常**Answer 10**

一般可能存在于低帧率文件, 或者音频分布不均匀导致。

Question 11 播放文件, 无法听到声音**Answer 11**

1. 需要确定码流中是否有音频数据, 音频源是有声音的。

2. 音频封装相关表示类型的字段不符合规定,如 RTP 封装 payload 错误,PS 封装 stream type 错误等。

3. 音频未加密而在封装层为加密,在播放库 6.2 系列版本后开始关注音频加密字段。如果之前版本可以正常播放,之后版本不行,则可能是设备问题,需要和设备确认。

关于文件模式问题

Question 12 实时流播放正常,文件播放有快放现象

Answer 12

确认码流中帧率或者时间戳是是否正常;另外可以检查是否有被抽帧的情况。

Question 13 文件打开失败

Answer 13

检查保存文件时是否有保存正确的文件头(设备网络 SDK 的实时流回调函数中码流类型为 NET_DVR_SYSHEAD,或者编码卡编码回调函数中当前帧类型为 PktSysHeader);[文件是否小于 4K 或者大于 4G](#);或者数据错误;或者没有关键帧(视频解码需要依赖于关键帧)。

Question 14 文件模式如何定位

Answer 14

是否成功创建文件索引;若索引创建成功,检查设定的定位参数(如时间或者帧号)是否正确;

关于解码回调问题

Question 15 如何获取多路数据流解码后的数据

Answer 15

1. 一路数据对应一个播放库 Port (通过 [getPort](#) 获取);
2. 调用 [setStreamOpenMode](#) 设定正确的播放模式;
3. 送入这路数据对应的文件头 [openStream](#);
4. 调用 [setDecodeCB](#) 注册解码回调函数,获取解码后的数据;
5. 调用 play 开始解码线程(若不需要显示图像,则可以将窗口句柄置为 NULL);
6. 调用 [inputData](#) 送入每路数据正确的数据;播放库支持多路并发解码,当前版本最多支持 16 路。

Question 16 实时流数据用解码回调有丢帧现象**Answer 16**

1. 实时录文件，先排除数据流是否有问题。
2. 如果数据没有问题，注释掉解码回调里面所有的实现，查看是否还有丢帧情况。解码回调里面实现耗时比较多的操作，也有可能出现此类的问题。
3. 请查看在实际调用过程中，检查 [inputData](#) 是否有失败，若有失败，请尝试将数据重新送入，错误发生频率比较高，建议 [openStream](#) 中设置缓冲区大小设置大一些。

Question 17 有些码流解码回调函数中获取的每帧视频高度只有显示的一半**Answer 17**

若原始图像分辨率是 2CIF（PAL 制为 704*288，NTSC 制为 704*240），实际显示是按照 PAL 制 704*576，NTSC 制是 704*480 进行显示，而解码回调函数中获取的是码流实际的视频分辨率，所以会出现这种情况。

Question 18 如何自己实现显示**Answer 18**

通过解码回调函数 [setDecodeCB](#) 回调函数获取到解码后的视频和音频数据后，再使用 OpenGL ES 实现渲染，音频用 AudioTrack。

Question 19 解码回调和抓图回调函数的区别**Answer 19**

[setDecodeCB](#) 解码以后就会将数据回调给用户，解码多少帧数据即回调多少帧，而 [setDisplayCB](#) 是将解码后的视频数据送入到显卡中显示的数据回调给用户。则是按照一个播放的帧率来回调的，比如帧率为 25 帧/s，回调的时候，就会间隔大概 40ms 回调一帧数据。

Question 20 解码回调函数中获取连续三帧音频数据时标信息重复**Answer 20**

这是播放库版本的问题，需要将播放库版本更新到最新版本。若需要得到准确的音频帧信息建议使用码流分析库 SDK。

关于同步回放问题**Question 21 如何实现多个文件的同步回放****Answer 21**

1. 多个同步回放的文件必须有相同的时间戳片段；

2. 在同步回放接口调用之前必须确认是否每个文件都有成功创建文档的索引；
3. 同步回放的过程中可以通过定位接口 [setPlayedTimeEx](#) 进行同步回放的定位。

关于倍帧问题

Question 22 调用倍帧接口没有效果

Answer 22

1. 倍帧适用于帧率在 20 以上的码流，若帧率在 20 以下的码流则调用倍帧接口无效；

关于 Android 版本的播放库问题

Question 23 Android 版本的播放库不能正常预览

Answer 23

1. Anroid 版本的播放库只支持真机，不支持虚拟机；
2. 建议使用品牌旗舰机型运行使用；